Chapter 5

Melodic similarity using COSFIRE filters

Abstract

The identification of repeating patterns in time signals - also known as motifs - has been a challenge for many years in the area of digital signal processing. For instance, in music, existing algorithms that detect similar motifs typically require multi-dimensional features such as chromagrams, similarity matrices and the use of the computationally expensive dynamic time warping (DTW) algorithm. We propose a time-scale and transposition invariant method for identifying repeating motifs and demonstrate its effectiveness in musical signals. We adapt the COSFIRE approach, which has been found effective in 2D signals, to 1D signals. Our method has a computational time complexity substantially lower than DTW and achieves better effectiveness compared to DTW, Symbolic Aggregate Approximation and cross correlation results. The proposed 1D COSFIRE approach is highly effective and efficient for extracting a symbolic representation of the melodies in a given song. Additionally, it is conceptually simple and versatile, in that it can be applied to any 1D signal.

5.1 Introduction

The identification of similar or identical sub-sequences that appear in different positions in 1D time series, has been a task for research in the fields of digital signal processing and pattern recognition for the last two decades. Such sub-sequences are typically called motifs. Computational methods have been applied in medical and biological signals (Abe and Yamaguchi 2005) and for tasks such as weather prediction (McGovern et al. 2007), speech recognition (Payne 2006) and music (Müller et al. 2011, Eronen 2007). In this work we propose a novel computational method to identify similar 1D patterns and demonstrate its effectiveness in musical signals. In the following, we therefore focus on the problems and literature relating to music pattern recognition.

In music theory, a motif can be defined as the smallest melody with an important thematic identity. Such a pattern can be a defining feature of a musical piece — think for instance of the sequence A-G-F-E-D-C♯-D in Toccata and Fugue in D minor by J.
Melodic similarity using COSFIRE filters

S. Bach – and thus be of central importance in the context of intellectual property. The verse of a song is an example of such a repeating pattern in popular music.

The term *melody* can be primarily defined as a sequence of musical tones and silences of specific duration over time. In a first approximation, a melody can be described mathematically by a sequence of fundamental frequencies over time called the *pitch track*, derived from the audio signal. The term *pitch* refers to a perceptual property of a musical tone and is closely related to the fundamental frequency of that tone. The complexity of the relation between the pitch and the fundamental frequency has been described in (Butler 1989, Howell et al. 1991).

The identification of motifs leads to an explicit understanding of the musical structure of a piece. The detection of such patterns is an essential step in musicological analysis. In Fig. 5.1, we illustrate an example from a data set that we compiled for this work. Fig. 5.1a shows the values of the microphone signal of a song. In Fig. 5.1b, we show a representation of the melody short-term fundamental frequency over time. In this case four motifs are present. The boundaries of the motifs are indicated with vertical lines. Motif 3 is a repetition of motif 1 and they are both indicated with the letter “A”. Similarly, the letter “B” is given to the motif 2 and motif 4. The musical structure of the song in Fig. 5.1 has the form “A” - “B”.

In this work we focus on monophonic songs of Eastern Mediterranean folk music. There are various differences between popular and folk music which have to be taken into account in computational analysis. Popular music is typically recorded in professional studios with high quality equipment and professional musicians. In folk music, non-professional performers may perform singing with rhythmic variations and pitch drifts. However, generally, it is difficult to determine whether a performer (professional or non-professional) is being expressive or hiding a mistake. In addition, recordings of folk music are often made in noisy environments. From a technical point of view, the majority of the methods for similar problems are using Musical Instrument Digital Interface (MIDI) information in their methodology. Folk music lacks from transcriptions since many times the composer is not known and therefore similar melodies can be found in different songs. Therefore, in the computational analysis of folk music it is preferable that the MIDI information is avoided.

In this chapter we propose a method to identify repeating patterns using a single feature, the short-time fundamental frequency, as a function of time. This method can be applied to monophonic (i.e. with a single melody) vocal folk tunes. It is based on a modification of the COSFIRE (Combination Of Shifted Filter Responses) approach (Azzopardi and Petkov 2013), a technique that has been introduced for trainable visual pattern recognition. So far, the COSFIRE method has been applied in two-dimensional signals. In this work, we adapt the COSFIRE approach to 1D

---

1 Data set can be downloaded from [https://www.cs.ucy.ac.cy/projects/folk/](https://www.cs.ucy.ac.cy/projects/folk/)
5.1. Introduction

Figure 5.1: Audio signal represented as (a) obtained by a microphone and as (b) a sequence of fundamental frequencies coming from a short-term frequency analysis. In this example, four motifs are present, whose starting and ending positions are indicated by the vertical lines.

signals. We compare our results with other distance metrics that are widely used in pattern recognition of 1D problems.

5.1.1 Related work

The majority of the applications in Music Information Retrieval (MIR) require the extraction of low-level features derived from the raw audio signal. There has been a significant amount of work in the research of audio feature representation (Peeters and Rodet 2004, McKinney and Breebaart 2003).

Depending on the application, different groups of features may be used. Temporal features include the autocorrelation coefficients or the note duration. Spectral information can be described by Fourier Transform (FT), spectral spread, spectral centroid, Mel frequency cepstral coefficients (Zheng et al. 2001). These features are usually used for other tasks such as instrument identification (Benetos
5. Melodic similarity using COSFIRE filters


In computational analysis of music structure, several methods have been proposed for the automatic identification and segmentation of important musical parts. These are often called audio thumbnailing (Bartsch and Wakefield 2005). Such parts include the introduction, the verse and the chorus. Other methods require note segmentation and the identification of repeating patterns is eventually achieved by creating a symbolic representation of the melody and the use of chromagrams (Bartsch and Wakefield 2005, Gómez 2006, Goto 2006, Müller et al. 2011). The studies in (Qi et al. 2007, Sandler and Aucouturier 2001) include audio segmentation using self-similarity measures and hidden Markov models.

In (Bartsch and Wakefield 2005) the authors propose a method for audio thumbnailing of popular music. Their method includes a frame-based audio segmentation where each audio frame is described with a set of features. Then, similarity matrices of the extracted features are computed and used to identify motif repetitions in the entire signal. The most common feature used for audio thumbnailing is the 12-dimensional chromagram proposed in (Gómez 2006).

