Testing a new tool for alignment of musical recordings

Hannah Ilea Robertson

Music Technology Area
Schulich School of Music
McGill University
Montreal, Canada

December 2013

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Arts.

Abstract

Audio-to-audio alignment of musical recordings is the mapping of events in one recording to their corresponding events in other recordings of the same underlying musical piece. Among other applications, musical audio-to-audio alignment is used for: comparing and analyzing musical performances; finding different performances and arrangements of a musical work in a database; discovering musical motifs in field recordings of folk music; automatically synchronizing multiple takes (re-recordings of specific excerpts) in a recording studio; and aligning a musician's performance to a score in realtime, for purposes of interactive performance such as automated accompaniment. This thesis investigates audio-to-audio alignment by an algorithm that has not previously been applied to music, the continuous profile model (CPM) (Listgarten et al. 2005).

In this thesis, the CPM is used to align pairs of recordings (pairwise alignment) as well as groups containing more than two recordings (multiple alignment). A standard evaluation methodology is used to systematically compare pairwise alignment by the CPM to pairwise alignment by dynamic time warping (DTW), the algorithm most frequently used for audio-to-audio alignment of music. The evaluation methodology is then generalized to multiple dimensions in order to compare two approaches to multiple alignment: simultaneous multiple alignment with the CPM and iterative pairwise alignment with DTW.

Résumé

L'alignement audio-audio des enregistrements sonores est le mappage de certains moments dans un enregistrement aux moments correspondants dans d'autres enregistrements de la même pièce musicale. Parmi d'autres applications, l'alignement audio-audio sert à : comparer et analyser des performances musicales; trouver de plusieurs performances et arrangements d'une oeuvre musicale dans une base de données; découvrir des motifs musicaux dans des enregistrements de terrain de la musique folk; synchroniser automatiquement de multiples prises (réenregistrements d'extraits spécifiques) dans un studio d'enregistrements; et aligner une performance d'un musicien qui utilise une partition en temps réel pour permettre des performances interactifs avec des ordinateurs, par exemple, l'accompagnement automatisé. Dans cette thèse, on examine l'alignement des enregistrements musicaux accompli par un algorithme qui n'a auparavant jamais été appliqué à la musique qui s'appelle le « modèle du profil continu » (« continuous profile model » ou « CPM » en anglais) (Listgarten et al. 2005).

Cette thèse examine la CPM pour les alignements deux à deux et les alignements multiples des enregistrements sonores. On emploie une méthodologie standard pour comparer systématiquement l'alignement deux à deux accompli par le CPM à l'alignement deux à deux accompli par la « déformation temporelle dynamique » (en anglais, « dynamic time warping » ou « DTW »), l'algorithme le plus souvent employé pour l'alignement audio-audio de la musique. La méthodologie d'évaluation est ensuite généralisé à plusieurs dimensions à comparer deux approches pour l'alignement multiple : alignement multiple simultanée par le CPM à itérative alignement par paires avec DTW.

Acknowledgments

This project would not have reached completion without the time, energy, and support of a number of people and organizations:

Many thanks are due my advisor, Professor Ichiro Fujinaga, without whom this project would not have been possible. His technical knowledge and constructive feedback have been invaluable both when researching and writing. I'd also like to acknowledge support from the rest of the faculty in the Music Technology area at McGill University's Schulich School of Music.

On a technical level, this thesis was made possible by the open-source CPM Toolbox for MATLAB, made available by Dr. Jennifer Listgarten, and access to the Chopin dataset, provided by Werner Goebl. I am grateful for these contributions.

I'd like to gratefully acknowledge sponsorship by the Audio Engineering Society (AES) through the Educational Grant for Graduate Studies.

A big thank you to Hallie Gammon for translating the abstract, and to Lizzie Gordon, Alison Mehravari, Emma Robertson, and my parents for their cheerleading.

I am especially grateful to my labmates for their support, both technical and personal. The thesis process would not have been half so enjoyable—or caffeinated!—without their friendship. In particular, thanks to Ashley Burgoyne, Anton Khelou, Andrew Hankinson, Jason Hockman, Alastair Porter, and Gabriel Vigliensoni. A special thanks to Cory McKay for deftly solving a particularly stubborn snarl.

Last but not least, I am thankful for the camaraderie of my cohort: Chuck Bronson, Greg Burlet, Chelsea Douglas, and Ryan Groves, among many others. It has been a pleasure working with you.

Contents

List of Figures

List of Tables

List of Acronyms

API	application programming interface
AMPACT	Automated Music Performance Analysis and Comparison Toolkit
BLAST	Basic Local Alignment Search Tool
CENS	chroma energy distribution normalized statistics
CLI	command-line interface
CPM	continuous profile model
CRP	cross recurrence plot
DLNCO	decaying locally adaptive normalized chroma-based onset
DP	dynamic programming
DTW	dynamic time warping
EM	expectation-maximization algorithm
FFT	fast Fourier transform
GUI	graphical user interface
HB-CPM	hierarchical Bayesian continuous profile model
HMM	hidden Markov model
LC–MS	liquid chromatography–mass spectrometry
MATCH	Music Alignment Tool CHest
MFCC	Mel-frequency cepstral coefficient

MIR	music information retrieval
MIREX	Music Information Retrieval Evaluation eXchange
MSA	multiple sequence alignment
MSD	Million Song Dataset
OMR	optical music recognition
PSD	peak structure distance
RQA	recurrence quantification analysis
SMC	sequential Monte Carlo
STFT	short-time Fourier transform

1—Introduction and motivation

Audio-to-audio alignment of musical recordings is the mapping of events in one recording to the corresponding events in other recordings of the same underlying musical content. Motivations for performing audio-to-audio alignment include comparing and analyzing musical performances; finding different performances and arrangements of a musical work in a database; discovering musical motifs in field recordings of folk music; automatically synchronizing multiple takes in a recording studio; and aligning a musician's performance to a score in realtime, for purposes of interactive performance such as automated accompaniment. In one specific example, multiple field recordings of the same folk song have been aligned to one another in order to study differences among them in tempo, tuning, and melodic ornamentation (Müller et al. 2010).

Some audio-to-audio alignment tasks require pairwise alignment: mapping the events between a pair of recordings. One example of pairwise alignment is the synchronization of two different performances of a composition, such as synchronizing Jean Pierre Rampal's recording of Claude Bolling's "Baroque and Blue" with the Roselli Quartet's recording of the same piece. Pairwise alignment is also required in tasks that involve a specific reference recording. For example, to determine which of a selection of live performances of Queen's "Under Pressure" is the most similar (in pitch/rubato/harmonization/etc.) to the version recorded on their 1982 album *Hot Space*, each of the live recordings could be aligned individually to the reference track—that studio recording—in order to calculate a similarity measure. The output similarity measures could then be ranked to determine the most similar of the live recordings. As of publication time, most audio-to-audio alignment tasks in the literature use pairwise alignment (as is made clear by Table 2.1 in Chapter 2).

Other audio-to-audio alignment tasks involve multiple alignment: mapping a whole group of recordings to one another. Tasks that fit into this category often involve no obvious reference recording. For example, to answer the question "which

phrases of the traditional Irish jig 'The Lark on the Strand' are often played with the same ornamentation, phrasing, and emphasis, and in which sections are there often variations?" multiple recordings of this tune[1] must be aligned to one another before an analysis can take place. This task contains no obvious reference recording, as no single recording is more significant than any other.

Algorithms used to perform audio-to-audio alignment are often generic time-series alignment algorithms, developed for alignment of data other than musical audio and later adapted to align music. For example, musical audio-to-audio alignment is often implemented through dynamic time warping (DTW), a pairwise alignment algorithm that was first introduced in the context of speech recognition tasks in the 1970s. The application of generic alignment algorithms to musical audio is possible because audio recordings are time-series data, sequences of successively measured data observations. In regards to terminology, time-series sequences are also called signals and data observations are often called features. "Temporal registration" is a term commonly used to describe alignment of time-series data.

There are two main approches for aligning more than two signals (musical or otherwise) to one another: an iterative pairwise approach and a simultaneous multiple alignment approach. These two approaches to multiple alignment are illustrated in Figure 1.1. The iterative pairwise approach involves first performing a series of pairwise alignments and then inferring the overall group mapping through iterative association among the pairs. This association can take place in a number of ways, such as through choice of a common reference signal (e.g., aligning signal B to signal A, then signal C to A, and finally D to A) or through sequential linking (e.g., aligning signal A to signal B, aligning B to C, and aligning C to D). In contrast, the simultaneous approach to multiple alignment involves leveraging the information from all signals when performing alignment. One simultaneous alignment approach involves combining the original signals into a hybrid signal and then using that hybrid as a reference signal (e.g., averaging four signals A, B, C, and D into signal Z and then pairwise aligning A to Z, B to Z, and so on). Simultaneous multiple alignment algorithms are also known as multiple sequence alignments (MSAs).

When multiple alignment is required, the iterative pairwise alignment approach is often chosen (Sapp 2007; Montecchio and Cont 2011b). While MSA algorithms have been applied to alignment tasks in non-musical fields, from biological sequence analysis (Chan et al. 1992) to the modeling of human motion (Zhou and De la

[1] As of March 2013, Irish folk-music website *The Session* notes 45 independent recordings of this tune: `www.thesession.org/tunes/1634/recordings`.

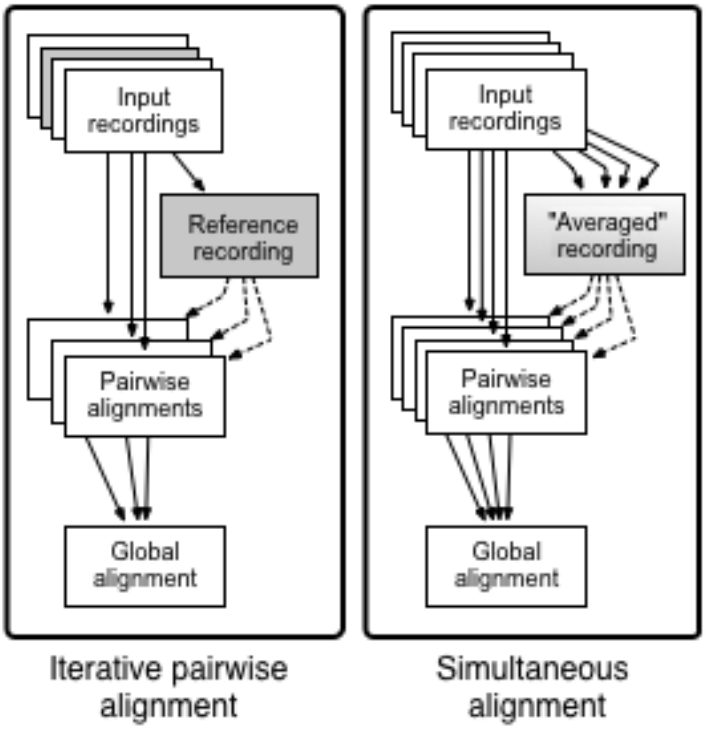

Fig. 1.1 Two approaches to multiple alignment: iterative pairwise alignment, using an arbitrarily chosen reference signal to map the signals to one another, and simultaneous alignment, using a reference signal created from a fusion of all input signals.

Torre 2012), applications of MSA algorithms to musical recordings—and audio in general—have been limited. To the best of our knowledge, only two papers have applied MSA algorithms to musical recordings: In the first, Basaran et al. (2011) propose a probabilistic model for aligning multiple musical recordings of the same performance, as recorded by multiple microphones of varying quality and noisiness. In the second, Sanguansat (2012) uses multi-dimensional DTW to query a musical database with a user-sung input: alignment is calculated in a multi-dimensional space, with as many dimensions as aligned signals, rather than the standard two-dimensional space of pairwise alignment (Zhou and De la Torre 2012).

