Received September 1, 2020, accepted September 10, 2020, date of publication September 18, 2020, date of current version September 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024951

A Robust Audio Fingerprinting Using a New Hashing Method

HEUI-SU SON[1], SUNG-WOO BYUN[1], AND SEOK-PIL LEE[2]
[1]Department of Computer Science, Graduate School, Sangmyung University, Seoul 13557, South Korea
[2]Department of Electronic Engineering, Sangmyung University, Seoul 13557, South Korea

Corresponding author: Seok-Pil Lee (esprit@smu.ac.kr)

This work was supported by the 2020 Research Grant from Sangmyung University.

ABSTRACT To enhance the tracking performance of illegal audio copies, we introduce a robust audio fingerprinting method against various attacks in this paper. Most audio fingerprints consist of the information in the frequency band of audio. These fingerprinting methods may lose the uniqueness of the audio fingerprint by irregular movement such as an attack with pitch value changes. The proposed fingerprint method in a fundamental frequency band makes up for the weakness of existing methods generated from frequency domain. Using the geometrical property of the proposed method, a new hashing method is employed in the similarity calculation process to compare the audio contents. In order to prove the validity of proposed algorithm, we experiment for six environments such as tempo, pitch, speed modification, noise addition, low pass filter and high pass filter. The proposed method shows the highest level of performance in most experimental environments. Especially, with respect to the tempo, pitch, and speed manipulation experiments, the proposed method archives the precision rate of the range from 95 to 100% according to the degree of manipulation, which compares favorably with the precision rate obtained by traditional approaches, and yields a precision rate between 85 and 100% in noise addition and filtering experiments.

INDEX TERMS Audio fingerprint, FFMAP, SAH, frequency band separation.

I. INTRODUCTION

Recently, media platforms such as YouTube have been growing at a fast rate, in the distribution and sharing of media contents. Also, a lot of content can be easily searched, especially audio contents [1]. This makes being able to access and enjoy the desired contents easily through various platforms, but causes many problems like illegal copies simultaneously [2]. Copyright infringement of audio data is a representative example. As the illegal use and the prohibited distribution of contents are constantly increasing, studies are ongoing to protect the copyright of the original audio and to detect illegally used data [3]–[5]. Today, anyone is able to easily copy and distribute audio content accessed on the internet. Moreover, if a manipulation aiming at deterring any detection is applied to the audio file, the uncovering of illegal use becomes difficult. Therefore, the necessity of copyright protection technology for audio data comes to the fore. The representative techniques to ensure the copyright of audio contents are audio watermarking [6], [7], and audio fingerprinting [8]–[11]. The audio fingerprint is the unique information that can represent a certain audio data. The Audio fingerprinting technique makes it possible to track pirates compared to the original in the database with this audio fingerprint.

The most audio fingerprinting methods employ frequency-based information from audio data to generate feature points to be used as audio fingerprints [12], [13]. However, the components of the frequency band can be changed irregularly when the speed or pitch manipulation occur in the original audio data. With the great potential, the audio fingerprints that are combined from these irregularly shifted frequency values cause the possibility of incorrect audio matching. The matching time is another remarkable point in existing methods. Given a short audio input or a small number of original audios stored in the database being compared, audio matching time may not be a big consideration. Conversely, short matching time is a very important technique when given large audio input or when searching among sufficiently large amounts of data is required. In addition, the lack of a techniques that

The associate editor coordinating the review of this manuscript and approving it for publication was Charith Abhayaratne.

allows components such as pitch to change in accordance with the manipulated audio to enable fast retrieval is also a problem.

To improve the performance of the audio fingerprinting, a fingerprint should capture and characterize the essence of the audio content. In this paper, we present an audio fingerprint method which is robust against various attacks. The fundamental frequency components are extracted and provided to preserve the majority of the audio's characteristics more effectively than STFT-based spectral peaks. The extracted components were matched with the frame-fundamental frequency domain and used to compose a fundamental frequency map (FFMAP). We employed a new hashing method named spatial adaptive hashing (SAH) in the similarity calculation process, to compare the audio contents. The method is able to compute optimal embedding along manipulated audio data so that correct similarity information is obtained between objects. In order to demonstrate the superiority of the proposed algorithm, we carried out comparison tests matching time and the performance with existing methods such as the quad-based method of Sonnleitner and Widmer [14], in addition to the method we proposed. The proposed method shows the highest level of performance in most experimental environments. Especially, with respect to the tempo, pitch, and speed manipulation experiments, the proposed method archives the precision rate of the range from 95 to 100% according to the degree of manipulation, which compares favorably with the precision rate obtained by traditional approaches, and yields a precision rate between 85 and 100% in noise addition and filtering experiments.

This paper is organized as follows. Section 2 introduces researches on existing audio fingerprinting technology. Section 3 discusses FFMAP and SAH, which are the key parts of the proposed method. Section 4 presents the experiment description and results, and then discusses and concludes with Sections 5 and 6.

II. RELATED WORK

To satisfy several practical requirements [15] for a successful audio fingerprinting system, various methods have been proposed [16]–[18]. Among the various approaches, 'Shazam' is the best-known public algorithm among studies about audio fingerprinting [17]. It uses spectrogram peaks as the local key points of an audio stream. It then generates a local descriptor using certain pairs of these key points. The local descriptor is based on the time difference between two adjacent peaks as well as their frequencies. The extracted fingerprints are highly robust to audio compression, foreground voices, and other types of noise. However, because of the nature of the descriptors, the algorithm is vulnerable to pitch or tempo manipulations. Based on the idea of Shazam, Pan *et al.* (2011) [18] proposed a local energy centroid for generating an audio fingerprint. Jiang *et al.* (2013) [19] proposed real-time peak-discovering method for audio fingerprinting.

