A NOVEL METHOD OF PLAGIARISM DETECTION FOR MUSIC MELODIES

Mu-Syuan Sie, Cheng-Chin Chiang, Hsiu-Chun Yang And Yi-Le Liu

Department Of Computerscience And Information Engineering
National Dong Hwa University, Taiwan

ABSTRACT

This paper presents an Automatic Music Melody Plagiarism Detection (AMMPD) method to detect and locate the possible plagiarism in music melodies. This proposed method addresses the challenging issues encountered in the AMMPD problem, including (1) the inexact matching of noisy and inaccurate pitches of music audio and (2) the fast detection and positioning of similar subsegments between suspicious music audio. The major novelty of our work is that the above two issues which are related toaudio signals of temporal domain are addressed by means of a novel path finding approach on a binarized 2-D image in spatial domain. In fact, the proposed AMMPD method can not only identify the similar pieces inside two suspicious music melodies, but also retrieve music audio of similar melodies from a music database given a humming or singing query. Experiments have been conducted to assess the overall performance and examine the effects of various parameters introduced in the proposed method.

KEYWORDS

Music Melody Plagiarism, Music Melody Retrieval, Subsequence Matching, Warping Time Series Join

1. INTRODUCTION

No matter how the lifestyle changes, listening to music is always the most affordable leisure activities for people to relieve the stress of life. Before the advent of the Internet era, people can enjoy a large amount of music over radio and television programs. Nowadays, the Internet has become the most convenient and fastest channel not only for acquiring music contents made by professional music producers, but also for sharing and publishing the compositions of amateur music writers. As a result, the vast amount of shared musical contents over the Internet also serve as the source of inspirations for many music writers, bringing about more and more great musical productions. Besides, the well-developed infrastructure of mobile communication is aggressively promoting the commercial market of networked music, thereby attracting more and more music composers to create their own musical contents.

Under the increasing demand from music-goers, fast creation of musical contents becomes an important policy of boosting the profits for many music producing companies and individual music producers. Additionally, internationalized and cross-cultural music creation is also getting more and more vital in expanding the commercial music markets. Therefore, music writers often rely more upon the stimulation from existing contents made by others to compose their new contents. During the composition, some writers can even intentionally or unintentionally plagiarize some parts of others' work, in either musical ideas or content pieces, without legal permits. Music plagiarism is illegal and harms seriously not only the intellectual property rights of other creators but also the environment of music creation and the development of the commercial market. To prevent the piracy, music plagiarism detection (MPD) is very important. However, detecting the plagiarism of music requires both time and musical expertise. To find if there are any identical or similar melodic segments in two songs, an expert needs to listen to the whole piece of each song for several times because the plagiarism may appear at any place in the

songs. Therefore, a system that can automatically locate the suspicious segments in two songs would be highly desirable to reduce the time and efforts. Besides the prevention of illegal piracy of music composition, such a system is also a useful assisting utility that helps the composer search very quickly the music pieces of their interests from a large collection of music contents. Perhaps, we can even expect the automatic composition of new music artworks by properly blending the compatible or coherence music melodies identified by the system. For consumers, this system can also mine the personal preferences by analyzing the common music patterns in the musical albums of personal interests and then recommend better contents for the consumers according to the preferences. Plagiarism in music, involving the using or imitation of other authors' music work, can occur in two contexts, including the musical idea (e.g., mimicking a melody or motif) and sampling (reusing a portion of one sound recording). Cases in the former context often cause ambiguity and difficulty in the arbitrating procedure because the melodies of plagiarized segments are usually not entirely the same. Aiming at this issue, we tackle the issue of locating identical and similar parts in two suspicious song melodies. A plagiarism detection system is different from a common music retrieval system. A music retrieval system compares one or more music query segments with music entries in an archive. The retrieval system outputs a music entry if the query segments and the entry or a part of the entry are alike. As the starting point and the endpoint of every given query segment are known, the system can locate the possible targets in database entries by simpler sequence matching. In contrast, a music plagiarism detection system must find all suspicious partial segments in two songs. As the plagiarized segments can be anywhere in the songs, the starting points and endpoints of suspicious segments are not known. Moreover, the deliberate melodic variations and ornaments in plagiarized music pieces can easily deceive those common techniques based on exact subsequence matching. Hence, locating the suspicious segments of music plagiarism is much more complicated and costlier than that in a common music retrieval system. The objective of our work presented in this paper is to propose a solution to address these challenging issues for developing an AMMPD system.

2. RELATED WORK

We have not found any related study on the issues of AMMPD according to our literature survey. Most prior arts that have been published are applications relating to retrieval of musical contents, such as the query by singing/humming (QBSH) systems [1, 2, 3]. The QBSH problem is different from the AMMPD problem. As mentioned previously, the query input in the QBSH problem has a clear pair of end points (both the starting point and the ending point) for matching with music entries in amusic archive. However, for the AMMPD problem, detecting and locating the plagiarised pieces inside two suspicious melodies are more cumbersome because we are not aware of where the plagiarised pieces may appear in both the query input and a database entry. Moreover, the problem becomes more complicated if multiple plagiarised pieces exist inside two suspicious melodies.

The Dynamic Time Warping (DTW) [4] algorithm, which is a nonlinear sequence matching algorithm, is a common way to address the QBSH problem. Athitsos et al. [5] proposed a DTW-based method for approximate subsequence matching of time series. Their method is applicable only to the matching between the whole query input and subsequences of a sequence. For the AMMPD problem that requires intensive subsequence matching of both the query input and a database sequence, their method is thus not feasible. Some methods underlain by the longest common subsequences algorithm [6] are also not suitable because these methods deal with only exact matching of noisy-free sequences. The AMMPD requires the inexact matching because noises are very common in the extracted tonal features of music audio. Lin et al. [7] presented a rough longest common subsequence algorithm to address the music retrieval problem which is like the QBSH problem, but is still not feasible to solve the unknown endpoints issue of AMMPD problem. Not aiming at the problem of music processing, Chen et al. proposed a warp time series

join (WTSJ) algorithm to find the similar subsegments between two motion sequences as the best part for fusing the two motion sequences. Though the WTSJ algorithm can locate similar pieces in two motion sequences, the algorithm may suffer from some problems, such as poor alignment and over-segmentation of similar pieces, due to the weak capability in handling noisy elements in sequences.