This thesis implements audio-to-audio alignment with a relatively new MSA algorithm: the continuous profile model (CPM). The CPM is a probabilistic model developed to simultaneously align and perform difference detection (detection of overall signal similarity or difference) on a group of signals. While the CPM has been shown to align speech recordings (Listgarten 2007), to the best of our knowledge it has not yet been used to align musical recordings. Notably, a software implementation of the CPM has been made freely available as an open-source MATLAB toolbox (Listgarten 2007).[2]

1.1 Thesis structure

In this thesis, audio-to-audio alignment by the CPM is implemented and compared to audio-to-audio alignment by DTW. First, pairwise alignment with the CPM is compared to pairwise alignment by DTW. A standard methodology for evaluating pairwise alignment, set forth by Dixon and Widmer (2005), is used to systematically compare the algorithms' results. This evaluation involves measuring the amont of deviation between an algorithmically generated alignment and the ground-truth alignment. Second, two approaches to multiple alignment (simultaneous alignment with the CPM versus iterative pairwise alignment using DTW) are compared through aligning groups of three, four, eight, twelve, and sixteen recordings. The aforementioned pairwise evaluation metric is generalized to evaluate the results of multiple alignment, which are in the form of multi-dimensional alignment paths. The thesis is organized as follows:

> **Chapter 2** presents a history of the alignment algorithms as they pertain both to audio and to symbolic music alignment; an overview of common

[2] www.cs.toronto.edu/~jenn/CPM/

algorithms and features; a literature review of research on audio-to-audio alignment; and a summary of commercial and open-source audio-to-audio alignment software.

Chapter 3 explains the CPM algorithm and describes how it is used to align musical recordings.

Chapter 4 describes the evaluation methodology, including the dataset; the evaluation metrics; and the implementation.

Chapter 5 presents and discusses the results of the evaluation results.

Chapter 6 concludes the thesis with a discussion of future work.

2—Literature review

This chapter places audio-to-audio alignment in its historical context (Section 2.1), describes algorithms and features used for alignment (Section 2.2), and reviews audio-to-audio alignment research applications (Section 2.3). The chapter concludes with a summary of commercial and non-commercial software designed specifically for alignment of music (Section 2.4).

2.1 The inception of music alignment

The first alignment algorithms applied to music, string alignment and DTW, were borrowed from the tasks of biological sequence alignment and speech recognition. String alignment algorithms were first developed for biological sequence alignment in 1970 (Needleman and Wunsch). In biological sequence alignment, sequences of DNA, RNA, and proteins are compared in order to determine their structural and evolutionary relationships. DTW was developed for and popularized by spoken word recognition tasks (Sakoe and Chiba 1971; Itakura 1975; Sakoe and Chiba 1978). Speech recognition must take variations in speaking rate into account; alignment helps minimize the effects of time variations when calculating the similarity of two utterances. For example, "good" (spoken quickly) and "goood" (spoken slowly) need to be identified as the same word, despite their different deliveries. By lining up the start of each phoneme, identification is often more accurate. (Both string alignment and DTW are dynamic programming (DP) algorithms, and will be described in the following section.)

Early applications of alignment algorithms to music focused on symbolic music representations. Dannenberg (1984) used DTW for a score-following system based directly on a speech-to-speech alignment system designed for overdubbing film dialogue (Bloom 1984). This score-following system aligned both symbolic and audio inputs to a symbolic score in realtime. Independently that same year, Vercoe

(1984) also performed audio-to-symbolic alignment for realtime score following, using a trained learning strategy rather than DTW.

String-matching algorithms were first applied to music when Mongeau and Sankoff (1990) calculated the similarity between symbolic scores in a work modeled after biological sequence alignment (Kruskal and Liberman 1983; Kruskal and Sankoff 1983). Stammen and Pennycook (1993) performed realtime identification of symbolic music after first simplifying the input symbolic notes into melodic contours, using DTW to align those contours. This music identification system was modeled after Itakura's (1975) discrete word recognition system.

The earliest alignment of audio representations of music to other audio representations of music occurred in 2001, when Yang (2001) used DTW to perform content-based music retrieval—searching a database of musical audio for excerpts most melodically similar to an audio query. That same year, Orio and Schwarz (2001) and Meron and Hirose (2001) each performed audio-to-symbolic alignment by first converting both symbolic scores and audio recordings to pitch-based harmonic energy features and then aligning those features.

More recently, symbolic music alignment has shifted towards probabilistic modeling approaches (discussed by Orio et al. 2003), following trends in both speech processing (Rabiner 1989) and biological sequence alignment (Haussler et al. 1993). Audio-to-audio alignment applications, however, have continued to favor DP-based algorithms. A few audio-to-audio alignment applications have used probabilistic models for performance analysis (Devaney et al. 2009; Devaney et al. 2011), studio engineering (Basaran et al. 2011), and joint structure analysis (Tabus et al. 2012), but DP algorithms like DTW are still very popular. This algorithmic preference will be highlighted by Table 2.1 at the end of Section 2.3.

2.2 Overview of algorithms and features

This section describes algorithms (Section 2.2.1) and features (Section 2.2.2) often used for audio-to-audio alignment.

2.2.1 Algorithm selection

For a given alignment task, the choice of algorithm and manner in which that algorithm is implemented must take into account several considerations. These considerations include the approach to multiple alignment if more than two recordings

are to be aligned (as discussed in Chapter 1); the necessary algorithmic constraints, based on both the similarity in global structure across recordings and the operational deadline of the alignment task (i.e., whether alignment needs to be performed in realtime or not); the relationship among the timelines of the recordings to be aligned; and the type of algorithmic approach desired.

Constraints

Two considerations relating to algorithmic constraints include the structural similarity of the recordings to be aligned and the operational deadline of the alignment task. Structural similarity influences the choice between global and partial alignment algorithms. Global alignment is the alignment of entire recordings to one another, while partial music alignment aligns segments of recordings to one another despite the presence of structural variations between the recordings (e.g., an extended intro or the omission of a verse). Operational deadlines influence the choice between online and offline algorithms. In contrast to offline (non-realtime) alignment, online alignment algorithms are designed to run in realtime: the alignment of any given sample relies on past and present samples, but not future samples, for at least one of the recordings being aligned.

For a particular alignment task, consideration of these constraints influences the parameters and programmatic implementation of an algorithm but not necessarily the specific type of algorithm chosen in the first place. DTW, for example, is generally implemented with boundary constraints that force global alignment, but has also been used for partial alignment (Müller and Appelt 2008; Ewert et al. 2009). Similarly, although DTW is generally implemented as an offline algorithm, it has been tailored to run online for some applications (Dixon 2005a).

Timeline transformations

By definition, two or more recordings of the same music contain a common, ordered set of musical events (e.g., onsets, beats, or feature frames). In this thesis, the common, ordered set of musical events in a recording is called its event timeline. Using this terminology, audio-to-audio alignment maps the event timelines of a set of recordings to one another.

Recordings of the same musical piece therefore have the same underlying event timeline, even though their individual event timelines are not necessarily identical. For example, two performances of Happy Birthday have the same underlying event

timeline, as they contain the same ordered set of pitches, note durations, and lyrics. Their individual event timelines will differ, however, if they are sung at different tempi, have different durations of the pauses between verses, or are sung with different tempo fluctuations.

The relationships between the event timelines of any two recordings can be described in terms of three transformations—linear shifting, linear scaling, and nonlinear warping:

- In a linearly shifted transformation the timelines of the recordings are identical (i.e., inter-event timing is preserved) but one recording is shifted in time relative to the other (translated), so that there is a delay at the beginning of one recording in relation to the other. For example, in recordings of the exact same musical performance made by two different recording devices, all timing information of the audio will be identical because both are recordings of the exact same sounds. One will likely have at least a slight lag before the start of the musical performance, however, as the "record" buttons may not have been pushed at precisely the same instant on both recording devices.

- In a linearly scaled transformation the ratio between event timings in each recording is preserved but the entire timeline of one recording is scaled (evenly stretched or compressed in time) in relation to the other, so that the tempo is faster in one recording than the other. For example, recordings played back at a faster sampling rate than they were recorded have timelines that are linearly scaled in relation to the original, as the timing of events in the faster version is merely a condensed version of the original timing. (This playback manipulation is called "pitching" by radio stations, where it is used to fit more songs or advertising into an hour of programming (Cano et al. 2002).)

- In a nonlinearly warped transformation the ratio between event timing in the recordings is *not* preserved, so that there are different fluctuations in event timings in each of the recordings. For example, different performances of the same symbolic score are nonlinearly related, due to the intentional and accidental tempo fluctuations (rubato) introduced by the different performers, as no two human performances of the same piece will have identical timing.

Recordings to be aligned often contain a combination of these transformations, and these transformations inform the choice of alignment algorithm. Cross-correlation,

for example, is ideal for linearly shifted timeline transformations but performs poorly on linearly scaled and nonlinearly warped transformations. DTW and hidden Markov models (HMMs), however, are designed to account for scaled and nonlinear timeline relationships. It should be noted that the relationship between two timelines is completely independent of the differences in spectral content between them. For example, two different performances of the same score by the same performer on the same instrument (e.g., two takes of Jean-Pierre Rampal performing Bach's Flute Concerto in A) will contain at least slight nonlinear temporal fluctuations. In contrast, sonifications of a single MIDI score by two different instrumentations (e.g., flute and trombone), will differ in timbre but have an identical, non-transformed timeline relationship.

Similarity matrices, also known as cross recurrence plots (CRPs) and dissimilarity matrices, are often used to visualize the structural similarities between a pair of recordings. A similarity matrix is created by plotting the distances (similarity measures) between every feature in one recording and every feature in the other recording as a color value in a two-dimensional grid.[1] The x-axis of a similarity matrix represents the timeline of one recording; the y-axis represents the timeline of the other. Diagonal lines in a similarity matrix indicate passages of alignment between the recordings, and the diagonally trending line from the start of both recordings to the end of both recordings is the global alignment warping path. A similarity matrix that compares a sequence to itself is called a self-similarity matrix (Figure 2.1).[2]

Figures 2.2 to 2.4 show example similarity matrices for each type of timeline transformation. In general, if the global alignment path (diagonally trending line through the entirety of both recordings) is linear (i.e., straight), there is a linear timeline transformation (shifted or scaled) between the two recordings; if the alignment line is nonlinear (e.g., curved or meandering), there is a nonlinear timeline transformation between them:

- For a linearly scaled transformation, the diagonal line stretches from the first to the last features in both recordings. Since by one recording is longer than the other, the similarity matrix is rectangular (Figure 2.2).

[1] The similarity between features is often calculated with the Euclidean distance measure.

[2] Self-similarity matrices were first applied to musical audio by Foote (1999), and have since been used to find musical structure, among other musical applications (Foote 2000; Foote and Cooper 2001; Müller and Kurth 2006).

- For a linearly shifted transformation, the diagonal line spans an equal distance in the horizontal and vertical dimensions but does not originate at the first feature of both recordings. Since one recording is longer than the other, the similarity matrix is rectangular (Figure 2.3).

- For a nonlinearly warped transformation, the diagonal path stretches from the first to the last features in both recordings but is not a straight line. Because the lengths of the two recordings may or may not be the same, no generalization can be made about the shape of the similarity matrix (Figure 2.4).

In the case of no transformation between event timelines, the global alingment path is a straight line from the first feature of each recording to the last feature of each recording. Since the timelines are the same length, the similarity matrix is square. By definition, the two recordings of a self-similarity matrix have identical timelines; in this case, the alignment path is called the line of identity.