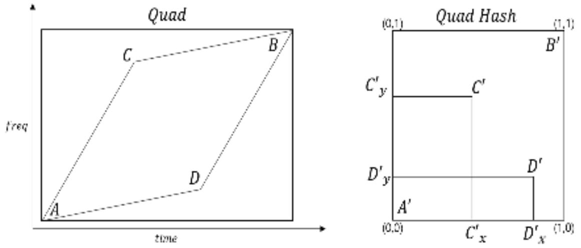


FIGURE 1. **The way to extract Quad-based audio fingerprint. The actual hash of the quad was given by C′ and D′, and was stored as a four-dimensional point (C′x, C′y, D′x, D′y) [14].**

Sonnleitner and Widmer (2015) [14] use a compact four-dimensional, continuous hash representation of quadruples of points called "quads", as shown in Fig. 1. We will take this as our reference method in the present paper, because it is the latest publication on this topic, and it reports high precision results for a certain range of speed and noise manipulation. Most audio fingerprinting algorithms applied approaches to extract audio fingerprint from spectrograms [16]–[22].

While most audio fingerprinting algorithms applied approaches to extract audio fingerprint from spectrograms, several studies captured and characterized the essence of the audio content from other domain instead of spectrograms [20]. Google uses an image processing technique to extract audio features from audio signals [22]. The STFT (Short Time Fourier Transform) converts a signal in the time domain into a signal in the time domain into the frequency domain to generate a spectral image. The wavelet transform is applied to the generated image to extract the top-t wavelet coefficients. A fingerprint is generated by obtaining a binary vector of the extracted top-t wavelets. This method uses hashing to efficiently use memory and provides robust performance against noise. Kim and Kim (2014) [20], based on the idea underlying Wang's method [17], proposed a method for extracting audio fingerprints from the modulated complex lapped transform, which improved the robustness of audio fingerprinting in an actual noisy environment. The authors extracted MCLT-based spectral peaks which preserve the majority of the sound's peaks more effectively. They showed that this approach is computationally efficient, and delivers high identification accuracy through experiments. Zhang *et al.* (2020) [21] proposed an audio retrieval method to modify both the fingerprinting and the matching to improve robustness against noise interference. The authors obtained a specific fingerprint from the intrinsic silent segments in both template and test audio in order to reduce the random bit errors caused by the noise. To match audio files, they divided the searching window into several segments for precise comparison between the template audio and the test audio.

III. AUDIO FINGERPRINTS BASED ON FFMAP

Audio fingerprinting technology is defined as extracting features from the original audio to generate unique information of the audio file and verifying copyright infringement. In order to be an ideal audio fingerprinting system, several conditions must be satisfied. The conditions are as follows [8].

- Accuracy: Accurate identification ability

- Reliability: Query audio must exist in a database for reviewing copyrights.

- Robustness: Accurate identification, despite external attacks such as compression, distortion and interference, should be possible.

- Security: It should be able to prevent the act of avoiding fingerprint detection through manipulation and deformation.

- Scalability: It has to have the high performance in the database of the large scale.

- Complexity: The efficient computing cost should be had in performing the algorithm.

Among the above conditions, this study prioritized satisfying 'accuracy', 'robustness' and 'scalability'. In other words, it focuses on precisely identifying the source of audio using the characteristics of the audio fingerprinting method based on fundamental frequency and ensuring the fast search time in a large amount of data by using the new hashing search method. Most audio fingerprinting algorithms, on the other hand, are implemented with satisfactory granularity. In this case, when the length of the audio query is short, a smooth search is performed. However, if the entire audio needs to be examined, the feature size becomes large and the audio identification is not done properly. In this study, we limit the audio input to almost half the length of the audio. This is to ensure that there is no problem in matching speed or performance in extracting and comparing feature values even if the input size of the audio is large enough.

A. COMPARISON WITH GENERAL STUDIES

Most studies of audio fingerprinting techniques perform Fourier transform on audio data, generate spectrograms in the time-frequency band, and the extract audio fingerprints by defining local maximums as local feature points of audio. The audio fingerprint generated through this process have uniqueness to the original audio, but the fingerprint is changed irregularly when the manipulation of components such as pitch or speed of the audio is applied. This is because local feature points in the frequency band cause irregular movement when the pitch manipulation is occurred as shown in Fig. 2 (a). The fingerprint value generated from irregularly shifted frequency value may also be irregular. Alternatively, (b) of Fig. 2 is an example of the distribution of local feature points in the time-fundamental frequency band of audio. In this method/band, even if the pitch of the audio modified, the fundamental frequency values generally move vertically. These characteristics can be used to design audio fingerprinting systems that resist a variety of attacks, including pitch and speed changes.

We present the audio fingerprinting algorithm generated from fundamental frequency band in our previous study [23]. This paper includes the experiments on various attack conditions such as the manipulation of pitch, tempo and speed and noise insertion. Such manipulations make most of the feature points of audio in fundamental frequency band shifted regularly. However, it is difficult to apply this characteristic of the fundamental frequency-based fingerprint to an attack that manipulates not the entire audio but only a specific part of the audio.