In this paper, we deal with the subsequence matching of two 1-D audio signal sequences of the AMMPD problem by an approach of 2-D spatial imageprocessing and analysis. The proposed method tackles the subsequence matching problem for AMMPD by the path exploration method over a binary image mask, which incorporates some image processing techniques, such as gap filling, block partitioning, connected path finding, and polyline approximation of distorted paths. Some tricky designs have been incorporated in the proposed method to handle the noisy tonal features of music audio. The proposed method can not only locate all possible plagiarised subsegments in two suspicious melodies, but also retrieve music contents that contain the melody pieces similar to the input query of a music clip. Hence, the proposed method is feasible for both the AMMPD problem and the QBSH problem.

3. THE PROPOSED METHOD

3.1. Overview

Fig. 1 shows the process flow of the proposed AMMPD method. Given two suspicious music melodies, the pitch extraction module extracts the pitch vectors, $Q = [q(1), q(2), \dots, q(F_1)]$ and $D = [d(1), d(2), \dots, d(F_2)]$, of the two melodies by a pitch tracking method. Since the two pitch vectors may contain some noisy pitch elements, we smoothen them by applying the median filter of the window size m. From the two smoothed pitch vectors, the pairwise distances between the pitch elements in the pitch vectors are calculated to derive a local distance matrix. By setting a threshold ε, we binarized this local distance matrix as a binary 2-D mask M where an entry $M(i, j) = 1$ indicates a possible correspondence between the pitch pair $(q(i), d(j))$ in the two melodies Q and D. In contrast, an entry of 0 on the mask M means a dissimilar pitch pair in the two melodies. Hence, a rectangular block containing contiguous entries of 1's on M actually identifies preliminarily a potential similar subsegments in the two sequences. One important task for the subsequent processing is to partition the mask into several rectangular blocks that reveals the potential plagiarised subsegments. To avoid the over-partitioning of the mask due to the unexpected pairwise pitch dissimilarity caused by some singular pitch elements, we fill the minor gaps, whose sizes are defined by a parameter g, between blocks before the mask partitioning. After the minor gap filling, the binary mask M is partitioned by finding vertical and horizontal gaps (consecutive entries of 0 along vertical and horizontal directions) between the blocks. Note that the partitioning is repeatedly done on every block until no vertical or horizontal gap appears on the block. After the partitioning, all paths of connected 1-entries inside each block are explored. Meanwhile, the detailed correspondences between the pitch elements along the path are also derived. Afterwards, multiple overlapping paths in each block can be ranked according to the derived detailed correspondences. An explored path may also have some distorted sections that reveal unreasonable many-to-one or one-to-many correspondences between pitch elements. These many-to-one and one-to-many correspondences alone the path can usually be found on the sections that are horizontal/near-horizontal or vertical/near-vertical. Hence, we remove these distorted sections by approximating the path with multiple polylines and removing those polylines that are horizontal/near-horizontal or vertical/near-vertical. Eventually, those paths that are too short to identify plagiarised subsegments are eliminated.

3.2. Pitch Vector Extraction of Music Audio

A digitized music data contains a sequence of audio data sampled at a fixed rate such as 8KHz or 16KHz. The data sequence is then partitioned into frames with each containing a fixed number of samples and overlapping partially with its neighboring frames. Applying the pitch tracking method [8] to each frame, we can estimate the fundamental frequency of the audio samples as the pitch of the frame. Given a fundamental frequency f, the pitch corresponds to a semitone $n = 69 + 12\log_2(f/440)$ which ranges normally over 35~72 for male sounds and 45~83 for female sounds. Therefore, an music audio of F frames can get a pitch vector of F semitones to characterize the tonal feature.

Unfortunately, the estimated fundamental frequency of each frame is not always robust under the disturbance of noisy audio signal. The output pitch vector may contain some noisy semitones which can incur undesired effects on matching music melodies. To reduce the effect of the noisy semitones, we perform the median filtering on the elements of the pitch vector to remove some singular noise pitch elements. However, for some continuous occurrences of noisy pitch elements, other methods is still necessary to remedy the possible side effects. We will present a way in later descriptions of the proposed method.

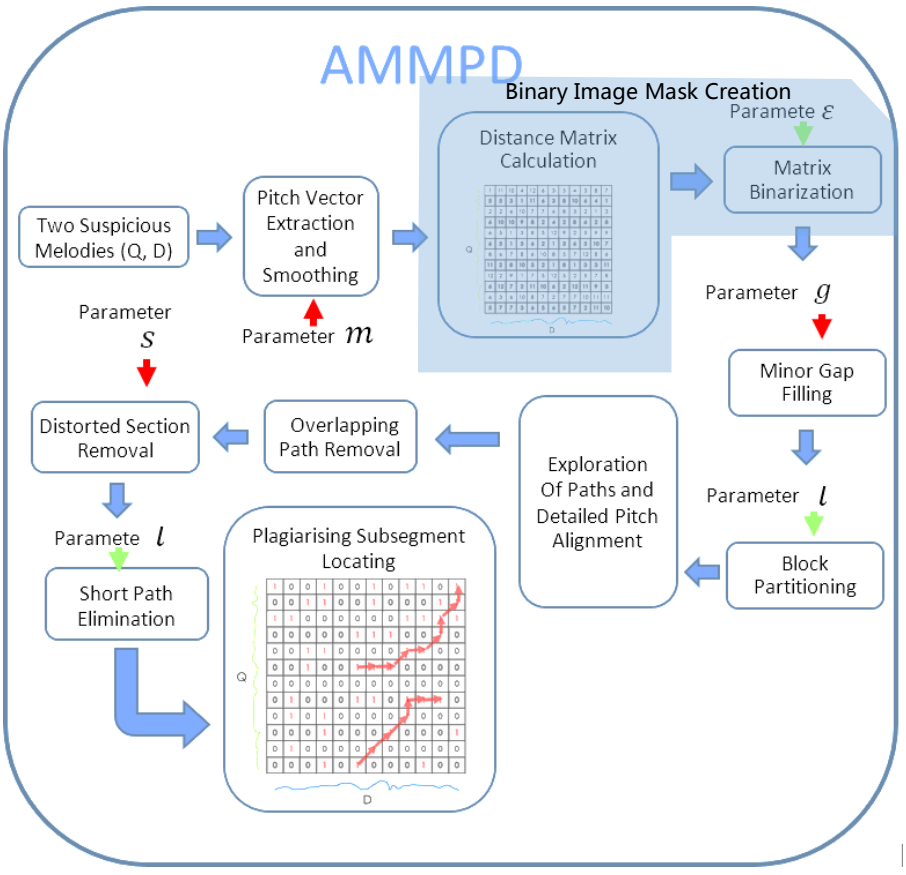


Figure 1: The process flow of the proposed AMMPD method.