Similar approaches to audio thumbnailing and automated identification of repeating musical aspects are described in (Goto 2006, Müller et al. 2011, Eronen 2007). The majority of those papers report the use of Dynamic Time Warping (DTW) in their methodologies. The extraction of chromagrams, the computation of similarity matrices and the DTW result in a complex system with a high computational cost. The use of low-level features for audio thumbnailing is explained in (Peiszer 2007, Aucouturier and Sandler 2002).

Other studies extract information from MIDI scores and correspond the musical sequence with letters of the alphabet (Crochemore et al. 2001, Hillewaere et al. 2009). Examples of such information include the duration and the fundamental frequency of the notes, the frequency difference between the previous, current and subsequent notes, temporal onsets, and others. The musical patterns and repetitions are identified with the use of string methods such as n-gram models.

In the last two decades, the query-by-humming (QBH) method has been widely used for melodic similarity (Song et al. 2002, Hu et al. 2003, Zhu and Shasha 2003, Dannenberg et al. 2004, Dannenberg and Hu 2004, Ryynänen and Klapuri 2008, Huq et al. 2010, Kotsifakos et al. 2012). Commercial systems such as Shazam, MiDomí, musipedia and SoundHound as well as academic systems such as Tunebot are examples of applications that use the QBH method. The procedure is as follow: the audio signal is first converted into a MIDI information. This step requires pitch track extraction and note segmentation that are still a challenging problem in the
5.1. Introduction

MIR. Then, the symbolic representation of the input signal is compared with a large set of songs in a database. In each comparison, a similarity value is returned. The best match between the input signal and a song in the database is chosen as the pair with the lowest similarity value.

The constraint of the methodologies discussed above, is that the MIDI scores have to be known. In the majority of the folk songs a MIDI score is not available, and therefore these methods cannot be applied. Another limitation of this methodology is the inability of analysing non-Western music. This arises from the fact that many cultures do not have a written language for describing their music, therefore a protocol such as MIDI is not available.

Another approach that is reported in the literature for motif recognition and classification is the use of wavelets (Velarde et al. 2013, Jeon et al. 2009). The melody is first represented with a single-scale signal that is derived from the continuous Haar wavelet transform. Then, wavelet coefficients are used to form feature vectors and processed for classification purposes with standard machine learning tools.

The Symbolic Aggregate Approximation (SAX) is a method that converts a time series into a symbolic representation (Keogh et al. 2006). Its major novelty that differentiates it with other established methods is that it allows for dimensionality reduction and the ability of applying distance measures on the symbolic representation. In (Lin et al. 2007) it is shown that this method had been applied for several types of signals such as the analysis of protein unfolding data (Ferreira et al. 2006), meteorological data (McGovern et al. 2006), telemedicine time series (Duchene et al. 2005) and others.

While various methods for automatic segmentation into meaningful musical sections have been proposed for Western and popular music, little work has been done on folk music (Dutta and Murthy 2014, Ishwar et al. 2013, Ross et al. 2012, Volk and Van Kranenburg 2012, Rao et al. 2014). However, the data sets used in the aforementioned studies are not available, or they are encrypted with the MIDI protocol that is not appropriate for the application at hand.

There are essential differences between these two categories. The structure of the songs includes well defined repetitions with constant tempo and rhythm. In contrast, folk music is produced by non-professional performers and most of the times these artists do not have formal musical education. This results in non-professional performances, such as singing out of tune or rhythmical variances throughout the song. Furthermore, the environmental conditions include low quality recordings in noisy public places. Existing methods for the automatic identification of repeating patterns are not equipped to deal with the mentioned aspects of folk music.

This chapter is structured as follows. In Section 5.2 we present our proposed method, while in Section 5.3 we demonstrate its effectiveness and compare it with existing methods on a new data set of monophonic folk songs. In Section 5.4 we discuss certain aspects of the proposed approach and then we draw conclusions in
5.2 Methods

5.2.1 Overview

In Fig. 5.2 we illustrate the whole pipeline of our method. It consists of a configuration and an application stage.

In the configuration, we first divide the audio signal into overlapping frames of 30ms in length with 3ms overlap. For each such a frame, we extract the fundamental frequency (Fig. 5.1b) using the YIN algorithm (de Cheveigné and Kawahara 2002) as explained further in Section 5.2.2. Then, we manually segment the audio signal into motifs as shown in Fig. 5.1b. After the segmentation is done, we take the first motif which we denote by the letter “A” and use it to configure a set of COSFIRE filters that describes its properties.

In the application stage we apply the configured set of COSFIRE filters to the remaining motifs of the audio signal. In order to come to a single value for every motif, we combine the responses of the set of COSFIRE filters by geometric mean. We label with the letter “A” every motif that evokes a combined response that is higher than a threshold, which is set experimentally. In the next iteration, we denote by “B” the next motif that has not been labelled and use it to configure a new set of COSFIRE filters. We then give the label “B” to motifs which exhibit responses higher than some threshold. We repeat this procedure until all motifs in a given audio signal are labelled.

These steps are explained in more detail in the sections below.

5.2.2 Fundamental Frequency Extraction

The fundamental frequency is closely related to the pitch of a musical note. The melody can be represented and modelled as the pitch in a function of time. The fundamental frequency estimation for musical signals has been an important and challenging task in the research of music information retrieval. Several algorithms and techniques have been proposed, some of them based on the time domain (Medan et al. 1991, Talkin 1995, de Cheveigné and Kawahara 2002) and others based on the frequency domain (Klapuri 2004, Dorken and Nawab 1994). We employed the widely used YIN algorithm proposed by Cheveign and Kawahara (de Cheveigné and Kawahara 2002) for the extraction of the fundamental frequency vector. The YIN algorithm takes as input the raw observations of the sound pressure and outputs a vector with fundamental frequency candidates for each audio frame. The algorithm is based on the autocorrelation function (ACF) of the audio signal where a number of modifications and corrections are applied after the autocorrelation, as
5.2. Methods

### Configuration

1. Audio signal
2. Transform to fundamental frequency
3. Manual segmentation of motifs
4. Split frequency signal into overlapping frames
5. Configure a COSFIRE filter with each frame

### Application

1. Test motif
2. Transform to fundamental frequency
3. Apply COSFIRE filter(s)
4. Combine COSFIRE responses
5. Symbolic representation

![Figure 5.2: The main steps of the proposed methodology.](image)

reported in (de Cheveigné and Kawahara 2002). The algorithm uses a frame-based approach and it gives the fundamental frequency candidates for a sequence of overlapped audio frames. For our analysis we used 1470 bins with a window length of 30ms and 128 bins for hop size (3ms).