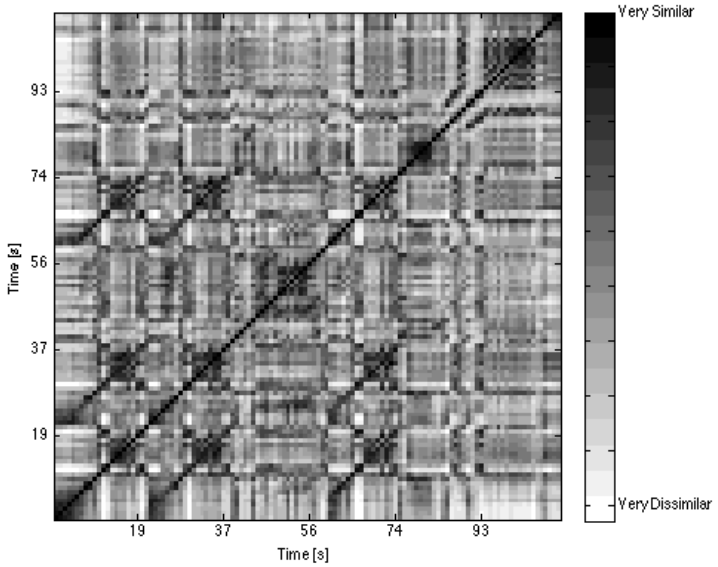

Fig. 2.1 Self-similarity matrix of a performance of Ballade from the Chopin dataset (discussed in Section 4.2.1).

Algorithmic approaches

Several different algorithms and types of algorithms have been used for audio-to-audio alignment. These algorithms, described below, include cross-correlation,

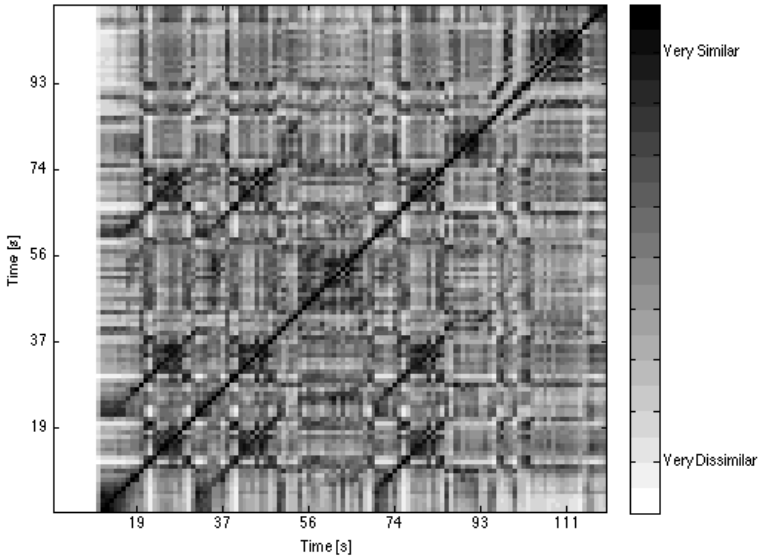

Fig. 2.2 Similarity matrix of a shifted timeline transformation. The $y-$axis recording is a performance of Ballade from the Chopin dataset; the $x-$axis recording is the same performance following ten seconds of silence.

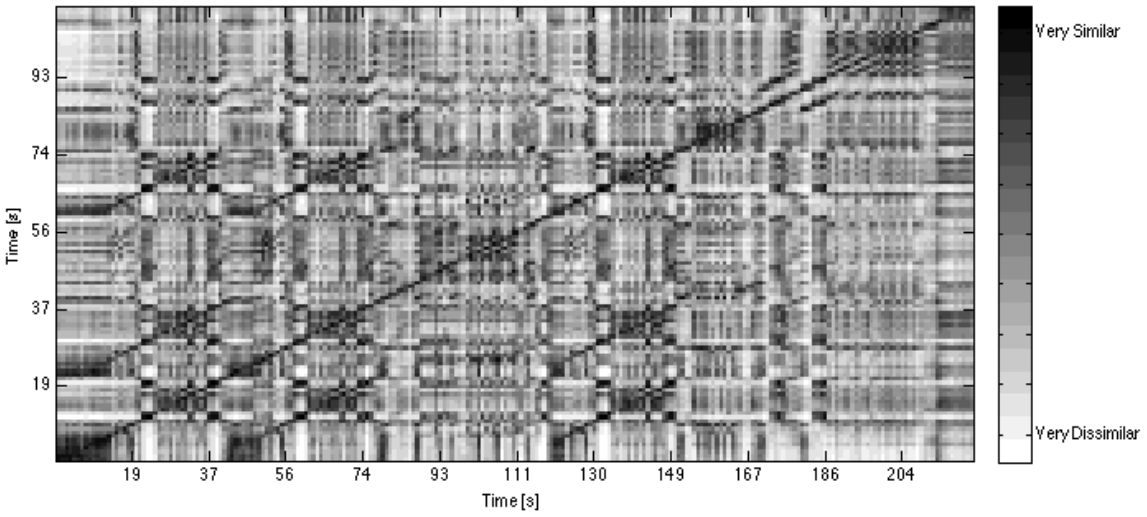

Fig. 2.3 Similarity matrix of a scaled timeline transformation. The $y-$axis recording is a performance of Ballade from the Chopin dataset; the $x-$axis recording is the same performance slowed down by a factor of two, with pitches preserved.

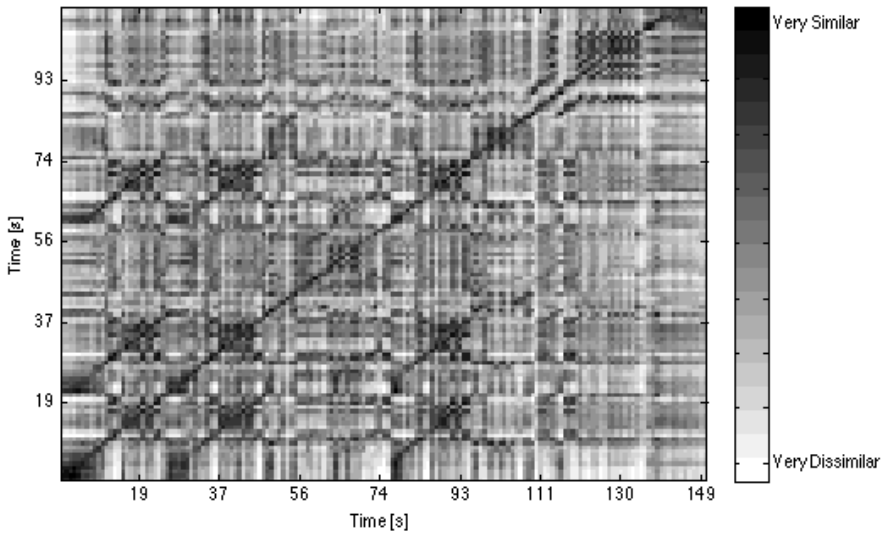

Fig. 2.4 Similarity matrix of a nonlinearly warped timeline transformation. The $x-$ and $y-$axis recordings are both performances of Ballade from the Chopin dataset (tracks 5 and 14, respectively).

recurrence quantification analysis, string matching and DTW (both DP-based), probabilistic models such as hidden Markov models and the sequential Monte Carlo method, and fingerprint clustering:

- With cross-correlation, recordings are aligned by linearly shifting one recording in relation to the other until the similarity between the two signals is maximized. Similarity between the recordings is often calculated by summing the Euclidean distances between pairs of features.

- With recurrence quantification analysis (RQA), partial alignments are calculated by identifying diagonal lines running through a similarity matrix, and taking those lines to be alignment mappings. RQA measures have been used to calculate partial alignment (Serrà et al. 2009). This is in contrast to DTW, discussed below, in which a single path through a similarity matrix is computed deterministically.

- Dynamic programming (DP) is a method of solving complex problems by breaking them down into smaller subproblems (Bellman 1952); DP-based

approaches to audio-to-audio alignment include string matching algorithms and DTW.

With string matching, two or more "strings" of features (time-series sequences) are aligned to one another through minimizing the number of edit operations (feature insertions, deletions, or substitutions) needed to transform one string to another. String alignment algorithms are used more often for alignment of symbolic rather than audio music representations; because most string-matching algorithms are designed for single-dimensional features, they are are poorly suited for aligning the multi-dimensional features most often extracted from audio recordings. Basic Local Alignment Search Tool (BLAST), developed for biological sequence alignment (Altschul et al. 1990), is one of the few string-matching algorithms adapted for audio-to-audio alignment.

With DTW, an optimal "warping path" is deterministically calculated through a similarity matrix. The warping path optimally minimizes the total similarity, or distance, between the features of the input recordings. Various constraints are often placed on the possible warping paths, to favour certain alignment characteristics and reduce computation time. Examples of constraints include forced monotonicity (preserving the order of events in each recording), forced continuity (ensuring that the path is restricted to neighbouring points), boundary conditions, and global search space constraints (upper and lower bounds placed on the warping path) (Sakoe and Chiba 1990). DTW is often used instead of measures like Euclidean distance as a tool to compare overall similarity between pairs of signals, through summing the total cost/similarity along the warping path. As seen in Table 2.1, DTW is the alignment algorithm of choice for most published audio-to-audio alignment research.

- With probabilistic algorithms like hidden Markov models (HMMs), a model is trained from recordings or a symbolic score; an alignment mapping is then calculated between the model and an input recording. This mapping is often performed via Viterbi alignment (Viterbi 1967)—a DP-based algorithm that calculates the likelihood of a sequence of events in a probabilistic model.

  Another probabilistic model used for audio-to-audio alignment is the sequential Monte Carlo (SMC) method, a simulation-based model estimation technique also known as particle filtering. With SMC, weighted "particles" (short excerpts from one recording) are used to predict the position of features

from a second recording. Because they are often implemented to run in realtime, SMC models are often used in score-following applications.

- With fingerprint clustering, k-nearest neighbor classification is used to match individual audio fingerprints (short, often reduced excerpts of an audio file) from one recording to fingerprints created from another recording. The time of the closest match from the second recording is then mapped to the time of the query fingerprint from the first recording.

Specific applications of each algorithm to audio-to-audio alignment applications are included in Table 2.1, following the summary of research presented by Section 2.3.

2.2.2 Feature selection

Feature selection is critical to the success of any audio analysis task, and alignment is no exception: the type of features selected for alignment need to relate to the musical content that forms the underlying link between the versions. For example, if two recordings contain the same melody played by two different instruments, the features chosen for their alignment need to emphasize pitch while de-emphasizing timbre. If two recordings contain a similar rhythmic pattern that is to be aligned, but those patterns are played on different pitches or by different instruments, features chosen for the recordings' alignment need to emphasize note onsets while being pitch- and timbre-invariant.

Feature extraction is the conversion of a waveform into sequential frames of single- or multi-dimensional feature vectors. Frames are generally of uniform size, and are often overlapped in order to increase feature resolution. Some alignment applications, however, use mid-level features in which each frame encompass an entire note event, beat, or even bar. Using beat-based features can reduce a nonlinear alignment problem to a linear one, so they are often used in version identification of popular music (Ellis and Cotton 2007). Event- and beat-based features require a preprocessing step to determine where each event begins and ends (called segmentation) and are thus dependent on the quality of the segmentation algorithm.

Audio features used for audio-to-audio alignment are of several varieties:

- Spectral features, as in the Short-time Fourier transform (STFT) features of spectrograms, describe the frequency content in terms of both magnitude and phase of each frame of the signal. Full spectral feature vectors are often reduced

to a smaller number of features through grouping the original frequency range of a short-time Fourier transform (STFT) into a smaller number of frequency bands that encompass frequency ranges of particular interest; the highest and lowest frequencies are sometimes thrown out altogether.

- Pitch-based features like fundamental frequency (F0) correspond to frequencies associated with pitched musical notes. They are often calculated in the frequency domain, as by binning STFT features around those pitch frequencies, as well as in the time domain, by means of signal filtering (as for peak structure distance (PSD) features).