3.3. Creation of Binary Image Mask

Let $Q = [q(1), q(2), \dots, q(F_1)]^T$ and $D = [d(1), d(2), \dots, d(F_2)]^T$ be the pitch vectors of two suspicious music melodies. Computing the pairwise distances between the pitch elements in the two vectors lead to a 2-D distance matrix $\mathbb{D}(i,j) = |q(i) - d(j)|$ for $1 \le i \le F_1$ and $1 \le j \le F_2$. The entries in the matrix $\mathbb{D}$ can be quantized into similar indications and dissimilar indications by specifying a distance threshold ε which defines the maximal dissimilarities allowed between the pitch elements of two plagiarised pieces. In other words, we can derive a 2-D binary mask M by

$$\mathrm{M(i,j)} = \begin{cases} 1, & if\ \mathbb{D}(i,j) < \varepsilon, \\ 0, & otherwise, \end{cases}$$

where an an entry $M(i,j)$ of 1 indicates a similar pair of pitch elements $(q(i), d(j))$ in the two suspicious melodies. On the contrary, an entry of 0 implies that the pitch elements $q(i)$ and $d(j)$ are too different to be two corresponding pitches in plagiarised pieces. Fig. 2 shows an example of $\mathbb{D}$ and M with the threshold ε of 4.

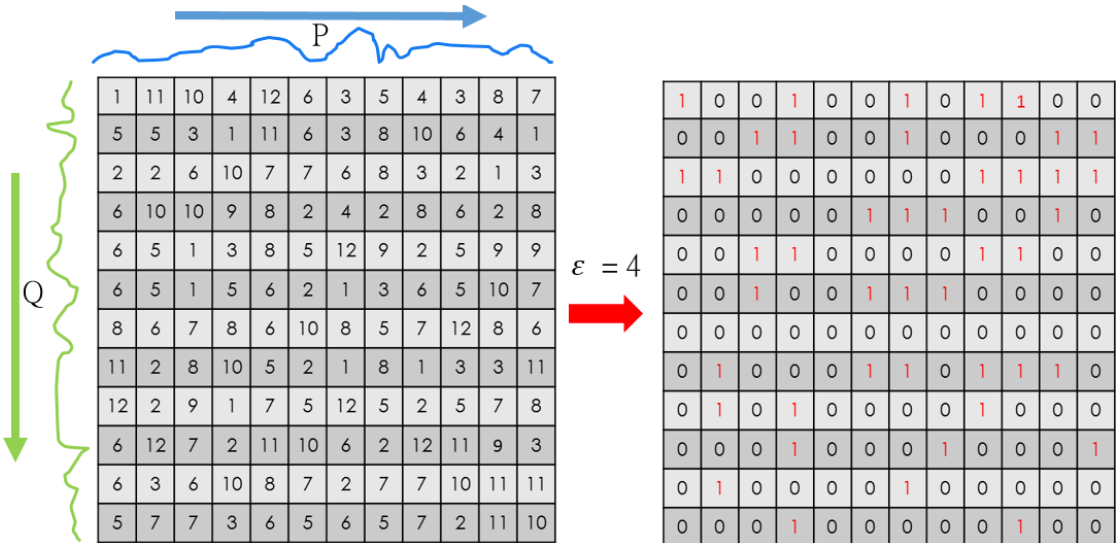


Figure 2: Given a threshold ε of 4, the 2-D distance matrix $\mathbb{D}$ (the left matrix) and the 2-D binary mask M (the right matrix) derived from two pitch vectors Q and D.

3.4. Minor Gap Filling

On the derived binary mask, we can find some entries of 1 connected as many paths. A path may start from a certain entry $M(i,j)$ and stop at another $M(i',j')$. We denote such a path as $o(u(i,j) \Rightarrow u'(i',j'))$, where u and u' are called nodes. Note that the entry of the mask on a node is always 1. Besides, a node (i,j) can have at most only three possible preceding nodes, i.e., $(i-1,j)$, $(i,j-1)$, $(i-1,j-1)$. For convenience, we also use the notations, $u.i$ and $u.j$, to represent the row index and column index of the node u, respectively. In physical meaning, each path actually corresponds a pair of potential plagiarised subsegments $q(i:i')$ and $d(j:j')$ in the two melodies, respectively. Hence, one important task of the proposed AMMPD method is to explore all potential connected paths over the binary mask M.

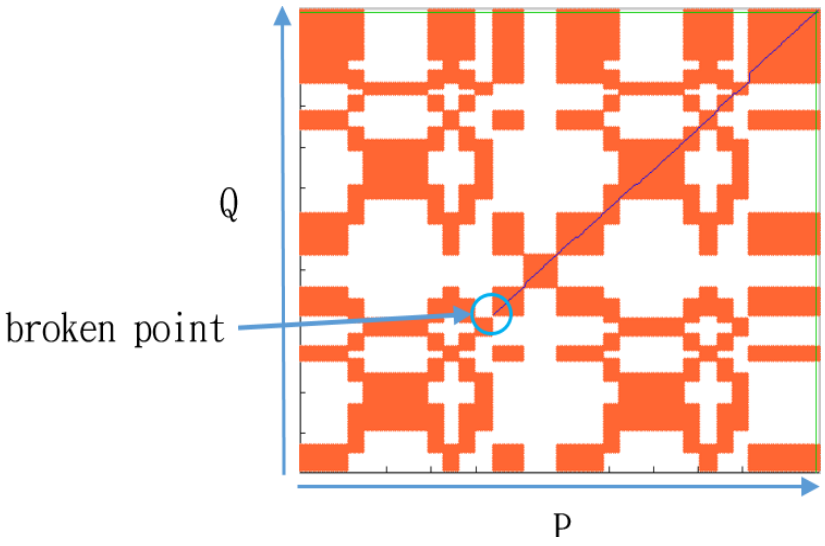
Since every path always breaks at an entry of 0 on M, the exploration of the paths can be done by finding rectangular blocks separated by horizontal and vertical gaps formed by 0-entries.

However, the separation of rectangular blocks are prone to the gaps formed by some singular nodes (i,j)'s with larger pairwise distances $\mathbb{D}(i,j)$'s incurred from noisy pitch elements. Like the binary mask shown in Fig. 3 (a), the red areas and white areas contain respectively 1-entries and 0-entries. A small gap appearing on the singular node of entry-0 highlighted with the blue circle breaks a path. As a result, the broken path may cut the true plagiarised subsegments, as shown in Fig. 3 (b), into shorter ones.