#### 5.2.3 Configuration and response of a COSFIRE filter

A COSFIRE filter is configured by the values at certain positions in a given training motif (also referred to as a prototype signal). We consider a regularly spaced set of \( n \) points along the given signal where the midpoint of this set lies on the center of the prototype. In Fig. 5.3a, we present a synthetic signal consisting of five different patterns. The first pattern shows a sinusoidal signal that we use as a prototype along which we consider seven points centered around the zero crossing. We describe each point \( i \) by a pair \((f_i, t_i)\), where \( f_i \) is the value of the signal at time point \( t_i \) with respect to the center of the filter support (labeled by the * marker). We denote by \( P_c \) a COSFIRE filter that is defined as a set of such pairs:

\[
P_c = \{(f_i, \rho_i) \mid i = 1 \ldots n \}
\]  

(5.1)

where \( \rho_i = \delta(i - (n + 1)/2) \), \( n \) is the total number of considered time points and \( \delta \) is the length of the interval between the time points.
Melodic similarity using COSFIRE filters

For the given prototype shown in Fig. 5.3 we use \( n = 7 \) time points uniformly distributed in intervals of \( \delta = 10 \) time points to configure a COSFIRE filter with parameter values specified by the pairs in the following set:

\[
P_c = \left\{(f_1 = 0.87, \rho_1 = -30), \right. \\
\left. (f_2 = 0.95, \rho_2 = -20), \right. \\
\left. (f_3 = 0.58, \rho_3 = -10), \right. \\
\left. (f_4 = 0, \rho_4 = 0), \right. \\
\left. (f_5 = -0.58, \rho_5 = 10), \right. \\
\left. (f_6 = -0.95, \rho_6 = 20), \right. \\
\left. (f_7 = -0.87, \rho_7 = 30) \right\}
\]

The response of a COSFIRE filter is computed by first comparing the values in a given test signal to the preferred ones defined by \( P_c \). Thus, we obtain a similarity value for each pair in \( P_c \). Then we use a smoothing function to allow for some temporal tolerance and finally we combine all the smoothed similarity values with geometric mean.

**Similarity measure**

For a given point in time \( t \) of a test signal \( T \) we use a Gaussian kernel function to compute a similarity value for each pair in set \( P_c \) that defines a COSFIRE filter:

\[
D_i(t) = \exp \left(-\frac{(f_i - T_{t+\rho_i})^2}{2\sigma^2}\right), \quad \sigma = \sigma_0 + \alpha(|\rho|) \tag{5.2}
\]

where \( f_i \) is the preferred value of the \( i \)-th pair in set \( P_c \), and \( T_{t+\rho_i} \) is the corresponding value in the concerned neighbourhood of a signal \( T \) at time \( t \). The standard deviation \( \sigma \) of the Gaussian kernel function increases linearly with increasing distance from the center of the filter. In this way we allow more tolerance to the values of time points that are on the periphery of the support of the filter than those that are closer to the support center.

In Fig. 5.4 we present an example of the prototype explained in Section 5.2.3 that comprises seven points. We use vertical lines to indicate the seven positions of the time points that are considered in this COSFIRE filter. Their heights indicate the standard deviation \( \sigma \) that we use in Eq. (5.2) for \( \sigma_0 = 0.1 \) and \( \alpha = 0.9 \).

**Temporal tolerance**

In order to allow for temporal tolerance we smooth the similarity signals \( D_i \) of each pair of a COSFIRE filter. We denote by \( S_i'(t) \) a smoothing function that we com-
5.2. Methods

- Input 1D signal
  - Prototype
  - Random pattern
  - Noisy sine wave (SNR: 10 dB)
  - Noisy sine wave (SNR: 30 dB)
  - Amplitude (50%)
  - SNR: 10 dB

Figure 5.3: (a) Input 1D signal. The enframed sinusoidal signal is used as a prototype to configure a COSFIRE filter. (b) The responses of the filter to the input signal. The highest response is achieved for the prototype signal, but strong responses are also recorded at the center of similar but noisy patterns. Negligible response is achieved for the random pattern.
5. Melodic similarity using COSFIRE filters

The value of the standard deviation increases for each point \( i \) away from the support center of the filter.

Figure 5.4: The value of the standard deviation increases for each point \( i \) away from the support center of the filter.

In Fig. 5.4, we use the same prototype that we used in the example in Fig. 5.3a to configure a COSFIRE filter with \( n = 100, \delta = 1, \sigma_0 = 0.01 \) and \( \alpha = 0.001 \). The test signal is a noisy sinusoidal signal similar to the prototype but with a lower frequency. We use this example to illustrate the effectiveness of the smoothing function to achieve temporal tolerance. The image in Fig. 5.5b is a matrix of \( (n =) 100 \) similarity signals, one for each pair in the set \( P_c \) of the concerned COSFIRE filter. Due to the boundary effect, we do not compute similarity values in the beginning and end parts of the input signal. In Fig. 5.5c, we show the smoothed similarity signals.
5.2. Methods

Input 1D signal

(a)

Amplitude

0

-1

1

2

Similarity signals for each pair in a COSFIRE filter

(b)

n = 100

Geometric mean of similarity signals

(c)

Responses

0

0.5

1

Geometric mean of smoothed similarity signals

(d)

Smoothed similarity signals

n = 100

Responses

0

0.5

1

Time (s)

(e)

Figure 5.5: a) 1D input signal. The sinusoidal signal on the left is used to configure a COSFIRE filter with parameters \( n = 100, \delta = 1, \sigma_0 = 0.01 \) and \( \alpha = 0.001 \). The arrow indicates its support center. b) The similarity signals between the preferred values of the COSFIRE filter and the values of the input signal. c) The geometric mean of the similarity signals. d) The smoothed similarity signals. e) The COSFIRE filter responses computed as the column-wise geometric from the matrix in (d).
Response

We denote by $R(t)$ the response of a COSFIRE filter (Fig. 5.5c) at time $t$ that is computed as the geometric mean of all the involved smoothed similarity values:

$$R(t) = \left( \prod_{i=1}^{n} S_i'(t) \right)^\frac{1}{n}$$  \hspace{1cm} (5.4)

Configuration of one COSFIRE filter

In Fig. 5.6a we present an example of the audio signal of the song #22 from our data set, which consists of four motifs. The start and end positions of the vocal pauses are manually annotated with vertical lines. We connect consecutive motifs using linear interpolation in order to fill in the gaps with real numbers and thus avoiding numerical errors.