- Chroma features, also called chroma pitch features, take advantage of octave equivalency (the perceptual similarity of notes an octave apart) to collapse an entire frequency range into a single octave. This octave often consists of the twelve equal-tempered semitone pitch classes of Western music notation (A, A♯/B♭, B, ..., G♯/A♭).[3] A number of chroma variants have been introduced, to reduce and normalize timbral differences while preserving pitch content. Variants used in audio-to-audio alignment include binary chroma (Nagano et al. 2002), the harmonic pitch class profile (Gómez 2006), chroma energy distribution normalized statistics (Müller et al. 2005a), and chroma discrete cosine transform (DCT)-reduced log pitch (Muller et al. 2009).

- Timbre-based features attempt to eliminate fundamental periodicity (pitch) from the audio while preserving spectral structural features (e.g., formants), in order to approximate the human auditory response. Timbre features are often based on perceptual scales; the nonlinear Mel-scale (Stevens et al. 1937) and the Bark scale, which corresponds to 24 critical bands of perception (Zwicker 1961), have both been used for audio-to-audio alignment.[4] Common timbre-based features include Mel-frequency cepstral coefficients (MFCCs), which are also favoured in the field of speech processing.

- Onset-based features emphasize the onset of events in a performance, often through taking the first-order difference between successive frames of other feature vectors. In this way, onset-based features can capture pitch or chroma

[3] Smaller divisions of the octave have been used; for example, Martin et al. (2012) divided the octave into 36 bins, thereby capturing a 1/3 semitone resolution.

[4] The Bark scale has been replaced in psychoacoustics by a more up-to-date measure of critical bandwidth, the equivalent rectangular bandwidth (ERB). To the best of our knowledge, however, the ERB has not yet been used as a feature for audio-to-audio alignment.

features as well as onsets, as in decaying locally adaptive normalized chroma-based onset features (Ewert and Müller 2009; Ewert et al. 2009)—another chroma variant.

Multiple features are often combined into a single feature vector, either by concatenating the different feature vectors (as in Jehan 2005) or, as when using DTW, by summing multiple similarity matrices, each created individually by different features extracted from the same audio (as in Dixon and Widmer 2005).

Specific usage of each feature in audio-to-audio alignment applications are included in Table 2.1, following the summary of research presented by Section 2.3.

Good features for audio-to-audio alignment

Several papers compare the performance of different features for audio-to-audio alignment (Hu et al. 2003; Turetsky and Ellis 2003; Müller et al. 2006; Basaran et al. 2011; Duan and Pardo 2011). Pitch-, chroma-, and onset-based features have been found to yield better alignment results than timbre-based features. Although pitch tends to slightly outperform chroma features, both pitch and chroma perform well (Hu et al. 2003; Duan and Pardo 2011). In general, chroma and chroma variants are a popular feature choice for audio-to-audio alignment, as they capture the "harmonic progression of the audio signal" (Müller et al. 2005a).

In a study on feature optimization for audio-to-audio alignment, Kirchhoff and Lerch (2011) found that combining multiple features, such as chroma and onset, tended to improve the robustness and accuracy of alignment. Weighting individual features also significantly improved results in some cases. This study highlighted the importance of selecting features based on the specific alignment application and the audio being aligned.

2.3 Audio-to-audio alignment research

Since its first appearance in 2001, audio-to-audio alignment has been used for a wide range of musical tasks. These tasks fall under two categories: similarity-based tasks and synchronization-based tasks. Similarity-based tasks (Section 2.3.1) use audio-to-audio alignment as a means to an end: to improve the measure of overall similarity (melodic, timbral, or otherwise) between two recordings. Much as in spoken word recognition, similarity-based tasks often involve determining if two recordings have the same underlying content, for example to determine if they contain different

performances of the same musical score. In contrast, synchronization-based tasks (Section 2.3.2) make use of the alignment mapping generated by the algorithm. Often, this alignment mapping is used to synchronize one recording to others for playback, sequentially or simultaneously. Sequential playback involves playing only one of the synchronized recordings at a time, but maintaining the current playback position in the event timeline when jumping between them. Simultaneous playback involves playing the synchronized recordings at the same time, after first distorting the timelines of the recordings so that they all have an identical event timeline and using a technique like phase vocoding to preserve pitch content despite the temporal distortion. Both similarity- and synchronization-based tasks are included in the following literature review.

The scope of the research included in this review is limited to works in which the alignment is performed algorithmically, rather than by hand (as in Sapp 2007), and works in which audio is aligned to other audio, rather than to symbolic music. Any of the following works that do take symbolic scores as input convert the symbolic input to audio feature vectors before sending it to the alignment algorithms, so that the alignment algorithms only ever receive input in audio feature form. This conversion is performed in one of two ways: either by directly converting the symbolic notes into feature vectors based on the pitch of a note, with or without its known harmonics (Hu and Dannenberg 2005; Kurth et al. 2007; Duan and Pardo 2011), or by first sonifying the score (synthesizing audio from it) and extracting feature vectors as for a regular recording. Tools available for sonification include Timidity[5] and the MIDI Toolbox for MATLAB,[6] among others.[7]

All research cited below is summarized in Table 2.1 at the end of the section.

2.3.1 Similarity-based tasks

Most similarity-based audio-to-audio alignment tasks focus on content-based music retrieval, in which a query recording (either an excerpt or a full recording) is used to search and return similar recordings from a database. Grosche et al. (2012) provide an overview of content-based music retrieval systems, and Skopal and Bustos (2011) provide an overview of general database query techniques for audio and symbolic music retrieval.

[5] www.onicos.com/staff/iz/timidity/index.html
[6] www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/miditoolbox
[7] www.wikipedia.org/wiki/List_of_MIDI_editors_and_sequencers

Content-based music retrieval tasks are often grouped into three classes: audio identification, audio matching, and version identification (Grosche et al. 2012). High-specificity retrieval, called audio identification, is used to find matches identical to the query. Mid-specificity retrieval, called audio matching, is used to find near-duplicates of a given query, with results that may include some variation in musical properties including tempo, rubato, dynamics, phrasing and articulation, instrumentation, and harmonization. Low-specificity searches, called version identification, return recordings that are even less similarly related to the query in regards to the above musical properties than for audio matching.

Audio identification

Audio identification involves identifying an unknown audio query by comparing it to a database of known recordings; audio-to-audio alignment is used to improve this identification. Yang (2001) performed audio identification with a database of modern and Western classical works, and Hu et al. (2003) used audio queries from albums by the Beatles to search a (sonified) MIDI database, in one of the most widely cited papers on audio-to-audio alignment. Sanguansat (2012) proposed a MSA-based query-by-humming system. (Query-by-humming tasks are identification tasks in which the query is a recording of a melody sung by a user.)

Audio matching

Audio matching, also called work recognition, is a form of audio identification that includes finding items "similar to" the query recording. The recordings sought are often originally performed from the same score, so that the variations between them are limited to tempo, rubato, and dynamics (with notes and instrumentation constant across the different performances). Nagano et al. (2002) used audio-to-audio alignment to improve similarity between results when querying a database for transpositions, re-instrumentations, and re-takes of an audio query; Müller et al. (2005a, 2005b) used short audio excerpts to search a database for alternate recordings of Western classical works, as did Camarena-Ibarrola and Chávez (2006).

Version identification

Version identification, also called cover song identification, broadens audio matching to include "semantically motivated variations as they typically occur in different interpretations of a piece of music" (Kurth and Müller 2008).

A number of version identification works have sought to identify cover songs of popular music. Ellis and Poliner (2007) searched for covers of modern popular music in a database that included recordings of live performances and had the top-performing system in the 2006 Music Information Retrieval Evaluation eXchange (MIREX) audio cover song identification competition.[8] Serrà and Gomez (2007) and Serrà et al. (2008) also used audio-to-audio alignment for cover song identification; the former was the top-performing system in the 2007 MIREX cover song identification competition. Serrà et al. (2009) expanded on this work with an RQA-based cover song detection model. Martin et al. (2012) searched the Million Song Dataset (MSD), a large database of popular music, for cover songs. Each of these research groups implemented key-invariant systems—systems that successfully identify versions of the query recording that have been transposed to other musical keys.

Other version identification systems that make use of audio-to-audio alignment are designed for music research. Antonopoulos et al. (2007) matched and ranked the similarity between rhythmic signatures extracted from traditional Greek, Rwandan, and Congolese music in order to find other recordings with those rhythmic signatures. Niedermayer et al. (2011) detected different versions of 18^{th}- and 19^{th}-century music played by historical musical automata—among them, music boxes and flute clocks. Ross et al. (2012) detected motifs in Hindustani vocal music compositions (*Bandish*), both within and across different performances, by computing similarity measures of aligned audio segments. Bohak and Marolt (2012) bypassed a need for traditional symbolic transcriptions and scores when calculating similarity measures between candidate musical stanzas from field recordings of vocal folk songs, through pairwise alignment of these stanzas.

2.3.2 Synchronization-based tasks

As discussed above, synchronization-based tasks make use of the alignment path between the recordings itself, not just the overall measure of how similar the recordings are post-alignment. This alignment path is often exploited for both performance tracking, the time-linking of a recording to a score or other recording, and performance analysis, in which different performances are aligned for the purpose of analyzing differences and similarities among them, as for motif detection. Depending on the algorithm employed, synchronization can happen in realtime

[8] MIREX is an annual competition for solving music information retrieval (MIR) tasks using standardized evaluation frameworks (Downie 2008).

(online alignment) or not (offline alignment).

Synchronization-based audio-to-audio alignment research, grouped by task is as follows, grouped by task: audio-score alignment, joint structure analysis, multimodal browsing, performance analysis, and studio engineering.

Audio-score alignment

Audio-score alignment, alternatively called performance-score alignment or score-following when in realtime, maps an audio performance to a symbolic score. On sonified symbolic scores, general audio-to-symbolic alignment research for offline alignment to scores has been carried out by Meron and Hirose (2001), Orio and Schwarz (2001), Dannenberg and Hu (2003), Devaney et al. (2009), and Niedermayer and Widmer (2010). General audio-to-symbolic alignment research for online alignment to scores has been carried out by Dixon (2005b), Dixon (2005a), Camarena-Ibarrola and Chávez (2010), Montecchio and Cont (2011a), Duan and Pardo (2011), Carabias et al. (2012), and Xiong and Izmirli (2012). Additional research featuring audio-score alignment as an internal step but not the end goal will be noted in later sections.

Alignment for multimodal browsing

Multimodal alignment involves synchronizing multiple modes of music (e.g., audio, symbolic, and/or lyrics if applicable) to facilitate user interaction and browsing. Often, the various modes and/or different versions of a piece and its performances are linked to ensure consistent browsing of all versions at the same time. During playback, it is then possible to jump between these versions without pausing the underlying timeline.

The SyncPlayer is one audio-to-audio alignment-based multimodal music interface (Kurth et al. 2005). Within the SyncPlayer framework, Müller et al. (2006) worked to speed up audio-to-audio alignment for performance synchronization, to facilitate faster performance browsing. The SyncPlayer has also been used to perform image-audio synchronization: a scanned image of a musical score taken as input is aligned both to audio recordings of that score and to its symbolic score, as extracted from the image through optical music recognition (OMR) (Kurth et al. 2007; Fremerey et al. 2008; Fremerey et al. 2009). In this way, recordings are linked to specific pixels in the image, so that during playback the time in the audio is tracked in the image of the score.

Joint structure analysis

Joint structure analysis involves studying the similarities and differences in musical structure between two recordings of the same underlying piece that have variations in global structure; for example, one recording may contain more verses or longer cadenzas than another. Joint structure analysis involving audio-to-audio alignment usually makes use of partial alignment, as in Müller and Appelt (2008) and Müller and Ewert (2008). Ewert et al. (2009) used audio-to-symbolic alignment to identify both frequently varied and usually stable (unvaried) passages in a collection of musical recordings by the Beatles. Tabus et al. (2012) presented a general method for finding and aligning similar segments within a song.