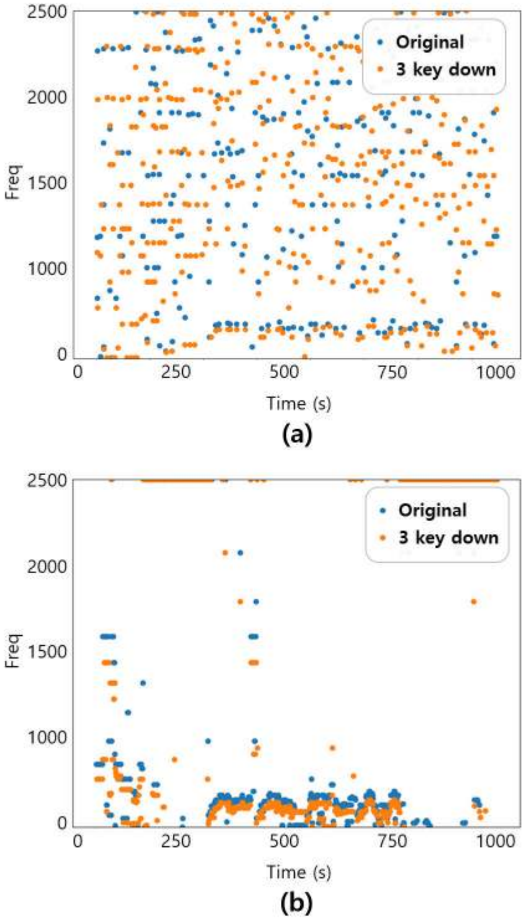


FIGURE 2. **The distribution of feature points from original audio and manipulated audio. (a) Local maxima comparison between an original audio with the 3 key down audio. (b) Fundamental frequency comparison between the original audio with the 3 key down audio [23].**

In this section, we present a fingerprinting method to reinforce the performance of existing audio fingerprinting for attacks that induce changes only in specific frequency ranges such as low pass filter and high pass filter. Fig. 3 is a simplified flow chart of the process of extracting an audio fingerprint. When the input audio data comes in, it enters the fingerprint extraction step. Each input audio is divided into audio having low frequency information and that having high frequency information through a low pass and a high pass filter. The cutoff frequency whose the filter is applied to audio in the experiments is set to 1000Hz. This is the middle point of the fundamental frequency range to be extracted, as described in Section 3.3. Therefore, a low fingerprint and a high fingerprint are finally extracted for one audio.

When unlabeled audio is given as an input, it goes through the same fingerprint extraction process as in Fig. 3. The audio fingerprint pre-extracted from the original audio is

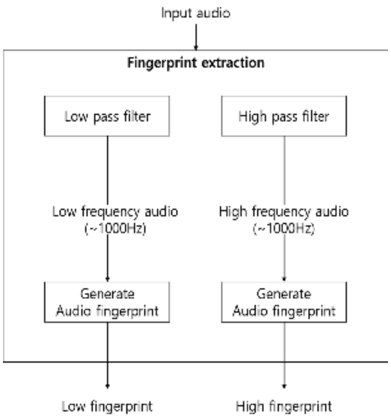


FIGURE 3. The extraction procedure of low/high frequency fingerprints.

stored in the database and then used as a similarity calculation with the audio fingerprint extracted from the unlabeled audio. After performing the similarity calculation, matched audio is returned for data satisfying the matching condition.

B. FEATURE EXTRACTION

The sound has the property of having to each frequency. There are very low frequency areas that are not heard by people, and there are high frequency areas that can hurt your ears. There are many characteristics in the frequency domain included in music. We extract the characteristics of music in the frequency range from 100 to 2000Hz among the frequency ranges of music. This frequency range includes from the very low notes of the instrument to the sound of instrument and the vocals, which people feel comfortable listening to except for the very high note of vocals and instruments. In order to extract the characteristics of audio accurately, it is necessary to remove the voiceless sounds from the audio. The voiceless sound is a signal of non-period, so if the periodicity of a particular section is not periodic, that section is judged to be a voiceless sound section and removed. Normalized autocorrelation is used as a method to investigate the periodicity of the certain section. The value of normalized autocorrelation is greater than or equal to the threshold it is determined as a periodic signal. And the threshold is set to 0.55 in our experiment. The equation of normalized autocorrelation is as follows:

$$R_s(l) = \sum_i s[i] \times s[i-l] \tag{1}$$

$$E(x) = \sum_i \left| x[i]^2 \right| \tag{2}$$

$$\text{Normalized autocorrelation} = \frac{\mathrm{R_s}(l)}{\sqrt{E(s[n]) \times E(s[n-1])}} \tag{3}$$

where R is autocorrelation function, and S and E are time series signal and the energy of the signal, respectively.

The voiceless sounds are removed from the input audio and their values match to zero. On the other hand, in case of voice sound, the value is matched with the corresponding frequency value where the normalized autocorrelation has the largest value in the range of 100 to 2000Hz.

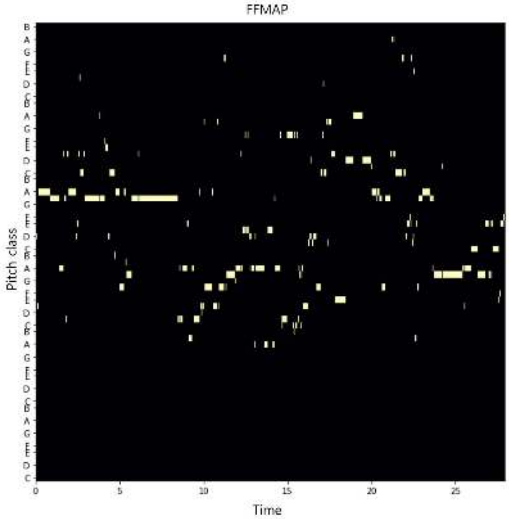


FIGURE 4. The example of fundamental frequency map (FFMAP) has x-axis as the number of frames of audio and the y-axis as the frequency value. Here, the white part has the value of 1 while the black has the value of 0 [23].

C. FFMAP

Once all section have been matched to their frequency values, they are mapped to a new two-dimensional space. The newly constructed two-dimensional space has x-axis as the number of frames of audio and the y-axis as the frequency value. We call this the FFMAP, which constructed by Eq. (4).

$$FFMAP(frame_i, frequency_i) = 1, \quad otherwise = 0 \tag{4}$$

The example of FFMAP is as shown in Fig. 4.