In this example, the first motif is similar to the third and the second motif to the fourth. All of the four motifs are similar in their first half. This is common in such singing data and it is important that the methods for the identification of melodic repetitions identify them as dissimilar, since the second half is different. For instance, in the example shown in Fig. 5.6a, the first two motifs should be classified as dissimilar. The AND-type character of the COSFIRE filters allows the discrimination of two motifs even when they are similar in their first half. The product of the low similarities in the second half of the motifs and the high similarities of the first halve results in a low COSFIRE response. As a result, the two signals will not be considered similar.

We use the training motif in Fig. 5.6a to construct one COSFIRE filter. In this example we use the fundamental frequency and the time positions of the prototype with $n = 1436$ points, $\sigma_0 = 0.05$ and $\alpha = 0.01$ for the configuration. Then we apply this COSFIRE filter to the entire input signal that consists of four motifs. In Fig. 5.6c we plot the response of the COSFIRE filter for each point of the input signal. The highest response is located in the middle of the training motif with a value of 1. For this and in the remaining parts of the signal, the local maxima are marked with dark spots. The second local maximum response is located in the middle of the third motif. Significantly lower local maxima appear for motif 2 and 4. This is due to the similarity of the motifs only in the first half.

Configuration of multiple COSFIRE filters

As an alternative to the above single filter approach, we split the prototype signal into small parts of equal length and configure a COSFIRE filter by each such part. This is inspired by the hierarchical object recognition approach proposed in
5.2. Methods

Figure 5.6: a) Audio signal of the song #22 - “Afkoritissa”. b) Fundamental frequency of the audio file. Four motifs are manually selected. The first motif is used as a prototype to construct a COSFIRE filter. c) The response signal of the configured COSFIRE filter. The black spots indicate the local maxima points.

(Azzopardi and Petkov 2014). It turns out that the system becomes significantly faster and more accurate when multiple COSFIRE filters with smaller areas of support are used. The advantage of having multiple filters selective for shorter signals is that they allow for more deformation between different parts of the signals, yet
they keep certain rigidity within local parts. For a training motif, we construct a COSFIRE filter every \( d \) points, a parameter that defines the distance between the support centers of the concerned filters. In Fig. 5.7 we illustrate an example of a training motif where \((K = 5)\) COSFIRE filters are configured. The filters are configured with prototypes of width \((n = 30)\), the intervals between the time points of length \((\delta = 1)\) and the prototype centers separated with a distance of \((d = 120)\) time points. The thick curve segments show the prototype signals that are used for the configuration of the 5 filters.

In Fig. 5.8a we use the training motif in Fig. 5.6 to construct \((K = 15)\) COSFIRE filters with \(n = 20, \delta = 1\) and \(d = 90\). The centers of the filters are shown with black spots on the training motif. The response signals of the 15 COSFIRE filters \(R_{k-1 \cdot K}(t)\) are illustrated in Fig. 5.8b.

The final response is the combination of all the response signals from the five filters. For every time position \(t\) we obtain a similarity value that indicates the degree of similarity between the input motif and the training motif. To achieve this combination, it is necessary that all the response signals are aligned to the center of the training motif as shown in Fig. 5.8c. This is done by shifting each response
Figure 5.8: (a) Fundamental frequency of the song “Afkoritissa”. The centers of 15 COSFIRE filters are shown with black spots. (b) The responses of every COSFIRE filter. (c) The shifted responses. (d) The blurred shifted responses. (e) The final response.
Figure 5.9: (a) Fundamental frequency of the song #22 “Afkoritissa”. 15 COSFIRE filters are configured using the fundamental frequency of the second motif. (b) The responses of every COSFIRE filter. (c) The shifted responses. (d) The blurred shifted responses. (e) The final response signal.
signal $R_k(t)$ by the distance from the corresponding prototype $k$ to the center of the training motif. Then, we apply the same smoothing function (with standard deviation $\hat{\sigma}$) explained in Section 5.2.3 to the shifted response signals defined as $R'_k(t)$. The final response is computed as the geometric mean of the smoothed and shifted responses:

$$R(t) = \left( \prod_{k=1}^{n} R'_k(t) \right)^{\frac{1}{n}}$$

(5.5)

The smoothed and shifted responses and the final response signals are shown in Fig. 5.8d and 5.8e, respectively. It is shown in this example that, besides motif 1, which was used for configuring this hierarchical COSFIRE filter, we obtain a local maximum in the middle of the third motif. We label motifs 1 and 3 with the letter “A”. In Fig. 5.9 we take the second motif as a prototype signal and label it with the letter “B”. We follow the same procedure explained above to configure a hierarchical COSFIRE filter and achieve the response signal in Fig. 5.9e. Motifs 2 and 4 are similar and they are represented with the letter “B”.

In order to make the process more efficient we apply each low level COSFIRE filter in a specific area of the test motif which we call the response area. First we compute the relative position of the center of the filter in percentage with respect to the normalized duration of the training motif. To find the response area, we multiply this percentage with the duration of the test motif. Then, we apply the COSFIRE filter in an area of three times the value of the $n$ parameter around the relative position in the test signal. Even if the training and the test motifs have a considerable difference in duration but the melody is similar, the approach will detect their similarity.

### 5.3 Experiments and Results

Here, we demonstrate the effectiveness of the proposed 1D COSFIRE filters in the identification of melodic patterns in monophonic singing folk tunes.