Performance analysis

Performance analysis, also called expressive performance extraction, is the analysis of different interpretations of a piece. As discussed by Devaney et al. (2011), automated performance analysis is still a relatively young musicological subfield. Audio-to-audio alignment has been used to analyze and compare electroacoustic music (Orio and Zattra 2007) as well as to assist in automatic extraction of harmonic information and to find "harmonically stable" musical passages in cover songs (Konz and Müller 2012).

Similarly, a number of performance analysis tasks use audio-to-symbolic alignment to facilitate segmentation of audio into notes or phrases, or to assist in further score-based audio analysis: creating a database of MIDI transcriptions, to later be used as training data for transcription algorithms (Turetsky and Ellis 2003); facilitating the segmentation of audio into small units to be used in a sound synthesis system (Schwarz 2003); estimating note boundaries, as a first step in automating audio segmentation (Hu and Dannenberg 2005); segmenting field recordings of vocal folk songs into stanzas (Müller et al. 2009); investigating asynchronies in alignment of polyphonic a cappella vocal recordings to scores (Devaney and Ellis 2009); studying differences in tempo, key, tuning, and melody between stanzas in field recordings of vocal folk songs (Müller et al. 2010); for studying the intonation in singers (Devaney et al. 2011); and as a first step in source separation (identifying and isolating a particular musical line from a polyphonic mix) (Ewert and Müller 2012).

Studio engineering

There are several studio engineering tasks that use audio-to-audio alignment to automate or semi-automate laborious tasks normally performed by music producers and audio engineers. Dannenberg (2007) used audio-to-symbolic alignment to annotate notes and facilitate other studio engineering tasks like balancing instrument levels in the Intelligent Audio Editor (IAED) environment. Basaran et al. (2011) proposed a method for aligning multiple fragments of a single performance recorded on different microphones with different quality levels, as when multiple recordings are taken by different people at the same concert. Montecchio and Cont (2011b) aligned multiple takes of a musical recording to the full performance, a tedious but critical task sound mixing engineers frequently perform. Gerber et al. (2012) aligned cover songs recorded as multitrack source mixes to original audio to aid in source separation.

Audio fingerprint alignment

Audio-to-audio alignment has also been applied to audio fingerprints—unique feature sets derived from short excerpts of recordings used in audio matching. Harte (2010) aligned fingerprints to their counterparts in an evaluation database, to prevent asynchronies between the original, timing-critical metadata annotations created for audio in the database and the local copies of those same recordings. Ramona and Peeters (2011) matched purposefully distorted fingerprint queries to an audio database to check for fingerprint mismatches.

2.3.3 Application-agnostic audio-to-audio alignment

The remainder of audio-to-audio alignment research focuses on alignment for music-related tasks in general, and neither focuses on nor demonstrates a particular use-case.

To investigate the effects of feature selection given different types of acoustic similarity (timbre, rhythm, and pitch) between recordings, Jehan (2005) sonified a score with systematic differences in timbre, rhythm, and pitch, and then compared these recordings to one another. To study the systematic effects of varying algorithms (cross-correlation versus DTW) and algorithmic parameters (chroma resolution and similarity, transposition, beat tracking, and DTW constraints), Serrà et al. (2008) re-implemented the works of Ellis and Poliner (2007) and Serrà and Gomez (2007) for cover song identification in a collection of commercial songs of different musical

genres. Ewert et al. (2009) and Ewert and Müller (2009) investigated feature optimization for synchronizing of polyphonic music. Thomas et al. (2012) sped up audio matching in a database of Western classical music by segmenting all the audio into equal-sized, overlapping segments and then precomputed and stored their DTW similarity.

Table 2.1 summarizes the research from this section, along with the algorithms and features implemented by each.

Table 2.1: Research on audio-to-audio alignment of music

	Algorithm[i] (⋆ = online)	Features[ii]	Frame [Overlap][iii] [ms]	Musical input[iv] [s]
Audio identification				
Yang (2001)	DTW	Onset	46 [50%]	$A_{30–60} \Leftrightarrow A_{30–60}$
Hu et al. (2003)	DTW	Chroma, Pitch, $\text{Timbre}_{\text{MFCC}}$	250 [0%]	$A \Leftrightarrow S$
Sanguansat (2012)	M-DTW	Pitch contour	Event [*NA*]	$A_{10} \Leftrightarrow A_{10\text{ (multiple)}}$
Audio matching				
Nagano et al. (2002)	DTW	Chroma_{B}	Beat [0%]	$A \Leftrightarrow A_{19}$
Müller et al. (2005a)	DTW	$\text{Chroma}_{\text{CENS}}$	4100 [76%]	$A,S \Leftrightarrow A_{10–30}$
Müller et al. (2005b)	DTW	$\text{Chroma}_{\text{CENS}}$	4100 [76%]	$A,S \Leftrightarrow A_{10–30}$
Camarena-Ibarrola and Chávez (2006)	⋆DTW	$\text{Onset}_{\text{B+BARK}}$	1500 [50%]	$A \Leftrightarrow A$
Version identification				
Ellis and Poliner (2007)	CC	Chroma	Beat [0%]	$A \Leftrightarrow A$
Serrà and Gomez (2007)	DTW	Chroma_{36}	93 [50%]	$A \Leftrightarrow A$
Antonopoulos et al. (2007)	DTW	$\text{Timbre}_{\text{CH+MFCC}}$	93 [88%]	$A,A_{\text{seg}} \Leftrightarrow A,A_{\text{seg}}$
Serrà et al. (2008)	DTW	$\text{Chroma}_{12,24,36}$	93 [50%]	$A \Leftrightarrow A$
Serrà et al. (2009)	CRP	$\text{Chroma}_{\text{HPCP}}$	464 [0%]	$A \Leftrightarrow A$
Niedermayer et al. (2011)	DTW	Chroma	4096 *x* [75%]	$A_{25} \Leftrightarrow A_{25}$
Ross et al. (2012)	DTW	Pitch	10 [*NA*]	$A_{\text{seg}} \Leftrightarrow A_{\text{seg}}$
Martin et al. (2012)	BLAST	Chroma_{36}	743 [50%]	$A \Leftrightarrow A$
Ibid.	BLAST	Chroma	80–300 [0%]	$A \Leftrightarrow A$
Bohak and Marolt (2012)	DTW	Chroma	50 [*NA*]	$A_{\text{seg}} \Leftrightarrow A_{\text{seg}}$
Audio-score alignment				
Orio and Schwarz (2001)	DTW	$\text{Pitch}_{\text{PSD}}$	*NA*	$A \Leftrightarrow S$
Meron and Hirose (2001)	DTW	Pitch	45 [67%]	$A \Leftrightarrow S$
Dannenberg and Hu (2003)	DTW	Chroma	250 [0%]	$A \Leftrightarrow S$
Dixon (2005a)	⋆DTW	Onset	20 [0%]	$A \Leftrightarrow A$
Dixon (2005b)	⋆DTW	$\text{Onset}_{\text{Pitch}}$	46 [57%]	$A \Leftrightarrow A$
Devaney et al. (2009)	HMM_{DTW}	$\text{Pitch}_{\text{PSD}}$	10 [7%]	$A \Leftrightarrow S$
Camarena-Ibarrola and Chávez (2010)	⋆FC	$\text{Onset}_{\text{Bark}}$	185 [75%]	$A \Leftrightarrow A_{0.185}$
Niedermayer and Widmer (2010)	DTW	Chroma	4095 *x* [25%]	$A \Leftrightarrow S$
Duan and Pardo (2011)	⋆SMC	Pitch, Chroma	46 [78%]	$A \Leftrightarrow S$

Continued on the following page.

Table 2.1 – Continued

	Algorithm[i] (⋆ = online)	Features[ii]	Frame [Overlap][iii] [ms]	Musical input[iv] [s]
Montecchio and Cont (2011a)	⋆SMC	*NA*	*NA*	A ⇔ A,S
Carabias et al. (2012)	⋆DTW	*NA*	*NA*	A ⇔ S
Xiong and Izmirli (2012)	⋆SMC	Chroma	*NA*	A ⇔ A
Joint structure analysis				
Müller and Appelt (2008)	DTW	$\text{Chroma}_{\text{CENS}}$	1000 [*NA*]	A ⇔ A
Müller and Ewert (2008)	DTW	$\text{Chroma}_{\text{CENS}}$	1000 [*NA*]	A,S ⇔ A,S
Ewert et al. (2009)	DTW	Chroma	500 [*NA*]	A ⇔ S
Tabus et al. (2012)	HMM	Chroma_{B}	2048 *x* [25%]	A ⇔ A
Multimodal browsing				
Müller et al. (2006)	DTW	$\text{Chroma}_{\text{CENS}}$	100–9000 [*NA*]	A ⇔ A
Kurth et al. (2007)	DTW	Chroma	*NA*	A ⇔ OMR
Fremerey et al. (2008)	DTW	$\text{Chroma}_{\text{CENS}}$	1000 [*NA*]	A ⇔ $\text{OMR,S}_{\text{seg}}$
Fremerey et al. (2009)	DTW	Chroma	*NA*	A ⇔ OMR
Performance analysis				
Schwarz (2003)	DTW	$\text{Pitch}_{\text{PSD}}$	*NA*	A ⇔ S
Turetsky and Ellis (2003)	DTW	FFT, Onset, FFT+Onset	93 [50%]	A ⇔ S
Hu and Dannenberg (2005)	DTW	Chroma	50 [0%]	A ⇔ S
Orio and Zattra (2007)	DTW	FFT	*NA*	A ⇔ A
Müller et al. (2009)	DTW	$\text{Chroma}_{\text{CENS}}$	100 [*NA*]	A ⇔ S
Devaney and Ellis (2009)	DTW	$\text{Pitch}_{\text{PSD}}$	*NA*	A ⇔ S
Müller et al. (2010)	DTW	$\text{Chroma}_{\text{B+Pitch}}$	100 [*NA*]	A_{seg} ⇔ S_{seg}
Devaney et al. (2011)	HMM_{DTW}	*NA*	*NA*	A ⇔ A
Konz and Müller (2012)	DTW	Chroma	Beat, Bar [*NA*]	A ⇔ S
Ewert and Müller (2012)	*NA*[v]	*NA*	*NA*	A ⇔ S
Studio engineering				
Dannenberg (2007)	DTW	Chroma	5 [*NA*]	A ⇔ S
Montecchio and Cont (2011b)	SMC	*NA*	*NA*	A ⇔ A_{15}
Basaran et al. (2011)	GM	FFT_{Bark}, $\text{Onset}_{\text{BARK}}$	25 [0%]	$\text{A}_{\text{2–60 (multiple)}}$
Gerber et al. (2012)	*NA*[vi]	*NA*	Beat [*NA*]	A ⇔ A

Continued on the following page.