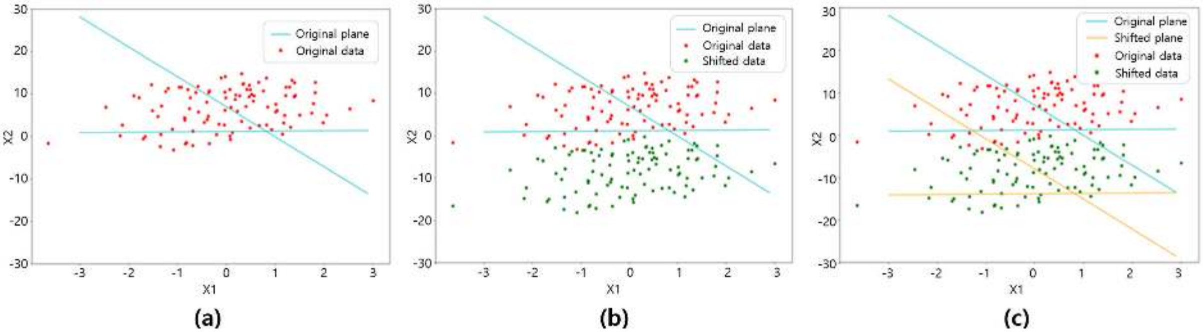
The frequency values matched to generate an audio fingerprint are mapped in order to the FFMAP. Since the height of FFMAP is a frequency value, the range of the value is always constant from 100 to 2000Hz. However, its width is proportional to the length of the audio. This mean that the FFMAP is asynchronous to the same two music when the change in music length is applied. Therefore, we add the process of normalizing the width of FFMAP to keep the same music in synchronization. An affine transform is used as normalization method whose expression is as follow in Eq. (5) and cubic spline interpolation is used as an interpolation method.

$$I(x, y) \begin{pmatrix} w & 0 \\ 0 & H \end{pmatrix} = I^*(x, y) \tag{5}$$

where I is an original image, I^* is a scaled image, and w and H are the width and the height of the music. The w is set to 2000 in our experiment.

D. AUDIO RETRIEVAL SYSTEM

The purpose of retrieval systems is returning a set of objects whose similarity are close to a query object. Thus, it is



(a)

(b)

(c)

FIGURE 5. The examples of an extracted fingerprint and a shifted fingerprint manipulated with pitch. (a) An original vector and original random planes (b) The distribution of a shifted vector (c) Shifted random planes along the shifted vector.

significantly important to define the similarity between objects in the retrieval system. Representative methods for obtaining such similarities are Euclidean distance, Manhattan distance, Cosine similarity, and others. However, these methods calculate the distances of all the stored data pairs, the similarity review of those is very time-consuming process.

Hashing techniques enable the system to fast compare similarity between numbers of objects [24]–[27]. It evaluates the similarity between i-th and j-th objects by calculating Hamming distance based on generated binary hash codes. The calculated Hamming distance can be closely approximated to distance between the two hash codes, vector, z_i and z_j. Assuming an audio fingerprint defined in section 3.4 is extracted from audio data, it can be distributed as shown in Fig. 5 (a). The X axis and Y axis of this extracted vector are time and pitch of the audio, respectively. If pitch and tempo are changed by manipulating the original audio, the Y axis is shifted as in Fig. 5 (b). Similarity calculated by Euclidean distance method of the original vector and the manipulated vector is different even though the two vectors have similar shapes in 2-D space. That is, when conventional hashing methods are used, the similarity is decreased in these conditions. The reason is that when generating the hash code form the shifted vector, the original plane is used in the same way as original vector. Therefore, to consistently generate hash codes despite manipulated audio data, an audio retrieval system needs the planes that can shift along the shifted vector as shown in Fig. 5 (c). This research proposes Spatial Adaptive Hashing method which is able to compute optimal embedding along manipulated audio data so that correct similarity information between objects. This can closely approximate Pearson's correlation coefficient which is a measure of the linear correlation between two variables X and Y.

$$\mathrm{z}_{1\times l} = x_{1\times n} \cdot w_{n\times l} - \left(\sum_{i=1}^{n} w_{il} \times \mu_x\right) \tag{6}$$

$$\mathrm{h}_{1\times 1} = T(z_{1\times l}) \tag{7}$$

where x and w are an input vector and random planes referred as hash function. Also, μ_x and n are the average of the input vector x and the length of x vector. Here, a function, given as if $x > 0$ $T(x) = 1$ and otherwise $T(x) = 0$, is applied to convert an input vector to a binary vector. The hash functions of the equation adaptively generate hash codes of length $1 \times l$ along the average of the input vectors. Where l is the length of generated hash code. Based on preliminary experiments, we set l to 200 for fingerprint that is a 2000-dimensional input vector.

TABLE 1. Retrieval speed comparison between the methods.

Method	Quad-based	None	SAH
Average time(ms)	630	230	10

The fingerprint vector x extracted from FFMAP is converted to the binary hash code h by Eq. (6) and Eq. (7). After converting audio objects to hash codes, the similarity between two audio objects o_p and o_q is replaces by the calculation of the hamming distance between the generated 200-dimensional hash codes as follows:

$$\mathrm{d}(\mathrm{o_p}, o_q) = d_h(h_p, h_q) \tag{8}$$

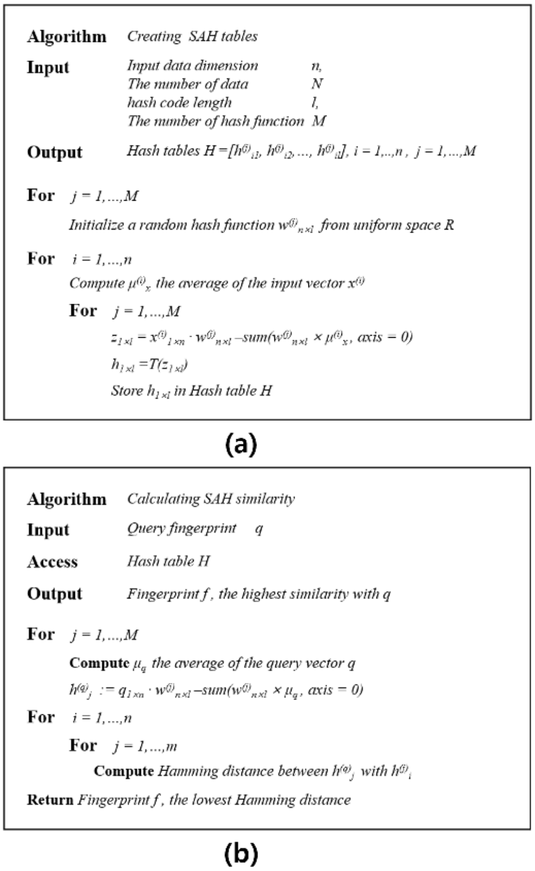
$$d_h = \sum_{i=0}^{l} \left| h_p^{(i)} - h_q^{(i)} \right| \tag{9}$$

where d_h is hamming distance between two hash codes.

Fig. 6 (a) and (b) show constructing a hash table and calculating similarity through SAH.

The benefit of such hashing algorithm is to improve retrieval speed. To verify the performance of SAH, we compare the method with other methods in Tab. 1 in terms of matching time. From now on, the FFMAP fingerprinting method with Pearson's correlation coefficient is called the None method [23].

The average time of Quad-based method for calculating similarity between the databases with one audio is about 630ms. When similarity is calculated using the None method, the average time is about 230ms. Meanwhile, the retrieval speed of SAH method takes 23 times as fast as the None method and 63 times as fast as Quad-based method. Therefore, the proposed hashing algorithm considerably improve matching time as compared to the other methods.



```
Algorithm   Creating SAH tables
Input       Input data dimension            $n$,
            The number of data              $N$
            hash code length                $l$,
            The number of hash function     $M$
Output      Hash tables $H = [h^{(j)}_{i1}, h^{(j)}_{i2}, \ldots, h^{(j)}_{il}]$, $i = 1,\ldots,n$, $j = 1,\ldots,M$

For  $j = 1,\ldots,M$
    Initialize a random hash function $w^{(j)}_{n\times l}$ from uniform space $R$
For  $i = 1,\ldots,n$
    Compute $\mu^{(i)}_x$ the average of the input vector $x^{(i)}$
    For  $j = 1,\ldots,M$
        $z_{1\times l} = x^{(i)}_{1\times n} \cdot w^{(j)}_{n\times l} - sum(w^{(j)}_{n\times l} \times \mu^{(i)}_x, axis = 0)$
        $h_{1\times l} = T(z_{1\times l})$
        Store $h_{1\times l}$ in Hash table $H$
```

(a)

```
Algorithm   Calculating SAH similarity
Input       Query fingerprint   $q$
Access      Hash table $H$
Output      Fingerprint $f$, the highest similarity with $q$

For  $j = 1,\ldots,M$
    Compute $\mu_q$ the average of the query vector $q$
    $h^{(q)}_j := q_{1\times n} \cdot w^{(j)}_{n\times l} - sum(w^{(j)}_{n\times l} \times \mu_q, axis = 0)$
For  $i = 1,\ldots,n$
    For  $j = 1,\ldots,m$
        Compute Hamming distance between $h^{(q)}_j$ with $h^{(j)}_i$
Return Fingerprint $f$, the lowest Hamming distance
```

(b)

FIGURE 6. The algorithm of creating SAH tables. (b) The algorithm of calculating SAH similarity.

IV. EXPERIMENTS

In this section, we present the performances of proposed FFMAP based audio fingerprinting method. We compare the performance with Quad-based method and None method. Son *et al.* [23] is the method of fingerprint generation in the fundamental frequency band of the original data, and it use Pearson's correlation score as an audio matching method. For convenience, the methods proposed by Sonnleitner *et al.* (2015) [14] and Son *et al.* [23] are called quad-based and none respectively. In this paper, the data to be used as an input audio is the entire audio. Thus, it is not appropriate to proceed performance comparison with audio fingerprinting method such as Shazam with a short length of input data.

The dataset used in the experiments consists of 100 audios. The original audio data are genre-insensitive, and the length of them is a minute, the sampling frequency is 8000Hz on a mono channel. The experimental environments are largely defined as 1)tempo manipulate, 2)pitch manipulate, 3)speed manipulate, 4)noise addition, and 5)filtering. Performance comparisons are proceeded in all environments for the quad-based and None methods. Among these, experimental environments 1), 2), 3), and 4) are some of the conditions defined in the papers that proposed the Quad-based method, and 5) are the condition which is additionally introduced in this paper.

The proposed method classifies the same audio if the hamming distance is less than or equal to 80 when the hash code length l is 200. In the case of None method, it determines same audio if the Pearson's correlation value is greater than or equal to 0.4 in the similarity calculation. The matching criteria between audio fingerprints used in the Quad-based method is different.

Precision is used as an index for evaluating the performance of the algorithm. The *tp* and *fp* used to express precision are confusion matrix elements. The *tp* (true positive) signifies the number of times that data with True value is classified as True by classification model, and *fp* (false positive) means the number of times that data with False value is determined as True by classification model. Therefore, precision is the ratio of what the model classifies as True to what is actually True. When applied to the corresponding algorithms, it implies the ratio of well-matched times of total matching times. Precision can be expressed as Eq. (10).

$$precision = \frac{tp}{tp + fp} \tag{10}$$

Before moving on to the results of the various experiments, definitions of the attacks are needed. "Tempo" manipulation attack is an attack environment that only changes the time scale of audio without changing the pitch. On the other hand, a "speed" manipulation attack is define as an attack environment in which the time scale and pitch value of the original audio are simultaneously changed proportionally.