#### 5.3.1 Data

We have created a new data set that consists of 38 audio files of monophonic singing folk tunes of Cyprus with a total duration of 89 minutes (average of 2.3 and standard deviation of 1.2 minutes) and 878 motifs. In all of the songs the performer is a different person. In Table 5.1 we present the data used together with their properties sorted in ascending order of their durations. It is observed that the number of motifs roughly increases linearly with the durations of the songs. The songs were
performed by different persons in different locations and time periods. Of the 38 audio files, 12 were provided by Michalis Terlikkas\(^2\) who is a researcher and experienced musician of the folk music of Cyprus. The remaining 26 songs were collected from the freely available database that is provided by the Intercollege University\(^3\). The full names of the songs are reported in footnote\(^4\).

The equipment used for the recordings remains unknown. The main characteristic of these songs is that they are monophonic singing folk tunes of Cyprus. These are melodies that were transmitted orally for more than 500 years. Considering the geographical area of Cyprus that is an island in the Eastern Mediterranean sea and the conquerors that influenced the people in several aspects through the years, it is highly probable that cultural elements of those countries were also adapted to the local folk music. Therefore, this data set has a particular value also in the field of ethnomusicology.

### 5.3.2 Ground truth data

All of the songs in our data set were manually segmented into motifs by the aforementioned researcher of the folk music of Cyprus, Michalis Terlikkas. We used the annotation tool from the WaveLab software\(^5\) and exported the start and end positions of each motif in text format. Then we labelled the first motif of each song with the letter “A”. All subsequent motifs that are similar to the first one of each song also take the same label “A”. Then we gave the letter “B” to the next unlabelled motif and all subsequent ones similar to it. We repeated this procedure until all the motifs were labeled.

### 5.3.3 Pre-processing

We apply four pre-processing steps namely error correction, outlier removal, detrending and scaling.

The output of the YIN algorithm may contain two types of errors. The first is the failure of fundamental frequency candidates for non-harmonic sounds. Typically,
5.3. Experiments and Results

Table 5.1: Details of the new data set composed of 38 monophonic songs.

<table>
<thead>
<tr>
<th>Name</th>
<th>Duration (s)</th>
<th># Motifs</th>
<th>Name</th>
<th>Duration (s)</th>
<th># Motifs</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>53.3</td>
<td>14</td>
<td>S20</td>
<td>120.9</td>
<td>20</td>
</tr>
<tr>
<td>S2</td>
<td>65</td>
<td>6</td>
<td>S21</td>
<td>121.9</td>
<td>12</td>
</tr>
<tr>
<td>S3</td>
<td>67.6</td>
<td>15</td>
<td>S22</td>
<td>123.6</td>
<td>20</td>
</tr>
<tr>
<td>S4</td>
<td>67.7</td>
<td>14</td>
<td>S23</td>
<td>125.8</td>
<td>24</td>
</tr>
<tr>
<td>S5</td>
<td>70.9</td>
<td>8</td>
<td>S24</td>
<td>129</td>
<td>18</td>
</tr>
<tr>
<td>S6</td>
<td>74.9</td>
<td>18</td>
<td>S25</td>
<td>149.7</td>
<td>25</td>
</tr>
<tr>
<td>S7</td>
<td>75.2</td>
<td>15</td>
<td>S26</td>
<td>152.7</td>
<td>24</td>
</tr>
<tr>
<td>S8</td>
<td>78.2</td>
<td>10</td>
<td>S27</td>
<td>163.2</td>
<td>54</td>
</tr>
<tr>
<td>S9</td>
<td>78.7</td>
<td>16</td>
<td>S28</td>
<td>165.1</td>
<td>36</td>
</tr>
<tr>
<td>S10</td>
<td>80.6</td>
<td>16</td>
<td>S29</td>
<td>171.8</td>
<td>20</td>
</tr>
<tr>
<td>S11</td>
<td>84.6</td>
<td>8</td>
<td>S30</td>
<td>187.2</td>
<td>56</td>
</tr>
<tr>
<td>S12</td>
<td>84.6</td>
<td>6</td>
<td>S31</td>
<td>212.1</td>
<td>31</td>
</tr>
<tr>
<td>S13</td>
<td>89.5</td>
<td>14</td>
<td>S32</td>
<td>240.8</td>
<td>30</td>
</tr>
<tr>
<td>S14</td>
<td>98.6</td>
<td>17</td>
<td>S33</td>
<td>243.5</td>
<td>34</td>
</tr>
<tr>
<td>S15</td>
<td>106.6</td>
<td>8</td>
<td>S34</td>
<td>254.4</td>
<td>26</td>
</tr>
<tr>
<td>S16</td>
<td>106.6</td>
<td>21</td>
<td>S35</td>
<td>270.1</td>
<td>69</td>
</tr>
<tr>
<td>S17</td>
<td>106.6</td>
<td>16</td>
<td>S36</td>
<td>274.5</td>
<td>32</td>
</tr>
<tr>
<td>S18</td>
<td>112.3</td>
<td>24</td>
<td>S37</td>
<td>286.4</td>
<td>29</td>
</tr>
<tr>
<td>S19</td>
<td>116.6</td>
<td>18</td>
<td>S38</td>
<td>324.6</td>
<td>54</td>
</tr>
</tbody>
</table>

the first local maximum of the output of the ACF represents the fundamental frequency of a periodic signal. For non-periodic signals, such as noise, no fundamental frequency exists and the ACF returns no local maxima. In monophonic songs, non-periodic sounds appear within the vocal pauses. This fact results in the YIN algorithm producing artefacts as candidates for fundamental frequency. In order to correct such errors, we applied a method that was proposed in (Panteli 2011) to remove the wrong values and replaced them with linear interpolation.

The second type of errors of the YIN algorithm are the octave or fifth errors and they appear in situations where the algorithm erroneously outputs the frequency of one of the harmonics of the fundamental frequency. In order to correct such errors, we used the function FILTERF0.m from the MAKAM toolbox that is developed by Bozkurt (Bozkurt 2008).