Table 2.1 – Continued

	ALGORITHM[i] (⋆ = online)	FEATURES[ii]	FRAME [OVERLAP][iii] [ms]	MUSICAL INPUT[iv] [s]
Audio fingerprint alignment				
Harte (2010)	⋆CC	$\text{Waveform}_{\text{B}}$	1 *x* [0%]	$\text{A} \Leftrightarrow \text{A}_{0.0045}$
Ramona and Peeters (2011)	FC	$\text{Timbre}_{\text{dnSTFT}}$	2000 [98%]	$\text{A} \Leftrightarrow \text{A}_{\text{seg}}$
Application-agnostic alignment				
Jehan (2005)	DTW	$\text{Chroma+FFT}_{\text{Bark}}$ +Timbre	Events, Beats [*NA*]	$\text{A}_{\text{seg}} \Leftrightarrow \text{A}_{\text{seg}}$
Ewert et al. (2009)	DTW	$\text{Chroma, Onset}_{\text{CH}}$, $\text{Chroma+Onset}_{\text{CH}}$	20 [*NA*]	$\text{A} \Leftrightarrow \text{S}$
Ewert and Müller (2009)	DTW	$\text{Chroma, Onset}_{\text{CH}}$, $\text{Chroma+Onset}_{\text{CH}}$	20 [0%]	$\text{A} \Leftrightarrow \text{S}$
Thomas et al. (2012)	DTW	$\text{Chroma}_{\text{CRP}}$	1000 [0%]	$\text{A} \Leftrightarrow \text{A}_{25\text{–}125}$

General notes: *NA* means the information was not given. Works cited twice in the chapter are listed with their first appearance. Only pertinent problems are included in the table; for example, Ewert et al. (2009) uses both DTW and DP-based partial-alignment algorithms, but only DTW is included in this table as partial-alignment strategies are beyond the scope of this thesis.

[i] **Notes on algorithms:** Abbreviations are as follows: CC is cross-correlation; CRP are cross recurrence plots; FC is fingerprint clustering; GM are general generative models; M-DTW is multi-dimensional DTW; and SMC is the sequential Monte Carlo method. For HMMs, a subscript specifies the initialization prior, if given. These algorithms are discussed in Section 2.2.1.

[ii] **Notes on features:** Abbreviations are as follows: B are binary features, BARK indicates features that are based on the Bark perceptual scale; CENS are chroma energy distribution normalized statistics; CH are generic chroma; CRP are chroma discrete cosine transform (DCT)-reduced log pitch features; dnSTFT is double-nested STFT; HPCP are harmonic pitch class profile; MFCC are Mel-frequency cepstral coefficients; and PSD are peak structure distance. Subscripted numbers indicate the number of FFT bands or bins used; in the case of chroma features, subscripted numbers are the number of chroma bins per octave (default is twelve). Onsets are FFT-based unless otherwise indicated (e.g., Onset_{CH} are chroma-based onsets). Different types of features used independently in the same paper are separated by commas. Plus signs indicate multiple variants of a single feature type (e.g., $\text{Onset}_{\text{B+BARK}}$ means binary onset features based on the Bark scale). These features and feature variants are discussed in Section 2.2.2.

[iii] **Notes on frame:** The overlap between subsequent frames is given in brackets. Values with "*x*" are measured in number of samples rather than in milliseconds. Where frame length is given but overlap is *NA*, it is likely that zero overlap was used but not explicitly stated.

[iv] **Notes on musical input:** Musical input indicates the type of inputs to the problem, before conversion to audio (in the case of symbolic or image): A means audio, S means symbolic, and OMR means image. Excerpt duration is given (in seconds) as subscript; if no duration is given, the full duration of the recording was used—"seg" indicates an excerpt of unspecified duration.

[v] Cited method of Ewert et al. (2009).

[vi] Used the commercial software Beat Detective (see section 2.4.1).

2.4 Software for aligning musical recordings

To conclude the chapter, this section briefly covers both commercial and non-commercial software (libraries, toolkits, and standalone programs) designed for automated alignment of musical audio. Programs that are intended for generic signal alignment are not included,[9] nor are programs that rely on hand-annotation of every event or beat to be aligned in the recordings,[10] or programs that do not account for nonlinear timeline transformations, such as software that compensates for linear timeline shifts created by microphone delays when recording.[11] The algorithm and features used by each program are summarized in Table 2.2.

2.4.1 Commercial software

Three commercial software products are designed to align musical audio with underlying timelines that are potentially nonlinearly related. Zplane's Audio-to-Audio Alignment Kit (AtAAK!), advertised for musical audio-to-audio timing adjustment as well as alignment of overdubbed-to-original speech, uses user-determined combination of pitch, timbre, loudness, and onset times to time-align two files, then time-stretches one signal to synchronize with the other.[12] VocALign, by Synchro Arts, is designed to align pairs of vocals (lead or backing), instrumental, and dialogue tracks.[13] Beat Detective, a feature in Avid's digital audio workstation software Pro Tools, detects the beats in each audio file input and then shifts the beats of one to line up with the beats of another.[14]

2.4.2 Non-commercial software

Several non-commercial, open-source programs for audio-to-audio alignment also exist. A feature-agnostic MATLAB script for offline DTW between two audio files has been made available, along with a script for performing audio-to-symbolic alignment

[9] E.g., data-agnostic alignment tools like basic DTW implementations: `www.wikipedia.org/wiki/Dynamic_time_warping#Open_Source_software`.

[10] E.g., the AudioSnap feature of Cakewalk's SONAR X2 post-production software, which requires human beat annotation: `www.cakewalk.com/Products/SONAR/`).

[11] E.g., SoundRadix's delay-based Auto-Align, a production VST plug-in: `www.soundradix.com/products/auto-align`).

[12] `www.zplane.de/index.php?page=description-ataak`

[13] `www.synchroarts.com/index.php?PAGEID=products&ID=vocalign`

[14] `www.avid.com/US/products/Pro-Tools-Software/features`

(Ellis 2003; Ellis 2008; as introduced in Turetsky and Ellis 2003).[15,16] Scorealign, an alignment implementation that has since been incorporated into the Port Media project, performs both audio-to-audio and audio-to-symbolic alignment; it is available in the form of MATLAB scripts and also as a feature in the experimental branch of open-source audio editor Audacity.[17,18,19,20] The Automated Music Performance Analysis and Comparison Toolkit (AMPACT) is an audio-to-symbolic MATLAB toolkit tailored for performance comparison of vocal audio (Devaney et al. 2009; Devaney 2011; Devaney et al. 2011), which allows the user to choose between DTW and HMM algorithms and incorporates the MATLAB scripts for DTW introduced above (Ellis 2008).[21] SyncPlayer, the multimodal browsing software introduced in Section 2.3.2, is available as a standalone alignment program (Kurth et al. 2005).[22] Finally, the Music Alignment Tool CHest (MATCH) aligns multiple recordings through iterative pairwise alignment, by arbitrarily picking one reference track and aligning each of the other recordings to that reference (Dixon 2005b; Dixon and Widmer 2005).[23] In addition to the standalone program, MATCH is available as a Vamp plug-in for use with Sonic Visualiser, Audacity 2, and Sonic Annotator.[24,25,26,27]

[15] www.ee.columbia.edu/~dpwe/resources/matlab/dtw/
[16] www.labrosa.ee.columbia.edu/matlab/alignmidiwav
[17] www.cs.cmu.edu/~music/alignment/
[18] www.sourceforge.net/apps/trac/portmedia/wiki
[19] www.audacity.sourceforge.net/
[20] www.wiki.audacityteam.org/wiki/Experimental_Features
[21] www.ampact.org
[22] www-mmdb.iai.uni-bonn.de/projects/syncplayer/download.php
[23] www.eecs.qmul.ac.uk/~simond/match/
[24] www.vamp-plugins.org/download.html
[25] www.sonicvisualiser.org
[26] www.audacity.sourceforge.net/download/features-2.0
[27] www.omras2.org/sonicannotator

Table 2.2 Audio-to-audio alignment software

	Developer	Application	Algorithm (⋆ = online)	Features
Commercial				
Audio-to-Audio Alignment Kit (AtAAK!)[i]	zplane	C/C++ library	*NA*	Loudness, timbre, pitch, and/or onsets
Pro Tools Beat Detective[ii]	Avid	Standalone	*NA*	*NA*
VocALign[iii]	Synchro Arts	Plug-in or standalone	*NA*	*NA*
Open source				
Automated Music Performance Analysis and Comparison Toolkit (Ampact)[vi]	Devaney et al. (2009)	MATLAB toolkit	HMM, DTW	$\text{Pitch}_{\text{PSD}}$
DTW for audio[iv]	Ellis (2003)	MATLAB script	DTW	Agnostic
Music Alignment Tool CHest (Match)[vii,viii]	Dixon (2005b)	Standalone (GUI, CLI), Vamp plug-in	⋆DTW	STFT, Onsets
scorealign[ix]	Hu et al. (2005)	C++ (API, CLI)	DTW	Chroma
scorealign (original)[x]	Hu et al. (2005)	MATLAB script	DTW	Chroma
SyncPlayer [xi]	Kurth et al. (2005)	Standalone (GUI)	DTW	Chroma

Note: *NA* means the given specification is unavailable.

[i] www.zplane.de/index.php?page=description-ataak
[ii] www.avid.com/US/products/Pro-Tools-Software/features
[iii] www.synchroarts.com/index.php?PAGEID=products&ID=vocalign
[iv] www.ee.columbia.edu/~dpwe/resources/matlab/dtw/
[v] www.labrosa.ee.columbia.edu/matlab/alignmidiwav
[vi] www.ampact.org
[vii] www.eecs.qmul.ac.uk/~simond/match/
[viii] www.vamp-plugins.org/download.html
[ix] www.cs.cmu.edu/~music/alignment/
[x] www.portmedia.sourceforge.net/
[xi] www-mmdb.iai.uni-bonn.de/projects/syncplayer/download.php

3—Applying the CPM to musical audio

This chapter covers the algorithm at the heart of this thesis: the continuous profile model (CPM). The CPM is a generative Markovian model, designed to simultaneously normalize and align multiple related time-series signals (Listgarten et al. 2005; Listgarten 2007). Alignment with the CPM has been used in a variety of applications. Signals from liquid chromatography–mass spectrometry (LC–MS), a chemistry technique used to identify biological components in a mixture, have been aligned and compared in order to detect the presence or absence of specific proteins in human serum (Listgarten et al. 2006). Signals pertaining to behavior, such as movement tracking in video recordings, have been aligned in order to find and recognize behavior deviations in persons with bipolar disorder (Amor and James 2010). Most recently, alignment of movement data has been used to optimize grasping motions in robots (Wang 2012). Although the CPM has been used to align audio recordings of speech, in a proof-of-concept in its introductory publication (Listgarten et al. 2005), to the best of our knowledge it has not previously been used to align audio recordings of music.

The CPM algorithm is explained in Section 3.1; audio-to-audio alignment with the CPM is implemented in Section 3.2.

3.1 The CPM algorithm

The CPM is a generative model that operates under the assumption that each of the input time-series signals is a noisy, timeline-warped version of a common underlying signal—the latent trace. In other words, each input signal is a non-uniformly subsampled version of the latent trace, with local (amplitude) rescaling and the addition of noise. The latent trace acts as a reference timeline to link all of input signals to one another, as each point on the latent trace maps to a sample (or multiple samples) in each of the original signals. The latent trace does not necessarily

have a consistent sample rate at all, and the sample rate of the latent trace has no meaningful relation to the sample rate of any of the input recordings: it is merely a way to map among them.

The CPM can be considered to be an HMM with an additional set of parameters that are related to the latent trace. In general, an HMM is a statistical model for situations in which an unobservable, hidden stochastic process is observable through a second stochastic process that produces a sequence of observations. In the case of time-series signals, these observations are the signal's features. An HMM is characterized by five parameters: the number of hidden states in the model, the number of possible kinds of observations for each state, the probability distribution for transitioning from each state to any other state (the state transition probability distribution), the observation symbol probability distribution for each state, and the initial state distribution.

In the CPM, hidden states contain both a time component and a scale component. The time component controls how an input signal relates to the time steps in the latent trace, which in turn serves to link all input signals to a single latent trace. The scale component controls how the amplitude of each state in an input signal is scaled in relation to the latent trace, which accounts for amplitude variation across all input signals and within a single input signal. While a multiclass variant of the CPM, the hierarchical Bayesian continuous profile model (HB-CPM), has been made available, this thesis focuses on the basic, single-class CPM—more specifically called the expectation maximization-CPM (EM-CPM).