Tempo, pitch, and speed attacks all vary from 70% to 130% relative to the original audio, with a 10% interval between changes. Next, in the noise addition attack, white noise is added to the original audio. The noise range is from 5dB to 50dB, and the applied noise interval is 5dB. SNR(Signal Noise Ratio) is a measure of how large the signal of the original audio is compared to the noise. It is expressed in decibels(dB). The larger the SNR value, the weaker the noise level.

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{11}$$

Last, the filtering attack is divided into a low pass filter and a high pass filter. In the both filters, the filtering cut-off frequency range is applied at intervals of 100Hz from 500 to 1000Hz.

A. TEMPO MANIPULATION

The first experimental environment is the tempo manipulation attack. The vertical axis of Fig. 7 signifies precision of each methods, and the horizontal axis presents the relative length of manipulated audio compare to original one.

Both SAH and None method are the audio fingerprinting system based on fundamental frequency. Thus, regardless of the degree of manipulation, they get 100% of precision in the tempo attack environment. Quad-based method also achieves 100% of precision in the range from 80 to 120%, but the performances decrease significantly at 70% and 130% where the degree of manipulation is large.

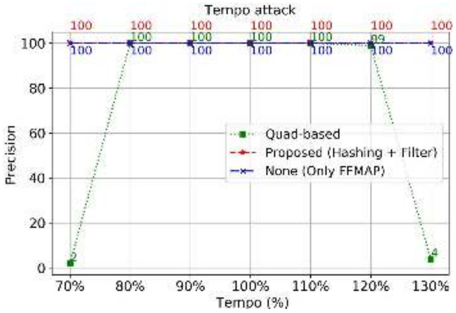


FIGURE 7. Comparison results of resisting the tempo manipulation.

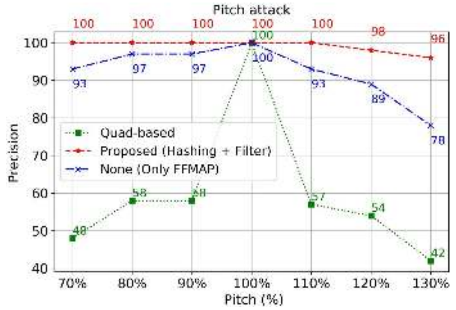


FIGURE 8. Comparison results of resisting the pitch manipulation.

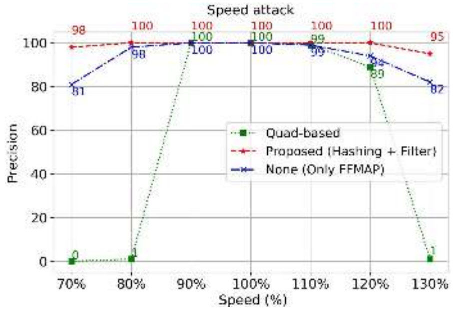


FIGURE 9. Comparison results of resisting the speed manipulation.

B. PITCH MANIPULATION

Next, the horizontal axis in Fig. 8 means relative pitch degree of manipulated audio compared to the original one. The 10% of manipulation rate is defined as semitone. And the vertical axis also signifies the precision of each algorithms.

The SAH method maintains its performances high in spite of large amount of manipulate rate, whereas the Quad-based method shows poor performances in all conditions except for the original state. The None method achieves flat precision rate, but its performance drops significantly in condition with the 3 semitones increase attack.

C. SPEED MANIPULATION

In Fig. 9, the horizontal axis presents the relative length of manipulation audio. In the speed manipulation attack, the change occurs proportionally to the time scale and pitch of the audio.

The SAH method achieves high performances in all conditions. The Quad-based method shows high performances similar to other methods in the range of 90 to 120% but does not perform sufficiently in the large variation range.

D. NOISE ADDITION

In our experiment, noise addition attacks can be classified into two types: 1) inserting white noise into the original audio and 2) inserting white noise into the speed manipulated audio. In the figure of experimental results with the noise attack, the horizontal axis of the graphs represents the SNR of the manipulated audio.

Fig. 10 is experimental result of noise addition environment with original audio and speed manipulation. In Fig. 10 (a), all algorithms show 100% of precision in the SNR range of 10 to 50dB. In the case of 5dB of SNR, the Quad-based and None methods do not achieve 100% of precision, but still record high detection rate of 98% and 97%, respectively. In the environment where noise is inserted into the original audio, the performances of three methods are almost the same.

Next are the noise addition environments with speed manipulated audio. The manipulated speed is divided into 95% and 105% of speed compare to the original audio. In Fig. 10 (b), SAH shows nearly 100% of precision in all conditions, which is the highest performance among those of the methods. In Fig. 10 (c), an environment operated at a 105% of speed, similar to in Fig. 10 (b), SAH achieves nearly 100% of detection rate in all conditions. In the environment of noise addition with speed manipulation, SAH, Quad-based, and None methods show superior performance in order.

E. FILTERING

Filtering attack is an attack environment that shows remarkable improvement in performance of proposed method compare to None method, which is a method of extracting fingerprints from the original audio.

Fig. 11 is the experimental result of applying low pass filter and high pass filter to audio. The horizontal axis in graph is the value of the cut-off frequency applying the filters.

In Fig. 11 (a), compared with the None method, the performance of the SAH method, which uses fingerprints generated from two audio having low and high frequency information, is much improved. On the other hand, the Quad-based method shows a precision of 66% under the condition of 1000Hz cut-off frequency, but rarely detected in the remaining sections.