In folk monophonic music, it is common that singers gradually change the pitch while singing. To illustrate an example, in Fig. 5.11a we show the fundamental frequency of the song “O Pramateftis”, which has an upward trend. In order to remove the trend in the signal, we subtract from it the best-fit polynomial of order 2. As for outlier removal, we identify and remove those fundamental frequency values
that are three standard deviations away from the mean of the entire signal. Finally, we apply min-max scaling such that the data points become in the range [0,1]. In this way, the method becomes transposition invariant. The preprocessed signal is shown in Fig. 5.11b.

5.3.4 Tuning of Parameters

We split the data set into an evaluation and a test set. The evaluation set consists of the first 19 songs that are listed in Table 5.1 which we use to fine tune the parameters of the proposed COSFIRE-based approach that is characterized by six parameters; width of prototype sub-motifs \( n \), distance between prototype sub-motifs \( d \), frequency tolerance parameters \( \sigma_0 \) and \( \alpha \), as well as temporal parameters \( \sigma_1 \) and \( \hat{\sigma} \). In practice, we perform a grid search over the sets of parameter values: \( n = \{30, 50, 70\}, d = \{30, 50, 70\}, \delta = \{1, 10, 20\}, \sigma_0 = \{0.001, 0.005, 0.01\}, \)

\( \sigma_1 = \{0, 0.01, 0.02\}, \hat{\sigma} = \{0, 0.01, 0.02\} \).
5.3. Experiments and Results

Figure 5.11: (a) The fundamental frequency of the song #35 “O Pramateftis” before and (b) after detrending.

\[ \alpha = \{0.001, 0.005, 0.01\}, \sigma^r = \{0.001, 0.005, 0.01\} \text{ and } \sigma^t = \{0.1, 0.5, 1\}. \]

The output of the proposed algorithm to a given audio signal is a string of letters and is obtained by the method explained in Section 5.2.1. Its length is equal to the number of motifs in the given audio signal. Then, we compute the Hamming distance between the resulting string and the string given in the ground truth and divide with the length of the strings. The set of parameters \( n = 30, d = 50, \delta = 1, \sigma_0 = 0.001, \alpha = 0.01, \sigma^t = 0.01 \text{ and } \sigma^r = 0.5 \) achieved the minimum mean Hamming distance over the validation set.

5.3.5 Comparison to other methods

We compare our results with other popular methods in signal processing namely cross correlation, Dynamic Time Warping (DTW) and Symbolic Aggregate approxi-
5. Melodic similarity using COSFIRE filters

Melodic similarity using COSFIRE filters (Keogh et al. 2006). The DTW is widely used in MIR applications and in similar problems such as the identification of melodic repetitions.

Cross Correlation

It slides one signal on the other and computes the dot product at each position. The maximum value is used as a measure of similarity between the two signals. We tune the threshold parameter for the dissimilarity classification. We compute the Precision and the Recall for a set of threshold values and we choose the value that returns the maximum $F_1$ score.

Dynamic Time Warping (DTW)

The DTW method has been used in a variety of applications in signal processing, particularly in MIR and in speech recognition (Peiszer 2007, Shiu et al. 2005, Jehan 2005, Muda et al. 2010). The main benefit of this method is that it allows for temporal tolerance when comparing two time signals with different durations. Other methods, such as cross correlation, are less robust in this respect. The DTW calculates the optimal path between two given time signals and it returns a value between 0 and 1. A value of zero indicates identical signals. This method has the threshold parameter to tune for the dissimilarity classification.

Symbolic Aggregate Approximation (SAX)

The SAX method computes a symbolic representation of a 1D signal by assuming that the data points are normally distributed. It uses two parameters, the size of a vocabulary and the length of the resulting symbolic string. We standardize the given signal to have a zero mean and a standard deviation of 1. For the technical details on this method we refer the reader to (Keogh et al. 2006).

For the three above mentioned methods we use a similar procedure that is explained in Section 5.2.1 to obtain a string representation. For a given audio signal, we take the first motif and use the concerned method (cross correlation, DTW and SAX) to obtain a similarity value for the remaining motifs in the same signal. We give the letter “A” to the first motif and to the motifs for which a similarity value above a certain threshold is achieved. Then, we take the next motif that has not yet been labeled and compare it with the remaining unlabelled motifs, and give the letter “B” to the ones that evoke a response greater than the threshold. We repeat this process until all motifs are labeled. For the SAX method we use the minimum (lower-bounding) distance $d$ suggested in (Keogh et al. 2006) to compute the similarity between the symbolic representations of the motifs. Moreover, we use the

\[d \text{ in practice we use the Matlab implementation provided in http://www.cs.ucr.edu/~eamonn/SAX.htm}\]
5.3. Experiments and Results

validation set to fine-tune the involved two parameters by a grid search. The parameters that achieve the minimum mean Hamming distance are a vocabulary of 8 letters and a length of 16 letters in the resulting symbolic string. The other two methods have no specific parameters to tune other than the threshold parameter.

5.3.6 Results

The test set consists of the last 19 songs that are listed in Table [5.1]. For the COSFIRE approach that we propose and for the SAX method we use the parameter values that returned the best results from the grid search on the validation set. For the other two methods we tune the threshold that is applied for the dissimilarity classification. We use a different threshold for each method that returns the maximum $F_1$ score over a set of threshold values.

We present the results of the four methods (COSFIRE, DTW, cross correlation and SAX) that we applied to our data set in terms of Precision $P=TP/(TP+FP)$ and Recall $R=TP/(TP+FN)$ and the harmonic mean also known as $F_1$ score $2PR/(P+R)$.
Figure 5.13: The sum of normalized Hamming distances achieved by the proposed COSFIRE approach in comparison to cross correlation, DTW and SAX, as a function of the involved threshold that is common to all methods. The markers indicate the minimum point of each plot.

For the symbolic representation matching, we calculate the normalized Hamming distance between the strings of the ground truth and the output of each of the four methods. Then, we obtain the mean Hamming distance over the entire test set for various values of the threshold $th$.

The TP, FP and FN are the abbreviations for True Positives, False Positives and False Negatives, respectively. A classification is called TP if a pair of motifs that are compared are annotated as similar in the ground truth and the method classified them as similar. A FP is the wrong classification of the system that two motifs are similar, while a FN is the wrong classification that two motifs are dissimilar.