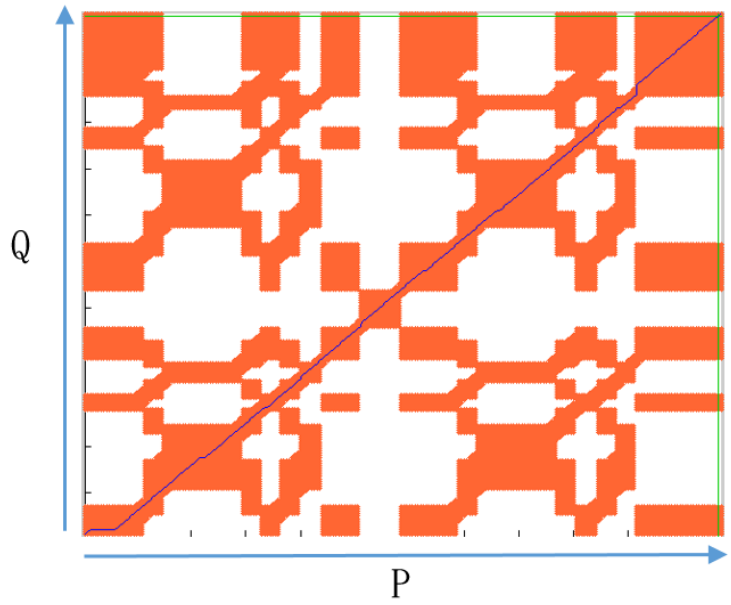
To avoid the bad break of paths, we perform a gap filling process to bridge the paths separated by the singular nodes. This filling process fills those entries between a node and the nearest node along the horizontal, the vertical, and the diagonal direction according to the following three rules:

- Horizontal Filling: $M(i+k,j)=1$ for $1 \le k \le k^* - 1 \le g$ if $M(i+k,j)=0 \wedge M(i+k^*,j)=1$;
- Vertical Filling: $M(i,j+k)=1$ for $1 \le k \le k^* - 1 \le g$ if $M(i,j+k)=0 \wedge M(i,j+k^*)=1$;
- Diagonal Filling: $M(i+k,j+k)=1$ for $1 \le k \le k^* - 1 \le g$ if $M(i+k,j+k)=0 \wedge M(i+k^*,j+k^*)=1$;

Note that the parameter g is set as the largest gap width allowed for the gap filling. The gaps with the width larger than g will not be filled. Taking the case in Fig. 3 (a) as an example, the resultant path explored after the gap filling is the one shown in Fig. 3 (b).



(a) the bad break point cuts a path into a shorter one.



(b) the longer path formed in the block without the bad break point.

Figure 3: Improper partitioning of blocks of caused by noisy pitch elements.

3.5. Recursive Block Partitioning

Some paths of connected 1-entries are not long enough to identify a pair of plagiarized subsegments in two melodies. To fast remove these paths before exploring the connected nodes in them, we exploit a recursive block partitioning procedure on the binary mask to extract all potential rectangular blocks that enclose the paths of connected entries of 1's.This block partitioning process is a recursive procedure which partitions the binary mask into blocks separated by horizontal and vertical gaps. Suppose that the mask **M** has m rows and and n columns. The partition procedure accepts four parameters, i.e., (b, t, l, r), which define respectively the top, bottom, left, and right boundaries of a rectangular area, denoted with $B(b{:}\,t, l{:}\,r)$, in the mask for the recursive block partitioning. In $B(b{:}\,t, l{:}\,r)$, the procedure first computes the row sums, $Sum_r(i)$ for $b+1 \le i \le t-1$, and column sums, $Sum_c(j)$ for $l+1 \le j \le r-1$, of the area, i.e.,

$$Sum_r(i) = \sum_{j=l}^{r} \mathbf{M}(i,j) \ \text{ for} i = b+1, \dots, t-1,$$

$$Sum_c(j) = \sum_{i=b}^{t} \mathbf{M}(i,j) \ \text{ for} j = l+1, \dots, r-1.$$

Clearly, a horizontal gap and a vertical gap in $B(t{:}\,b, l{:}\,r)$ would derive a row sum of zero and a columnsum of zero, respectively. We collect the indices of rows and columns that have zero sums as two sets of indices $G_r = \{row_i | Sum_r(row_i) = 0, i = 1, \dots, R, b+1 \le row_i \le t-1\}$ and $G_c = \{col_j | Sum_c(col_j) = 0, j = 1, \dots, C, l+1 \le col_j \le r-1\}$. Besides, the indices in both sets are sorted in the ascending order. If either of G_r and G_c is not empty, we can partition $B(b{:}\,t, l{:}\,r)$ into $(R+1) \times (C+1)$ smaller rectangular areas, say $B(b_{i,j}{:}\,t_{i,j}, l_{i,j}{:}\,r_{i,j})$ for $i = 1, \dots, R+1$ and $j = 1, \dots, C+1$, where

$$b_{i,j} = \begin{cases} b, & \text{if} i = 1, \\ row_{i-1}, & \text{if} i = 2, \dots, R+1, \end{cases}$$
$$t_{i,j} = \begin{cases} row_i, & \text{if} i = 1, \dots, R, \\ t, & \text{if} i = R+1, \end{cases}$$
$$l_{i,j} = \begin{cases} l, & \text{if} j = 1, \\ col_{j-1}, & \text{if} j = 2, \dots, C+1, \end{cases}$$
$$r_{i,j} = \begin{cases} col_j, & \text{if} j = 1, \dots, C, \\ r, & \text{if} j = C+1. \end{cases}$$

For each rectangular area, the partition procedure is performed recursively until the area cannot be further partitioned, i.e., $G_r = G_c = \phi$. For any rectangular area $B(b{:}\,t, l{:}\,r)$, its width, denoted with $Width(B)$, and height, denoted with $Height(B)$, are thus $(t - b + 1)$ and $(r - l + 1)$, respectively. The block B may enclose more than one path. The width and height of B set the upper bounds on the lengths of any pair of similar segments, say seg_P and seg_Q, in P and Q, respectively, which can be located inside B, i.e.,

$$Len(seg_P) \leq Width(B) \text{and} Len(seg_Q) \leq Height(B),$$

where $Len(seg)$ is the length of a segment seg in a pitch sequence. For any rectangular area B, if either $Width(B)$ or $Height(B)$ is too small, then the located pair of similar segments seg_P or seg_Q would be very short, implying that the located plagiarizing segments are unreasonable. Under such a situation, the rectangular area B can be discarded for subsequent path exploration. Removing the inadequate blocks for path exploration can significantly accelerate the plagiarism detection process. Therefore, we set a length threshold l and remove any candidate block B if $\min\{Width(B), Height(B)\} \leq l$. The threshold l can be set depending on how short a reasonable plagiarizing segment can be. The procedure $Partition()$ listed in Procedure 1 lists the steps of the recursive block partitioning. With the recursive procedure, the block partitioning begins by initializing the output set of blocks $\mathbb{B}$ as an empty set and then invoking the procedure with Partition$(1, m, 1, n, l)$ for partitioning a binary mask **M** of m rows and n columns.