Listgarten (2007), the CPM's developer, likens the CPM to a fancy tape player:

> It can be helpful to draw an analogy between generating an observed time series in the CPM and playing an audio cassette tape in a special type of tape player which in addition to the regular volume knob, also has a 'speed' knob. When playing a tape in this machine, one can keep one hand on each knob, controlling both the speed at which the audio is played, and also the volume at which it is played. Similarly, one can think of the generative process of a CPM as having a 'speed' knob and an 'amplitude' knob. When generating a single observed time series in a CPM, one can change either knob, or both at once. When the speed is increased (by moving more quickly through latent time with the hidden time states), less of the underlying latent trace will be represented in the observed time series, and when the speed is decreased (by moving more

slowly through latent time), more of that portion of the latent will be represented in the observed time series. The observed signal emitted at a particular latent time point is scaled by a factor proportional to the setting on the amplitude knob.

Following this analogy, a group of similar signals, such as different versions of the same song, can be considered to have been created using the same 'tape,' but played back with different, varying values of speed and amplitude for each version—not to mention different amounts of background noise.

Alignment using the CPM is executed in two stages: in the training stage, parameters for the CPM (and therefore the latent trace) are learned from the original input signals; in the alignment extraction stage, each original input signal is pairwise aligned to this newly modeled latent trace.

3.1.1 Training

During training, CPM parameters, including those of the latent trace, are learned from the input signals through the expectation-maximization algorithm (EM). EM is an unsupervised parameter-estimation method for iteratively improving the parameters that characterize a statistical model (Dempster et al. 1977): a model is first calculated from input parameters, and then improved, performance-maximizing parameters are calculated for that model. This estimation and maximization is then repeated using each newly maximized set of parameters, until either a best-fit threshold has been crossed or a predetermined maximum number of iterations have been performed.

Input signals can be single- or multi-dimensional, although all signals must have features with the same number of dimensions. Along with the latent trace, outputs from this training step include the Markovian transition probabilities for both the hidden scale and time states, as well as a global scaling factor and a noisiness level for the observed input signals.

3.1.2 Alignment extraction

During the alignment extraction stage, each of the original input signals is individually aligned to the timeline of the latent trace. This alignment is performed with Viterbi alignment, a dynamic programming technique used to find the best state

sequence through a stochastic model (such as the the CPM of the latent trace) for a given observation sequence, the input signal.

The output from this step is in the form of a mapping from each sample in the input signals to the samples of the latent trace. The resultant mapping among all signals is therefore in latent time, which is not independently meaningful as it has neither a real-world relationship to the timing of any one particular input signal nor a fixed sampling rate.

3.2 Musical implementation

The main consideration when applying the CPM to music is feature choice. The CPM is generalized to take any set of time-series sequences as input, in the form of a three-dimensional data array: the first dimension (rows) contain each sequence, the second dimension (columns) represents the consecutive samples, and the third dimension contains the multi-dimensional feature vector. For example, an input array of ten sequences of 400 samples and six-dimensional features has the dimensions $10 \times 400 \times 6$, and the data point at index [4, 3, 2] is the second feature of the third sample of the fourth recording. CPM operations on the input data include standard matrix operations such as transposition, multiplication, and inversion.

Pitch chroma features are a natural preliminary choice for aligning musical audio with the CPM since they have been found to yield good alignment of musical audio by other algorithms, both in general and on the Chopin dataset used in this thesis (Kirchhoff and Lerch 2011). As introduced in Section 2.2.2, a pitch chroma vector has twelve values, the strengths of each pitch class (A, A♯/B♭, B, ..., G♯/A♭) in the audio frame. The input array of musical recordings to the CPM is therefore a three-dimensional data array where the first dimensional represents each recording, the second represents the consecutive feature frames, and the third contains the twelve-dimensional feature vector. The size of the CPM input array for an alignment of musical recordings using pitch chroma features is therefore [the number of recordings] × [the number of frames in each recording] × 12.

3.2.1 Musical implementation

In this thesis, alignment with the CPM was performed with an open-source, MATLAB-based implementation of the CPM developed by the algorithm's author (Listgarten 2007).[1] The CPM paramater choices are listed in Table 3.1.

[1] The toolbox can be found at `www.cs.toronto.edu/~jenn/CPM/`. A patch created to update the toolbox to MATLAB's more recent 64-bit API can be found at `www.music.mcgill.ca/~hannah/CPM`.

Pitch chroma features were extracted with the Chroma Toolbox for MATLAB (Müller and Ewert 2011). As part of the feature extraction process, the audio was first converted to mono, normalized to have a maximum amplitude of one, and downsampled from 44.1 kHz to 22.05 kHz. A window size of 2408 samples with a 50% overlap was used, to give a feature resolution of 46 ms. Recordings were zero-padded so that each contained 600 feature frames, a duration of 27.6 seconds.[2]

An example of the output generated by a CPM alignment, both the training and the alignment extraction, is presented in Figure 3.1. Note that the output alignment mapping is in latent time, which contains just over twice as many samples as the original input files.

Table 3.1 Implementation variables

VARIABLE	VALUE
Maximum number of EM iterations	10
EM log-likelihood difference threshold	1.000e-04
Length of each time series (number of samples)	600
Number of bins (time-series dimensionality)	12
Number of HMM time states[i]	1260
Number of HMM scale states	7
Total number HMM states[ii]	8820
Use scaling spline	FALSE
Learn HMM emission variance	TRUE
Learn latent trace	TRUE
Learn scale transitions probabilities	FALSE
Learn time transitions probabilities	FALSE
Learn scaling parameter(s)	TRUE
λ (smoothing penalty)	FALSE
ν (inter-class penalty)	FALSE
Number of classes	1

[i] Auto-generated from length of time series.
[ii] Number of HMM time states $*$ number of HMM scale states.

[2] Due to normalization, all twelve bins in each "silent" zero-padded frame summed to one rather than zero, so the matrix was actually "$1/12^{\text{th}}$-padded" rather than zero-padded.

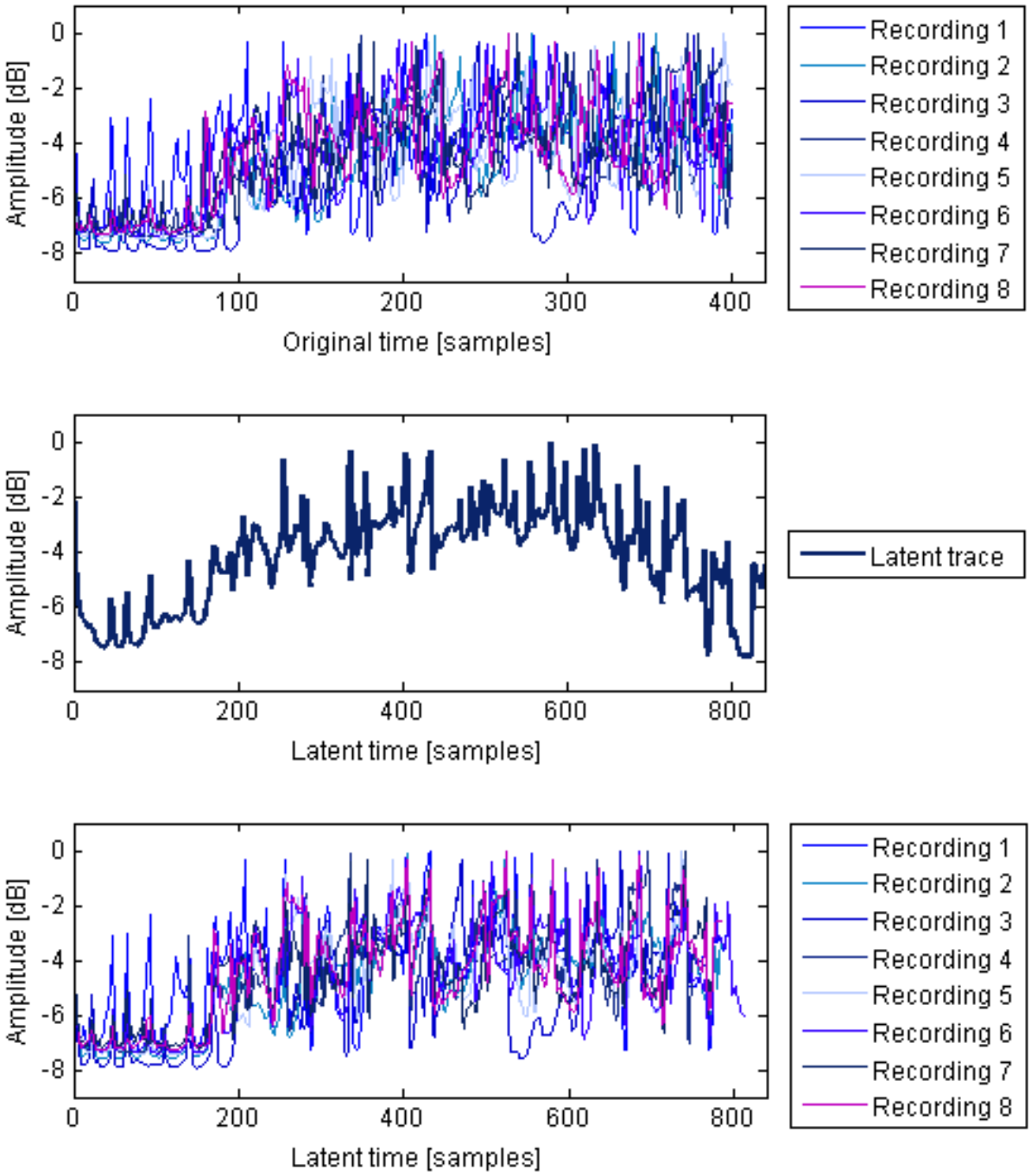

Fig. 3.1 Alignment output from multiple alignment by the CPM using the CPM Toolbox for MATLAB: eight original input recordings with their original timelines (top), the learned latent trace (middle), and the original recordings re-mapped to the timeline of the latent trace (bottom). The aligned recordings are the first 18 seconds of Ballade from the Chopin dataset. Each figure plots the amplitude of the signal's energy, in decibels (dB). In the top figure, each sample along the x–axis has a 46 ms duration.

4—Evaluation methodology

The methodology for evaluating alignment algorithms is non-trivial, as the choices of evaluation metric, dataset, and features often affect algorithm performance. In this thesis, pairwise and multiple alignment of musical recordings by the CPM is compared to pairwise and iterative pairwise alignment by DTW, by means of a straightforward, quantitative assessment that measures the distance between the algorithmically calculated alignments and the ground-truth alignment. After presenting approaches to alignment evaluation (Section 4.1) and dataset considerations (Section 4.2), the chapter describes the experimental design of the thesis—including the evaluation metrics, alignment mapping manipulations, and software implementations utilized (Section 4.3).

4.1 Approaches to evaluation of musical alignment

Evaluations of audio-to-audio alignment algorithms are often tailored to real-world use cases, in that they evaluate the results of a particular task that involves an alignment algorithm. Different approaches are taken in the evaluation of similarity-based alignment tasks as compared to synchronization-based tasks. The main difference between these tasks, as was discussed in Chapter 2, is that similarity-based tasks use alignment to improve an overall similarity measure between two recordings, often for purposes of retrieval, while synchronization-based tasks make use of the warping paths created by the algorithms, often for purposes of score following or comparison of musical events across different recordings. Correspondingly, similarity-based evaluations focus on an entire retrieval system, while synchronization-based tasks focus on the accuracy of the generated warping path.

Evaluations of similarity-based tasks generally focus on the overall success of the content-based music retrieval system making use of an alignment algorithm. This success is measured objectively, in terms of successful retrieval rates by the

whole system: rates of correctly identifying a query, incorrectly identifying a query, neglecting to identify a query, and correctly rejecting a query that does not have a match in the system's database. The MIREX cover song identification and query-by-humming competitions take this approach (Downie and Bay 2008).[1,2] In the cover song identification task, for example, system performance is measured by the number of correct audio matches in the top ten items that the system retrieves as most similar to the query.