The classification of a pair into similar or dissimilar has been done using a threshold on the output of the respective method. We have computed results for 101 values of the output threshold in the range 0 and 1 in intervals of 0.01. These results are presented in a precision and recall plot in Fig. 6.8. In Fig. 5.13 we plot the mean of Hamming distances over the test set for the four methods as a function of the threshold $th$.

The proposed COSFIRE approach outperforms the other methods. The har-
monic mean of the precision and recall reaches a maximum value of 0.83 at a recall R of 0.80 and a precision P of 0.86. The precision and the recall of the SAX method at its maximum harmonic mean of 0.46 is 0.41 and 0.52, respectively. The harmonic mean of the precision and recall of the DTW reaches its maximum $F_1 = 0.66$ at a (R = 0.74, P = 0.59). The maximum harmonic mean of the cross correlation is achieved with $F_1 = 0.60$ at a (R = 0.75, P = 0.5).

For the symbolic representation, COSFIRE achieves a minimum mean of Hamming distances of 0.13 with a threshold $th = 0.74$. On the other hand, the minimum mean of Hamming distances of the cross correlation, DTW and SAX are 0.66 ($th = 0.01$), 0.38 ($th = 0.85$) and 0.39 ($th = 0.46$), respectively.

Additionally, we calculated the processing time that each method requires to process every song. The execution time of the methods COSFIRE, SAX and cross correlation to compute 38 songs were 51.5, 0.9 and 0.1 minutes respectively, while that of the DTW was 729.6 minutes.

For each of the four methods and for each song, we present in Table 5.3.6 the $F_1$ score, the execution time and the normalized Hamming distance for the threshold that contributes to the minimum $F_1$ score and mean of Hamming distances on the test set. The execution time is calculated with sequential implementation of the methods running on a personal computer with a 1.7 GHz processor and 8 GB of RAM.

5.4 Discussion

The COSFIRE approach performs substantially better than the cross correlation, DTW and SAX methods. Most of the existing methods for the identification of motifs make use of the DTW in their methodology. Therefore, we believe that replacing DTW by the proposed COSFIRE approach may yield significant improvement in several applications. While the SAX and the cross correlation methods are more efficient than COSFIRE, they are much less effective possibly due to the insufficient robustness to temporal tolerance.

In contrast to the QBH method that has been a state-of-the-art for melodic similarity tasks, COSFIRE filters compare pitch tracks that are derived directly from the audio signal. In other words, the note segmentation and the database with the MIDI transcriptions are avoided. Similarly, wavelets have been effective in applications that are using the pitch tracks. Unlike COSFIRE, wavelets are linear functions and are not intrinsically robust to temporal tolerance. They are typically used to extract features that can be used in a classification model. This is in contrast to the proposed COSFIRE filtering approach that can be used directly as a similarity function without involving classification models.

Several data sets for folk music analysis are used in the literature, while not all
Table 5.2: The execution time (ET) and the normalized Hamming distances (NHD) of each song for the threshold that contributes to the minimum mean of hamming distances for the four methods.

<table>
<thead>
<tr>
<th>Song #</th>
<th>ET (min)</th>
<th>NHD</th>
<th>ET (min)</th>
<th>NHD</th>
<th>ET (min)</th>
<th>NHD</th>
<th>ET (min)</th>
<th>NHD</th>
<th>ET (min)</th>
<th>NHD</th>
<th>Mean</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.41</td>
<td>0.00</td>
<td>0.69</td>
<td></td>
<td>17.40</td>
<td>0.50</td>
<td>0.72</td>
<td></td>
<td>0.24</td>
<td>0.75</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0.25</td>
<td>0.00</td>
<td>1.00</td>
<td></td>
<td>15.09</td>
<td>0.25</td>
<td>0.78</td>
<td></td>
<td>0.08</td>
<td>1.00</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>0.39</td>
<td>0.25</td>
<td>1.00</td>
<td></td>
<td>12.18</td>
<td>0.15</td>
<td>0.88</td>
<td></td>
<td>0.18</td>
<td>0.60</td>
<td>0.87</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0.57</td>
<td>0.04</td>
<td>1.00</td>
<td></td>
<td>16.15</td>
<td>0.21</td>
<td>0.61</td>
<td></td>
<td>0.30</td>
<td>0.50</td>
<td>0.26</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.44</td>
<td>0.28</td>
<td>0.67</td>
<td></td>
<td>16.87</td>
<td>0.17</td>
<td>0.59</td>
<td></td>
<td>0.16</td>
<td>0.11</td>
<td>0.22</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0.70</td>
<td>0.00</td>
<td>1.00</td>
<td></td>
<td>22.72</td>
<td>0.08</td>
<td>0.95</td>
<td></td>
<td>0.30</td>
<td>0.00</td>
<td>0.85</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0.64</td>
<td>0.00</td>
<td>1.00</td>
<td></td>
<td>20.71</td>
<td>0.71</td>
<td>0.96</td>
<td></td>
<td>0.30</td>
<td>0.00</td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>1.54</td>
<td>0.04</td>
<td>0.97</td>
<td></td>
<td>23.97</td>
<td>0.56</td>
<td>0.95</td>
<td></td>
<td>1.17</td>
<td>0.59</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>1.10</td>
<td>0.31</td>
<td>0.90</td>
<td></td>
<td>23.62</td>
<td>0.58</td>
<td>0.83</td>
<td></td>
<td>0.55</td>
<td>0.42</td>
<td>0.90</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>0.63</td>
<td>0.05</td>
<td>1.00</td>
<td></td>
<td>25.93</td>
<td>0.05</td>
<td>0.97</td>
<td></td>
<td>0.20</td>
<td>0.00</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1.92</td>
<td>0.02</td>
<td>0.99</td>
<td></td>
<td>30.86</td>
<td>0.68</td>
<td>0.89</td>
<td></td>
<td>1.51</td>
<td>0.79</td>
<td>0.77</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>1.20</td>
<td>0.03</td>
<td>0.76</td>
<td></td>
<td>37.02</td>
<td>0.71</td>
<td>0.60</td>
<td></td>
<td>0.44</td>
<td>0.65</td>
<td>0.57</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>1.26</td>
<td>0.00</td>
<td>0.85</td>
<td></td>
<td>43.52</td>
<td>0.07</td>
<td>0.93</td>
<td></td>
<td>0.46</td>
<td>0.33</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>1.41</td>
<td>0.47</td>
<td>0.87</td>
<td></td>
<td>47.24</td>
<td>0.15</td>
<td>0.80</td>
<td></td>
<td>0.54</td>
<td>0.97</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>1.09</td>
<td>0.08</td>
<td>0.98</td>
<td></td>
<td>49.62</td>
<td>0.12</td>
<td>0.66</td>
<td></td>
<td>0.38</td>
<td>0.08</td>
<td>0.82</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>2.22</td>
<td>0.02</td>
<td>0.59</td>
<td></td>
<td>61.73</td>
<td>0.33</td>
<td>0.68</td>
<td></td>
<td>1.18</td>
<td>0.00</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>1.38</td>
<td>0.22</td>
<td>0.86</td>
<td></td>
<td>56.77</td>
<td>0.66</td>
<td>0.88</td>
<td></td>
<td>0.52</td>
<td>0.66</td>
<td>0.72</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>2.66</td>
<td>0.55</td>
<td>0.76</td>
<td></td>
<td>62.88</td>
<td>0.67</td>
<td>0.60</td>
<td></td>
<td>1.47</td>
<td>0.62</td>
<td>0.79</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>3.00</td>
<td>0.07</td>
<td>0.96</td>
<td></td>
<td>79.15</td>
<td>0.50</td>
<td>0.78</td>
<td></td>
<td>1.37</td>
<td>0.76</td>
<td>0.65</td>
<td></td>
</tr>
</tbody>
</table>