Procedure 1 $Partition(b, t, l, r)$

Input: the rectangular block B corresponding to $M(bottom = b, top = t, left = l, right = r)$, where $l < r \wedge b < t$.
Output: a set of disjoint subblocks ($\mathbb{B}$) inside the block B

Compute the sum of each row in M covered by the block B by $S_R(i) = \sum_{j=l}^{r} M(i,j)$, for $b \leq i \leq t$
Compute the sum of each column in M covered by the block B by $S_C(j) = \sum_{j=b}^{t} M(i,j)$, for $l \leq j \leq r$
Set $G_R = \{i | S_R(i) = 0\} = \{g_r(1), g_r(2), \dots, g_r(N)\}$, where $g_r(i) < g_r(i+1)$ for $i = 1, 2, \dots, N-1$.
Set $G_C = \{j | S_C(j) = 0\} = \{g_c(1), g_c(2), \dots, g_c(M)\}$, where $g_c(j) < g_c(j+1)$ for $j = 1, 2, \dots, M-1$.
if $M = 1 \wedge N = 1$ **then return** $\mathbb{B} = \{B\}$
else
    for each $i = 1, \dots, N-1$ **do**
        for each $j = 1, \dots, M-1$ **do**
            $b \leftarrow g_r(i)$
            $t \leftarrow g_r(i+1)$
            $l \leftarrow g_c(j)$
            $r \leftarrow g_c(j+1)$
            $\mathbb{B} \leftarrow \mathbb{B} \bigcup Partition(b, t, l, r)$.

3.6. Path Exploration by Finding Detailed Alignment of Pitch Elements

Once the blocks inside the binary mask are available, the paths inside each block need to be explored. A path is a sequence of connected nodes. The starting node (i, j) is an open node that has no preceding nodes, i.e., $M(i-1, j) = M(i, j-1) = M(i-1, j-1) = 0$. In contrast, the

stopping node (i,j) of a path is a close node that has no succeeding nodes, i.e., $M(i+1,j) = M(i,j+1) = M(i+1.j+1) = 0$. Any node that is neither an open node nor a close node is an intermediate node. Hence, all nodes in a block form a directed acyclic graph (DAG). Every node corresponds to a graph node. A graph node (i,j) connects only to its preceding nodes $(i-1,j)$, $(i,j-1)$, and $(i-1,j-1)$ if the three nodes of 1-entries do exist (nodes do not exist for 0-entries). With the constructed DAG, all paths from an open node (u_o) to a close node (u_c) can be found by the procedure $Explore(u_o,u_c)$ listed in Procedure 2. Some notations are introduced in the procedure. First, $predecessor(u)$ and $successor(u)$ mean the direct succeeding nodes and the direct preceding nodes of a node u, respectively. The notation $c(u \rhd u')$ denotes a connection connecting the two nodes u and u'. The operator $c \oplus o$ denotes the concatenation of a connection c and a path o. To explore all paths inside a block, we just need to find all open nodes and close nodes and then make a call to the procedure $Explore(u_o,u_c)$ for each pair of open node u_o and close node u_c.

There might exist relations of dominance among the paths explored by the$Explore()$ procedure. A path $o(u \Rightarrow v)$ dominates another path $o'(u' \Rightarrow v')$ if $(u.i <= u'.i) \wedge (u.j <= u'.j) \wedge (v.i > v'.i) \wedge (v.j > v'.j)$. The physical meaning of the dominance is that the plagiarised subsegment implied by the path o contains that implied by the path o'. In case that a path o dominates another path o' in the set of explored paths, then o' is removed from the path set because it is redundant. As shown in Fig. 4, four paths are explored in an example block. The red circles highlight the starting nodes while the blue circles highlight the stopping nodes. Among the four paths, the two paths highlighted with blue and green colors share the same stopping node. It can be seen that the green path dominates the blue one according to the above condition of dominance, meaning that the blue path is redundant.

Procedure 2 $Explore(u_0,u_1)$

Input: a node u_0 and a node u_1 in the same block.
Output: a set of paths (O) starting from u_0 and stopping at u_1.

1: **if** $u_0 \in predecessor(u_1)$ **then**
2: **return** $O' = \{o(u_0 \Rightarrow u_1)\}$
3: **else if** $(u_o.i > u_1.i) \vee (u_0.j > u_1.j) \vee (successor(u_0) = \phi)$ **then**
4: **return** $O' = \phi$
5: **for** $u \in successor(u_0)$ **do**
6: $O' \leftarrow Explorer(u,u_1)$
7: $O = \phi$
8: **for** $o \in O'$ **do**
9: $O \leftarrow O \bigcup \{c(u_0 \rhd u) \oplus o\}$

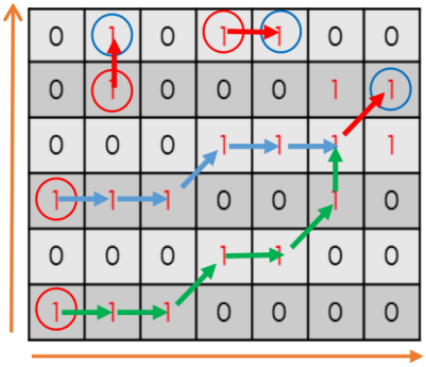


Figure 4: The starting nodes and stoppingnodes of paths in a candidate block..