High pass filtering is an experimental environment that retrieves audio only with high frequency components of audio data, as opposed to the environment that applies low pass filter. In Fig. 11 (b), Quad-based method has the highest detection rate in the range of cut-off frequency 500 to 900Hz. However, the precision is greatly decreased to 84% under the condition that the cut-off frequency is 1000Hz. On the other hand, SAH method shows the most stable performance overall.

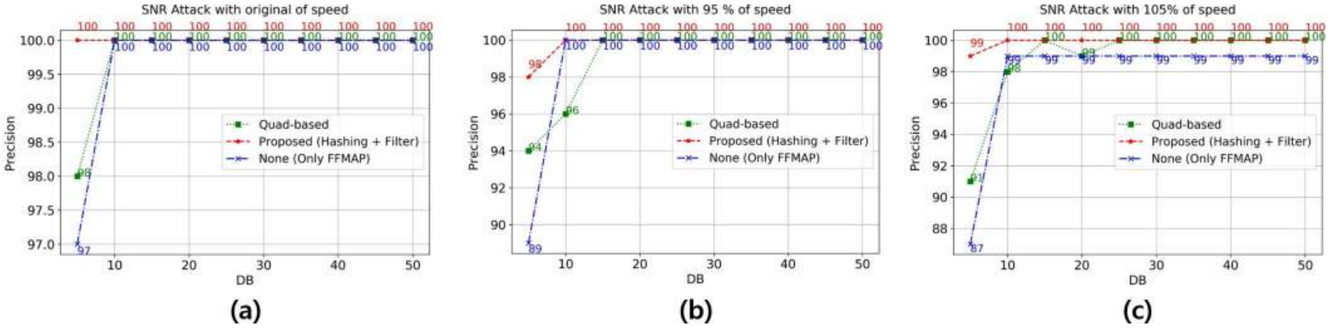


FIGURE 10. Comparison results of resisting the noise adding with original data (a), the noise adding with speed decreasing (b), and the noise adding with speed increasing (c).

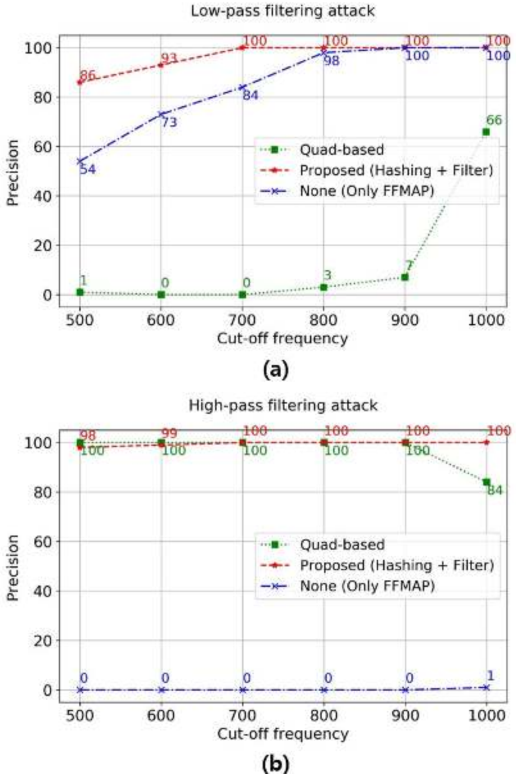


FIGURE 11. Comparison results of resisting the low-pass filtering attack (a) and the high pass filtering attack (b).

V. DISCUSSION

In this study we developed a method to extract fundamental frequency-based audio fingerprints. By applying a hashing method reflecting the characteristics of the audio contents, the similarity between audio data is effectively calculated. Since audio data can change irregularly in speed or pitch manipulation, audio retrieval systems need more sophisticated analysis to reflect the irregular properties. The proposed method using FFMAP is utilized effectively to search manipulated audio data. Previous studies extracted fingerprints using spectral peaks [16]–[22], but these were affected by the characteristics of the spectral peaks to change irregularly according to manipulations. Utilizing the fundamental frequency, the proposed method captured and characterized the essence of the audio content. The retrieval performance of the proposed method in extensive experiments showed that the method is an effective approach to an audio fingerprint system.

VI. CONCLUSION

In this paper, we propose an audio fingerprint method has a robustness against various attacks. The fundamental frequency components extracted from an audio data are matched with frame-fundamental frequency domain and compose fundamental frequency map (FFMAP). The initial FFMAP is returned after several steps to normalize it. We employ a new hashing method named as spatial adaptive hashing (SAH) in the similarity calculation process to compare the similarity between the audio contents. Similarity verification between the generated hash codes is performed through the hamming distance calculation. In order to prove the superiority of the proposed algorithm, we accomplish comparison tests for the matching time and the performance with the existing methods. When SAH method is applied, the time required for the matching audio is improved about 23 times compared to the matching method using the existing Pearson's correlation score. The proposed method in this paper is to perform the most stable audio detection with the comparison of Quad-based and None method in various experimental environments for random audios without any distinction of genre.

REFERENCES

[1] L. Xiang, X. Shen, J. Qin, and W. Hao, "Discrete multi-graph hashing for large-scale visual search," *Neural Process. Lett.*, vol. 49, no. 3, pp. 1055–1069, Jun. 2019.

[2] M. Malekesmaeili and R. K. Ward, "A local fingerprinting approach for audio copy detection," *Signal Process.*, vol. 98, pp. 308–321, May 2014.

[3] C. Bellettini and G. Mazzini, "A framework for robust audio fingerprinting," *J. Commun.*, vol. 5, no. 5, pp. 409–424, May 2010.

[4] H. Jegou, J. Delhumeau, J. Yuan, G. Gravier, and P. Gros, "BABAZ: A large scale audio search system for video copy detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 2369–2372.