Mean 1.20 0.13 0.89 34.92 0.38 0.79 0.60 0.46 0.63 0.51 0.65 0.66
Standard deviation 0.79 0.17 0.13 19.68 0.25 0.14 0.48 0.34 0.23 0.38 0.16 0.14

5. Melodic similarity using COSFIRE filters
of them are publicly available. We have tried to apply our method in the MTC-ANN-2.0 data set from the Meertens Tune Collections that is used in (Rodríguez-López and Volk n.d., Janssen et al. 2015, van Kranenburg and Tzanetakis 2010) for tasks such as melodic segmentation and motif identification. We have encountered some difficulties related to the consistency of the data sets that are used in (Janssen et al. 2015) and the publicly available data set, in order to compare our results. In (Janssen et al. 2015), the experiments are done using a subset of the MTC-ANN-2 data set of 16 tune families while the published data set contains 28. The data are stored in kern format and thus the music information is encoded with the MIDI protocol. In addition to this, the melodic contour of some of the annotated motifs that are used for training is significantly different than their similar ones that are used for testing. Therefore, it is possible that the annotators used other features than the melody for ranking the similarity of the motifs, such as the rhythmic pattern. For the above mentioned reasons, COSFIRE does not seem to be an appropriate method for that data set. In (Ross et al. 2012) the authors use data from a personal collection and the annotation of the motifs was done manually.

In Fig. 5.14, we present the distribution of the fundamental frequency of the first motif of the song #1 “Tis Sousas 1”. The values are concentrated around four frequencies, the ones that represent the most common notes that were performed during the analyzed song. The theoretical frequencies of these notes are C#4 (277.2Hz), D#4 (311.1Hz) and F4 (349.2Hz), which are indicated with dashed vertical lines in Fig. 5.14. This example demonstrates that the fundamental frequency of a song does not follow a normal distribution. In this respect, the SAX method may not be suitable for such signals because it assumes an underlying normal distribution.

The benefit of converting a song into a symbolic representation is twofold. First, it permits data compression and thus allows faster comparison between two musical signals. The symbolic representation is a shorter sequence of musical events that can be used as a higher level feature for classification. Second, from a musicological point of view, the sequence of the repeating motifs is an important and interesting feature in itself. It is very common in musicology to split a song into musical events and report their sequence and the frequency of their appearance. The automatic identification of such events allows the analysis of large data sets in relatively short period of time.

A system that automatically analyzes folk music is very important since there are no written musical scores for musicological analysis. In folk music, musicologists mark manually every musical event for their research. Similar to other applications, manual annotation may result in inaccuracies due to fatigue and is certainly much slower than automated methods.

We present a novel filter based approach for the identification of motifs and a
method for representing a monophonic song into a symbolic string. We have shown that the proposed COSFIRE filters allow temporal tolerance but are more effective and more efficient than DTW. This is mainly attributable to the hierarchical structure that COSFIRE uses. DTW gives the same temporal tolerance to every part of the signal irrespective of its position. As demonstrated in our experiments, COSFIRE filters can be configured to take input from other COSFIRE filters that are selective for smaller parts of the signal. This arrangement provides the possibility to have low tolerance in local parts and higher global tolerance between the involved parts.

The COSFIRE filters are trainable and easy to implement. Even though the data set used to apply our method refers to ethnomusicological interest, the proposed COSFIRE filters can be applied in any application that involves 1D signals, such as the identification of $k$-complexes and spindles in EEG signals (Camilleri et al. 2014).

The computation of a COSFIRE filter response is parallelizable as it relies on independent operations defined in the concerned set of tuples. While this was beyond the scope of this work, we speculate that such parallelization would largely improve the efficiency of the algorithm by a factor that corresponds to the number of tuples.

In this work, we are not dealing with automatic segmentation in motif and note level. Moreover, our data set is a simplistic representation of the general music that
is available as data since we applied our method in monophonic songs. Therefore, the major contribution of the COSFIRE filters that are proposed in this chapter is their use as a distance metric between two sequences. For future work we aim to use the nested COSFIRE filters in order to develop an automated system for motif segmentation. Additionally, we will explore the potential use of other algorithms for polyphonic melody extraction to expand the generalization ability of our method.

5.5 Conclusion

We propose an effective trainable filter approach for identifying motifs in 1D signals, with application to acoustic signals. The transformation of a song into a symbolic representation leads to data reduction and the identification of the most important melodies in a song.

By means of experiments on a new data set of monophonic folk songs, we demonstrated that the proposed method outperforms the existing cross correlation, SAX and DTW methods with a time complexity that is higher than that of the former two methods but substantially lower than that of DTW.