3.7. Overlapping Path Removal

After removing all dominated paths from the set of explored paths, some paths may overlap with others. The overlapping paths may share some connected nodes, but do not dominate each other. As shown in Fig. 5, three paths share a large portion of the paths. However, these three paths identify three different pairs of plagiarized subsegments in the two melodies. Similarly, the three pairs of plagiarized subsegments also have a large part in common and thus may cause redundant detection of plagiarism. One reasonable action is to choose the best one among them. To do this, we need to define a criterion for assessing the goodness of a path. As every explored path contains a sequence of connected nodes $u(i,j)$, with each meaning a corresponding pair of pitch elements $(q(i), d(j))$. Along the path, every two neighboring nodes must be a node u and a successor node of the node u. Hence, a path can consist of three kinds of connections for any two neighboring nodes, including vertical, horizontal, and diagonal node connections. A vertical node connection $(i-1,j) \triangleright (i,j)$ implies a two-to-one correspondence of pitch elements, i.e., $\{q(i-1), q(i)\} \rightarrow \{d(j)\}$. A vertical node connection $(i,j-1) \triangleright (i,j)$ implies a one-to-two correspondence of pitch elements, i.e., $\{q(i)\} \rightarrow \{d(j-1), d(j)\}$. A diagonal node connection $(i-1,j-1) \triangleright (i,j)$ implies a one-to-one correspondence, i.e., $\{q(i-1)\} \rightarrow \{d(j-1)\}$ and $\{q(i)\} \rightarrow \{d(j)\}$. Obviously, the one-to-one correspondence gives a better explanation to the correspondence between elements of plagiarised subsegments. Therefore, a feasible index to assess the goodness of a path o the number of diagonal node connections, say $Diag(o)$, along the path. Based on the criterion, $Diag(o)$, the procedure to remove those overlapping paths with lower values of $Diag(o)$ is designed as the $RemoveOverlappingPaths(O,l)$ listed in Procedure 3.

Procedure 3 $RemoveOverlappingPaths(O,l)$

Input: the set of paths (O) in a block; the lower bound on the length (l) of the overlapping part between two overlapping paths

Output: the paths O' which contains no overlapping paths

```
for o(u ⇒ v) ∈ O do
    for o'(u' ⇒ v') ∈ O ∧ o' ≠ o do
        if Overlap(o, o') ≥ l then
            if Diag(o) > Diag(o') then
                O ← O − {o'}
            else
                O ← O − {o}
return O
```

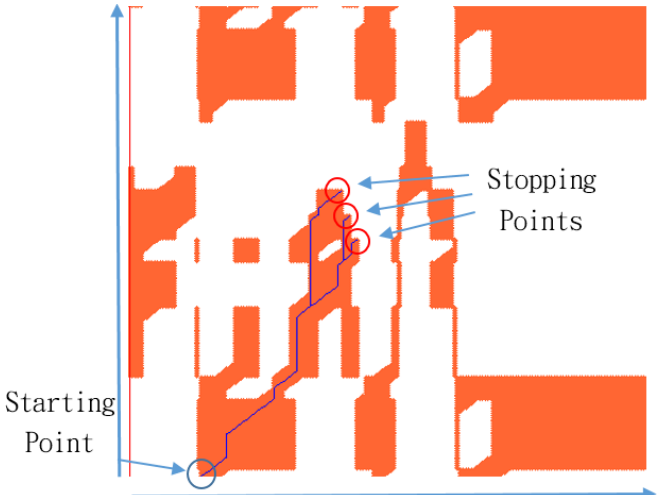


Figure 5: Example cases of overlapped paths.

3.8. Removal of Distorted Sections in Eligible Paths

Every path that remains after the overlapping path removal is an eligible path for identifying plagiarised subsegments between the two suspicious melodies. However, there may still exist one kind of oddness on the path. Taking Fig. 6 as an example, we can see that some nodes have one-to-many or many-to-one correspondences, as highlighted with blue circles. These one-to-many correspondences or many-to-one correspondences would cause longer sections of vertical node connections or horizontal node connections inside the path. We refer to this kind of sections as distorted sections which may not be appropriate to identify the pieces in plagiarised subsegments. Namely, we should remove these distorted sections from a path to prevent false detection of plagiarised pieces.

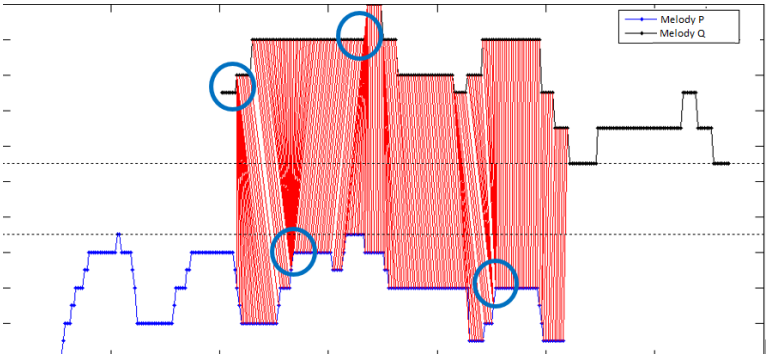
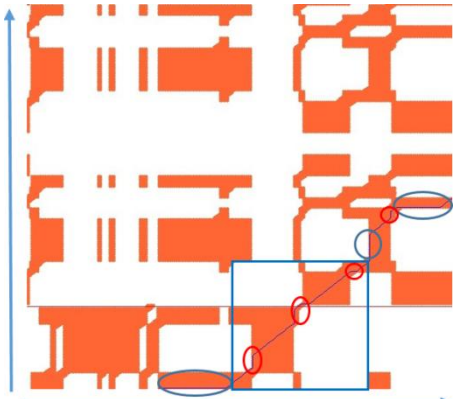
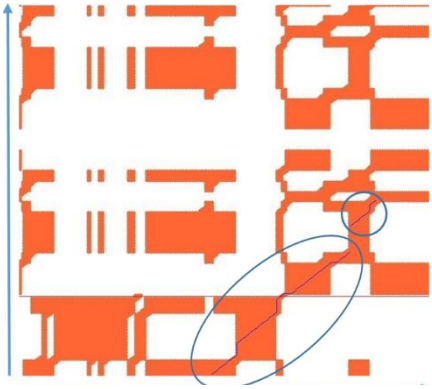


Figure 6: The alignment of the pitch elements between two melodies may contain unreasonable one-to-many and many-to-one correspondences.

As shown in Fig. 7(a), a long path may comprise several horizontal and vertical sections highlighted with circles. The distorted sections in blue circles are longer, while those in red circles are shorter. For the path illustrated in Fig.7(a), the true section corresponding to the plagiarised subsegments in the two melodies is the section enclosed by the large blue rectangle. To extract the non-distorted sections from an eligible path, our method first approximates the path with a polyline. Afterwards, the horizontal/vertical and near-horizontal/near-vertical segments in the polyline are removed. In doing so, a segment $\bar{o}$ in the polyline is removed if $|slope(\bar{o}) - 1| > \delta$, where $slope(\bar{o})$ is the slope of the segment $\bar{o}$ and δ is the threshold specified for the removal. Fig. 7(b) shows the result after applying the method to the path in Fig. 7(a). The result demonstrates that this method works fine to extract the desired non-distorted section from the path.