[5] Z. Mehmood, K. A. Qazi, M. Tahir, R. M. Yousaf, and M. Sardaraz, "Potential barriers to music fingerprinting algorithms in the presence of background noise," in *Proc. 6th Conf. Data Sci. Mach. Learn. Appl. (CDMA)*, Mar. 2020, pp. 25–30.

[6] H.-T. Hu and L.-Y. Hsu, "Robust, transparent and high-capacity audio watermarking in DCT domain," *Signal Process.*, vol. 109, pp. 226–235, Apr. 2015.

[7] I. Milas, B. Radovic, and D. Jankovic, "A new audio watermarking method with optimal detection," in *Proc. 5th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2016, pp. 116–119.

[8] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of audio fingerprinting," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 41, no. 3, pp. 271–284, 2005.

[9] J. S. Seo, "An asymmetric matching method for a robust binary audio fingerprinting," *IEEE Signal Process. Lett.*, vol. 21, no. 7, pp. 844–847, Jul. 2014.

[10] G. Yang, X. Chen, and D. Yang, "Efficient music identification by utilizing space-saving audio fingerprinting system," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2014, pp. 1–6.

[11] G. Sharma, K. Umapathy, and S. Krishnan, "Trends in audio signal feature extraction methods," *Appl. Acoust.*, vol. 158, Jan. 2020, Art. no. 107020.

[12] N. Jiang, P. Grosche, V. Konz, and M. Müller, "Analyzing chroma feature types for automated chord recognition," in *Proc. Audio Eng. Soc. Conf., 42nd Int. Conf., Semantic Audio*, 2011, pp. 1–10.

[13] X. Anguera, A. Garzon, and T. Adamek, "MASK: Robust local features for audio fingerprinting," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 455–460.

[14] R. Sonnleitner and G. Widmer, "Robust quad-based audio fingerprinting," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 3, pp. 409–421, Mar. 2016.

[15] Y. Liu, H. S. Yun, and N. S. Kim, "Audio fingerprinting based on multiple hashing in DCT domain," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 525–528, Jun. 2009.

[16] X. Zhang, B. Zhu, L. Li, W. Li, X. Li, W. Wang, P. Lu, and W. Zhang, "SIFT-based local spectrogram image descriptor: A novel feature for robust music identification," *EURASIP J. Audio, Speech, Music Process.*, vol. 2015, no. 1, p. 6, Dec. 2015.

[17] A. Wang, "An industrial strength audio search algorithm," Tech. Rep., 2003, pp. 7–13.

[18] X. Pan, X. Yu, J. Deng, W. Yang, and H. Wang, "Audio fingerprinting based on local energy centroid," in *Proc. IET Int. Commun. Conf. Wireless Mobile Comput. (CCWMC)*, Shanghai, China, Nov. 2011, pp. 351–354.

[19] T. Jiang, R. Wu, J. Li, K. Xiang, and F. Dai, "A real-time peak discovering method for audio fingerprinting," in *Proc. 5th Int. Conf. Internet Multimedia Comput. Service (ICIMCS)*, Huangshan, China, Aug. 2013, pp. 368–371.

[20] H.-G. Kim and J. Y. Kim, "Robust audio fingerprinting method using prominent peak pair based on modulated complex lapped transform," *ETRI J.*, vol. 36, no. 6, pp. 999–1007, Dec. 2014.

[21] X. Zhang, G. Zhan, W. Wang, P. Zhang, and Y. Yan, "Robust audio retrieval method based on anti-noise fingerprinting and segmental matching," *Electron. Lett.*, vol. 56, no. 5, pp. 245–247, Mar. 2020.

[22] S. Baluja and M. Covell, "Audio fingerprinting: Combining computer vision & data stream processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, vol. 2, Apr. 2007, pp. II-213–II-216.

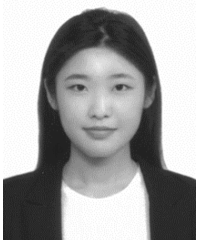
[23] H.-S. Son, S.-W. Byun, and S.-P. Lee, "Illegal audio copy detection using fundamental frequency map," in *Proc. 16th Int. Joint Conf. e-Bus. Telecommun.*, 2019, pp. 356–361.

[24] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," *VLDB*, vol. 99, no. 6, pp. 518–529, 1999.

[25] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.

[26] Y. Hao, T. Mu, R. Hong, M. Wang, N. An, and J. Y. Goulermas, "Stochastic multiview hashing for large-scale near-duplicate video retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 1, pp. 1–14, Jan. 2017.

[27] S. Byun and S. Lee, "Stochastic non-linear hashing for near-duplicate video retrieval using deep feature applicable to large-scale datasets," *TIIS*, vol. 13, no. 8, pp. 4300–4314, 2019.



HEUI-SU SON received the B.S. and M.S. degrees in media software and computer science from Sangmyung University, Seoul, South Korea, in 2018 and 2020, respectively. Her main research interests include signal processing, artificial intelligence, and audio digital processing.

SUNG-WOO BYUN received the B.S. degree from the Department of Digital Media Technology, Sangmyung University, Seoul, South Korea, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. His main research interests include signal processing, artificial intelligence, and personalized media processing.

SEOK-PIL LEE received the B.S., M.S., and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, South Korea, in 1990, 1992, and 1997, respectively. From 1997 to 2002, he was a Senior Research Staff with Daewoo Electronics, Seoul. From 2002 to 2012, he was the Head of the Digital Media Research Center, Korea Electronics Technology Institute. He was also a Research Staff with Georgia Tech, Atlanta, GA, USA, from 2010 to 2011. He is currently a Professor with the Department of Electronic Engineering, Sangmyung University. His research interests include artificial intelligence, audio digital processing, and multimedia searching.