(a) The path with distorted sections



(b) The segmented paths after removing distorted sections

Figure 7: Examples of distorted sections in a path.

After removing the distorted sections in a path, some paths may become shorter and even dominated by others. These paths should be purged, too. According to the specified parameter l in block partitioning, we discard those paths with the lengths smaller than l. Additionally, as done at the end of path exploration, the path dominated by others are also discarded.

4. EXPERIMENTAL RESULTS

4.1. Data Set and Types of Experiments

Several experiments are conducted to evaluate the performance of the proposed method. We use the MIR-QBSH data set created by Chang et al [9] for the experiments. The data set contains 48 ground-truth MIDI files and 4431 singing/humming clips from about 195 subjects. Since the songs in the data set does not contain the plagiarising pieces, we randomly partition the 48 MIDIs into 16 groups, with each having three. Then, the three MIDIs in each group are concatenated as on MIDI to get 16 concatenated MIDI songs. Besides, we also randomly compose 100 concatenated voiced songs from 300 signing/humming clips which enclose all melodies of the 48 MIDIs sung/hummed by at least 6 different subjects. Each concatenated voiced song also has three different signing/humming clips. Three types of experiments are conducted on the composed data set. Type A experiments use one concatenated voiced MIDI song as the input and search the possible plagiarising pieces in all concatenated voiced songs of the database. Type B experiments use one single MIDI song (not a concatenated MIDI song) as the input and search the possible plagiarism in all concatenated voiced songs of the database. Finally, Type C experiments use a single MIDI song as the input and search the similar single signing/humming clips. The indices for performance evaluation are the precision rate and the recall rate which are evaluated by

$$PrecisionRate(\mathbf{P}) = \frac{\#\ ofCorrectDetection}{\#\ ofDetection} \times 100\%;$$

$$RecallRate\ \ (\mathbf{R}) = \frac{\#\ ofCorrectDetection}{\#\ ofGroundTruths} \times 100\%.$$

A correct detection of a plagiarising subsegment requires that the following condition hold true for the detected subsegment d and the true plagiarising subsegments in two melodies a and b:

$$\left(\frac{len\,(d \cap a)}{len\,(a)} \geq \frac{1}{3}\right) \wedge \left(\frac{len\,(d \cap b)}{len\,(b)} \geq \frac{1}{3}\right),$$

where $len(s)$ denotes the length of a subsegment s.

4.2. Overall Performance Evaluation

The performance of the proposed method is evaluated from the results of the three types of experiments. Table 1 lists the averaged precision rate and recall rate for these three types of experiments. The achieved precision rate is about 84% and the recall rate is about 63%. Because of the inaccuracy of pitch extraction, the recall rates are much lower than the precision rates. The noisy pitch vectors can cause the mismatch of many subsegments with similar melodies in suspicious music audio. The median filtering and block gap filling proposed in the method seem take only minor effect in handling the noisy pitch features. Particularly, the query input, which is a MIDI clip, and the database targets, which are humming/singing clips, are different audio source which reveal different audio characteristics. Moreover, as the pitch of a humming/singing clip varies with different subjects, the pitch vectors extracted from the audio clips of the same melody still may differ significantly for different subjects.

Table 1. The overall precision rates and recall rates for the three types of experiments. X_p and X_r denote respectively the precision rate and the recall rate of Type X experiments.

A_p(%)	A_r(%)	B_p(%)	B_r(%)	C_p(%)	C_r(%)
82	64	83	63	87	63

4.3.Effects of Different Parameters

There are some parameters introduced in the proposed method, including

- ε: the upper bound on the pairwise distances of pitch elements in two suspicious melodies;
- l: the lower bound on the length of an eligible candidate path for locating the plagiarising subsegments in two melodies;
- g: the parameter defining the upper bound on the minor gaps to be filled between blocks on the binary mask M;
- δ: the slope parameter used for removing the distorted sections along an eligible path.

In the following, the effects of these parameters are examined and discussed with the experimental results.

4.3.1.The Effect of ε

The parameter ε affects mainly the number of blocks formed on the binary mask M. Table 2 shows the precision rates and recall rates corresponding to the different settings of ε, i.e., ε=0, 1, and 2. All other parameters are set as m=5, l=150, g=3, and δ=0.4. The extremal case of $\varepsilon = 0$ causes that two suspicious melodies are detected as plagiarism

only when they are exactly identical. The results show that a higher setting can increase the recall rate because it allows larger pitch differences between two plagiarising subsegments. However, loosened condition of plagiarism may also increases the number of false alarms and thus decreases the precision rate.

Table 2: The precision rates and recall rates for different setting of ε. X_p and X_r denote respectively the precision rate and the recall rate of Type X experiments.

ε	A_p(%)	A_r(%)	B_p(%)	B_r(%)	C_p(%)	C_r(%)
0	100	0	100	0	100	0
1	82	64	83	63	87	63
2	27	80	31	79	35	79

4.3.2. The Effect of l

The parameter l specifies at least how long a true plagiarisinig subsegement must be. We test l with the different values 120, 130, 140, 150, and 160. Note that a length of 150 pitch elements is about 150*0.032 (=4.8) seconds. Table 3 lists the precision rates and recall rates for these different settings. The larger the l, the longer the plagiarising subsegments must be. Therefore, some shorter plagiarising subsegments are more likely to be missed and the recall rate tends to decrease. On the other hand, some shorter false alarms can be avoided and thus the precision rate tends to increase.

Table 3: The precision rates and recall rates for different setting of l.

l	A_p(%)	A_r(%)	B_p(%)	B_r(%)	C_p(%)	C_r(%)
120	33	76	36	76	41	76
130	51	74	54	73	59	72
140	67	69	70	69	74	69
150	82	64	83	63	87	63
160	89	60	90	59	92	59

4.3.3. The Effect of g

The parameter g defines the upper bound on the minor gaps between blocks on the binary mask M. To prevent the binary mask being over-partitioned into many small blocks, the proposed method fills the minor gaps between blocks. A larger value of g causes more blocks connected through the gap filling process, thus usually resulting in more detected plagiarising subsegments. Though the increased detection of plagiarism improves the recall rate, the precision rate may decrease due to the increased number of false alarms. The precision rates and the recall rates for different settings of g are shown in Table 4. The results show that an appropriate value for the parameter g ranges between 3 and 5.

Table 4: The precision rates and recall rates for different setting of g.

g	A_p(%)	A_r(%)	B_p(%)	B_r(%)	C_p(%)	C_r(%)
0	88	38	88	38	90	38
1	85	49	86	49	89	49
2	84	54	86	54	89	54
3	82	64	83	63	87	63
4	79	69	80	69	86	69
5	72	74	74	73	81	73
6	66	76	70	75	77	74
7	59	76	63	75	71	75

4.3.4 The Effect of δ

The parameter δ specifies the deviation of the slope of a path from a 45° diagonal line. A large deviation indicate that the path is a distorted path which contains improper alignment of pitch elements and should be discarded. Hence we experiment the proposed method with the settings of δ ranging between 0.1 and 0.8. The smaller the δ is, the better element correspondences an eligible path may derive, meaning that a better precision rate we may achieve. However, the side effect is that some true plagiarising subsegments with slightly distorted correspondences may be missed. Table 5 shows the precision rates and recall rates for the different settings of δ.

Table 5: The precision rates and recall rates for different setting of δ.

δ	A_p(%)	A_r(%)	B_p(%)	B_r(%)	C_p(%)	C_r(%)
0.1	89	41	88	41	89	41
0.2	88	53	88	53	89	53
0.3	86	60	87	59	90	59
0.4	82	64	83	63	87	63
0.5	74	65	76	65	80	64
0.6	65	67	69	66	73	65
0.7	52	69	57	67	63	66
0.8	36	70	41	68	49	67

4.4.Speed Evaluation

The time taken for examining a pair of suspicious songs depends on how many similar parts appear in this pair of songs. If the two songs contain more isolated similar parts, then more blocks would be formed on the binary image mask and thus require more time to explore paths. In our Type-A experiments, the average time to detect one pair of plagiarizing segments in two 24-second songs is about 6 seconds. For the Type-B experiments, the average time to examine a 8-second song and a 24-second song is about 2 seconds. One major factor that slows down the detection is our brute-force approach to remove the possible pitch shift between two suspicious songs. To deal with the different pitch levelsof two suspicious songs, we exhaustively shift the pich of one song within a range to find the best match between the two songs.Therefore, for two pitch sequences with the pitch values falling in the range $[low_1, high_1]$ and $[low_2, high_2]$, respectively,

then maximum number of iterations for the pitch shifting would be $(high_2 + high_1 - low_2 - low_1 + 2)$. Hence, the detection process takes longer for songs with larger pitch ranges.

From the time statistics obtained from our experiments, the average fraction of time consumed by various procedures in the proposed method are listed in Table 6. The procedures near the ending phase of the detection take less time because the number of candidate paths gets fewer. The path exploration takes much time in collecting the source nodes of each non-zero entry.

Table 6: Time fraction of various procedures in the plagiarism detection method.

Procedure	Time Fraction (%)
Recursive Block Partitioning	22
Path Exploration	55
Overlapping Path Removal	12
Removal of Distorted Sections	11

3. CONCLUDING REMARKS

The paper presents a novel method to address the problem of AMMPD. The novelty originates from the tackling of the challenging issues of detecting and locating plagiarised subsegments between two suspicious music melodies through an image-based approach in spatial domain. The proposed method performs mainly the path exploration over a 2-D binary mask. The key contribution of the proposed method is a feasible solution to the two major issues in AMMPD, including (1) the inexact matching of noisy and inaccurate pitches of music audio and (2) the fast detection and positioning of similar subsegments between suspicious music audio. In fact, the proposed method is applicable to not only the applications of AMMPD, but also the applications of querying by humming/singing (QBSH).

ACKNOWLEDGEMENTS

This paper is supported by the Ministry of Science and Technology, Taiwan, under the project with the grant numbers 105-2221-E-259 -029 -.

REFERENCES

[1] Chung-Che Wang, Jyh-Shing Roger Jang, and Wennen Wang, "An improved query by singing/humming systemusing melody and lyrics information.," in ISMIR. Citeseer, 2010, pp. 45–50.

[2] Hung-Ming Yu, Wei-Ho Tsai, and Hsin-Min Wang, "A query-by-singing system for retrieving karaoke music,"IEEE Transactions on multimedia, vol. 10, no. 8, pp. 1626–1637, 2008.

[3] Yunjing Wang, "Similarity matching method for music melody retrieval," Journal of Multimedia, vol. 8, no. 4, pp.386–393, 2013.

[4] Donald J Berndt and James Clifford, "Using dynamic time warping to find patterns in time series." in KDDworkshop. Seattle, WA, 1994, vol. 10, pp. 359–370.

[5] Vassilis Athitsos, Panagiotis Papapetrou, Michalis Potamias, George Kollios, and Dimitrios Gunopulos, "Approximate embedding-based subsequence matching of time series," in Proceedings of the 2008 ACM SIGMOD International Conference on Management of data. ACM, 2008, pp. 365–378.

[6] Mike Paterson and Vlado Danˇ c´ ık, "Longest common subsequences," in International Symposium on MathematicalFoundations of Computer Science. Springer, 1994, pp. 127–142.

[7] Hwei-Jen Lin, Hung-Hsuan Wu, and Chun-Wei Wang, "Music matching based on rough longest common subsequence.," J. Inf. Sci. Eng., vol. 27, no. 1, pp. 95–110, 2011.

[8] Li Hui, Bei-qian Dai, and Lu Wei, "A pitch detection algorithm based on amdf and acf," in 2006 IEEE InternationalConference on Acoustics Speech and Signal Processing Proceedings. IEEE, 2006, vol. 1, pp. I–I.

[9] Roger Chang, "MIR-QBSH dataset," http://mirlab.org/dataSet/public/.

Authors

Cheng-Chin Chiang received his Ph.D. degree in the Department of Computer Science and Information Engineering from National Chiao Tung University in Taiwan in 1993. He is now a professor at the Department of Computer Science and Information Engineering in National Dong Hwa University. His research interests include neural networks, pattern recognition, machine learning, multimedia processing and analysis, and virtual reality and augmented reality.

Mu-Syuan Sie received his Master degree in the Department of Computer Science and Information Engineering from National Dong Hwa University in Taiwan in 2015. He is now a multimedia system software engineer in MediaTek Inc. His research interests include human-machine interactions, machine learning, pattern recognition, and multimedia retrieval.

Hsiu-Chun Yang is pursuing his Master degree in the Department of Computer Science and Information Engineering from National Dong Hwa University. His research interests include machine learning, pattern recognition, and musical informationprocessing and retrieval.

Yi-Le Liu is pursuing his Master degree in the Department of Computer Science and Information Engineering from National Dong Hwa University. His research interests include machine learning, neural networks, pattern recognition, and content-based image retrieval.

INTENTIONAL BLANK