In contrast, evaluations of synchronization-based tasks focus on how well the generated alignment path follows the known ground-truth ("true") alignment path, and both subjective and objective evaluations of synchronization-based tasks have been implemented. Turetsky and Ellis (2003) took a subjective approach by means of a listening-based evaluation: after automatically aligning synthesized symbolic (MIDI) scores to audio recordings via DTW, the original MIDI tracks were re-synthesized to have the same timeline as the recording to which they had been aligned. Researchers then listened to the original recording and the re-synthesized MIDI recording simultaneously and rated the alignment on a subjective five-point "quality of alignment" scale. Dixon and Widmer (2005) took an objective evaluation approach: alignment was automatically performed on recordings for which a ground-truth alignment path had previously been determined. The overall accuracy of each alignment was then calculated by measuring the distance (in a two-dimensional similarity matrix) from points on the ground-truth alignment to points on the automatically generated alignments. This approach, also reproduced by Kirchhoff and Lerch (2011), forms the basis of the evaluation performed in this thesis.

4.2 Dataset selection

Datasets for the objective evaluation of audio-to-audio alignment must contain at least two recordings of the same underlying music as well as ground-truth alignment annotations among those recordings. The number of publicly accessible datasets that fulfill both criteria is small. Few datasets with event annotations contain two or more recordings of the same underlying piece, and vice versa: of the datasets that contain multiple versions of a piece, many are designed for similarity-based

[1] www.music-ir.org/mirex/wiki/2010:Audio_Cover_Song_Identification
[2] www.music-ir.org/mirex/wiki/Query_by_Singing/Humming

alignment tasks like version identification and/or cover song identification and so do not contain alignment annotations (e.g., Downie and Bay 2008). Some evaluations have overcome the dearth of ground-truth alignment data through creation of ground-truth data from artificially warped duplications of a single original recording (e.g., the first evaluation in Kirchhoff and Lerch 2011) by mapping the events in the original timeline to the known post-warping times of those same events in the generated recordings. Other evaluations have used automated beat-tracking systems to annotate evaluation sets, by mapping successive beats in one recording to successive beats in another recording of the same piece (e.g., Dixon and Widmer 2005). This latter approach relies on successful automated beat segmentation of the recordings to be aligned.

4.2.1 The Chopin dataset

The dataset used in this evaluation, the Chopin dataset (Goebl 2001), is one of the few datasets that contains both multiple recordings of the same piece and an annotated ground-truth alignment mapping. Originally created for a study on melodic emphasis in piano performance, the Chopin dataset has since been used for other audio-to-audio alignment evaluations (Dixon and Widmer 2005; Kirchhoff and Lerch 2011). It contains both audio files and event onset annotations for excerpts from two compositions by Frédéric Chopin (the first 45 measures of Ballade in F major, op. 38, and the first 21 measures of Etude in E major, opus 10 No. 3) performed by 22 different skilled pianists on a Bösendorfer computer-monitored grand piano. The musical scores of these two excerpts are included in Appendix A and the audio is available online.[3]

In the Chopin dataset, onset annotations for each note were automatically generated by the computer-monitored piano during each recording session. The onset time of each note event in a recording is paired manually with its corresponding note in the original symbolic score. The resultant mapping of events in all recordings to events in the symbolic score provides the overall ground-truth alignment between any two or more recordings (of that same excerpt) in the dataset.

For this evaluation, a reduced subset of the Chopin dataset was used, consisting of the first 40 score events (approximately nine measures) of all 22 recordings of the Ballade. These ranged in duration from 20 to 26 seconds, with an average duration of 22.5 seconds. This dataset reduction was implemented to reduce the

[3] www.iwk.mdw.ac.at/goebl/mp3.html

overall experimental runtime; precedent for evaluating audio-to-audio alignment with short excerpts (in some cases, as short as two seconds) was set by Basaran et al. (2011), Müller et al. (2010), and Bohak and Marolt (2012), among others. One of the preliminary experiments in Chapter 5 compares DTW alignment of this reduced dataset to DTW alignment of the original full dataset.

4.3 Experimental setup

Two sets of experiments are conducted, to test performance of the CPM on each pairwise and multiple alignment. DTW is used as a benchmark audio-to-audio alignment algorithm against which to compare alignment by the CPM, since DTW is frequently used in audio-to-audio alignment applications (Chapter 2) and DTW alignment has been objectively evaluated on the Chopin dataset (Dixon and Widmer 2005; Kirchhoff and Lerch 2011).

The first experiment compares pairwise alignment by the CPM to pairwise alignment by DTW and the second experiment compares multiple alignment with the CPM to iterative pairwise alignment by DTW. For the pairwise alignment evaluation, all 231 unique pairings of the 22 performances were aligned. For the multiple alignment evaluation, groups of size three, four, eight, twelve, and sixteen recordings were randomly selected from all possible combinations of the 22 recordings. 231 unique groups of size three and four were aligned; 160 unique groups of size eight, twelve, and sixteen were aligned.

4.3.1 Evaluation metrics

Alignment in this thesis is evaluated with an objective, synchronization-based deviation metric proposed by Dixon and Widmer (2005). This deviation is a measure of distance between an alignment path and the ground-truth alignment path. The deviation measures of an alignment are the collection of distances between each known point along the ground-truth path to their nearest neighbor on the algorithmically generated alignment path.[4] Using this metric, algorithm success (or lack thereof) is determined by comparing the sets of deviations for different alignments, on the basis of statistics like mean, median, and maximum deviation.

To the best of our knowledge, this metric has not previously been applied to multiple audio-to-audio alignment; in this thesis, the only modification to extend

[4]Dixon and Widmer (2005) calls this distance metric "error" instead of "deviation."

it from pairwise to multiple alignment was in the choice of distance algorithm. For pairwise alignment, Dixon and Widmer (2005) and Kirchhoff and Lerch (2011) calculate deviation with the Manhattan distance measure. For calculating the distance between multi-dimensional points, however, Euclidean distance is more often used than Manhattan distance. In order to apply the same deviation calculation both to pairwise and to multi-dimensional alignment evaluation, Euclidean distance was used for both. (One of the preliminary experiments in Chapter 5 compares pairwise DTW alignment for each Manhattan and Euclidean distance measures; every other alignment evaluation in the thesis uses Euclidean distance.)

Whether calculated for two- or for multi-dimensional alignment paths, the resultant deviation values are single-dimensional scalars. To compare alignment success across alignment groups of different sizes in this thesis, an additional deviation-per-recording metric is calculated by dividing each deviation by the number of recordings in the group alignment.

4.3.2 Implementation details

Feature extraction

Each audio file in the Chopin dataset was recorded in stereo at 44.1 kHz. As in other audio-to-audio alignment evaluations (Kirchhoff and Lerch 2011), before feature extraction the audio is first converted to mono, normalized to a maximum amplitude of one, and downsampled from 44.1kHz to 22.05kHz. Pitch chroma features are then extracted with the Chroma Toolbox for MATLAB (Müller and Ewert 2011), using a window size of 2408 samples with a 50% overlap, resulting in twelve-dimensional feature vectors with a time resolution of 46 ms. Finally, the features are normalized with respect to the Euclidean norm.

Ground-truth alignment mapping

The ground-truth annotations of the Chopin dataset are in the form of text files, one file per recording. Each file contains a list of the symbolic notes in the score paired with the timestamp of its onset in the corresponding audio recording.[5] The annotations contain minimal information about performance errors. If a note has been skipped by the performer, its time annotation is listed as "`no_played_note`;" any

[5]The timestamps are provided in units of MIDI ticks, along with the tick-to-second conversion rate.

other performance errors (e.g., accidental additions) are ignored in the annotation file. (The reduced dataset used in this thesis contained no skipped notes.)

As the symbolic score is identical across all 22 recordings of each excerpt, ground-truth alignment paths are calculated by matching the onsets of each note in one annotation file to the corresponding onsets of that same note in the other annotation files. As in Dixon and Widmer (2005), simultaneous notes in the score are grouped together into a single score event to prevent measurement inconsistencies due to the lack of fixed order of chord notes not played at precisely the same time. The timestamp for each score event is the average of the timestamps of its constituent notes. Ornamental notes such as grace notes are not, however, grouped into score events, despite having the same score onset times as the non-ornamental notes they precede, since they have a fixed performance order.[6]

Iterative pairwise alignment with DTW

In general, the evaluation was scripted in a combination of R and MATLAB (R Core Team 2012; MATLAB and MATLAB Signal Processing Toolbox 2011). Implementation of the CPM, in MATLAB, was discussed in Chapter 4; DTW is implemented as follows.

Pairwise DTW was performed with the `dtw` package for R (Giorgino 2009);[7] entries in the distance matrix were calculated as the Euclidean distance between the feature vectors. A standard global path constraint was used, which force-aligned the start and end of each recording to one another. To calculate the warping path, a standard single-step cost path (in horizontal, vertical, and diagonal directions) was implemented. Audio-to-audio alignment has been performed both with and without a penalty applied to diagonal steps along the cost path, a penalty that serves to remove the diagonal bias of the basic single-step cost path (Dixon and Widmer 2005; Kirchhoff and Lerch 2011). In this thesis, both cost path variants were implemented and compared in a preliminary experiment in Chapter 5.

Iterative alignment using the DTW-aligned pairs was scripted in MATLAB. To perform iterative pairwise alignment, one of the recordings from the group was chosen as the reference. Each remaining (secondary) recording was then mapped successively to that reference using the pairwise DTW alignment mapping between the two. When multiple frames in a secondary recording map to a single frame in the reference recording, extra frames were inserted into the reference until there existed

[6] No mention of ornamental notes is made in Dixon and Widmer (2005) or Kirchhoff and Lerch (2011), so treatment of grace notes in those works is ambiguous.

[7] `dtw.r-forge.r-project.org/`

one reference frame per corresponding secondary recording. The inserted frames were copies of the preceding frame, for all secondary recordings already mapped to the reference. (Single frames in the secondary recording that map to multiple frames in the reference recording required no reference timeline insertion or alteration; they were simply repeated as indicated by their pairwise alignment mapping.)

The choice of reference recording in iterative pairwise alignment with DTW was found to be non-trivial. As shown in Figure 4.1 and quantified in the next chapter, the alignment success of any given group of recordings varies greatly depending on the choice of reference recording. To more fully explore this variation in the thesis evaluation, each group alignment was repeated as many times as recordings in the group, with each recording used once as the reference recording (e.g., for a group alignment of four recordings, four different global alignment paths were calculated). This means that for each alignment group two sets of deviation measures were reported: one set containing all possible alignments of the group, combined (i.e., all deviations reported for each group alignment, as generated using each of the different possible reference recordings), and one set consisting of just the deviations from the best alignment of the group. In this latter case, the best alignment is considered to be the alignment that generates the lowest total deviation, where total deviation is calculated by summing all pointwise deviations from one individual alignment.

From latent time to a more meaningful timeline

To compare CPM to DTW using the metrics introduced above, the alignments must exist in the same alignment space. The warping path generated by DTW (for both pairwise and iterative pairwise alignment) exists in an alignment space defined by the timelines of the original recordings (one per axis). By definition of the warping path constraints used, it has a consistent sample rate and therefore path resolution of no more or less than one frame per sample.

The warping path generated by the CPM is output in the latent time space, however, and when used to map back to the same alignment space utilized in DTW has a non-consistent sampling rate and resolution, due to a lack of meaningful or consistent sampling rate in the calculated latent trace. To transform this mapping to the multi-dimensional alignment space of DTW, rather than one indexed by the inconsistently sampled latent trace, the trace is used as a dictionary to map back into the alignment space defined by the original timelines. To ensure the same warping-

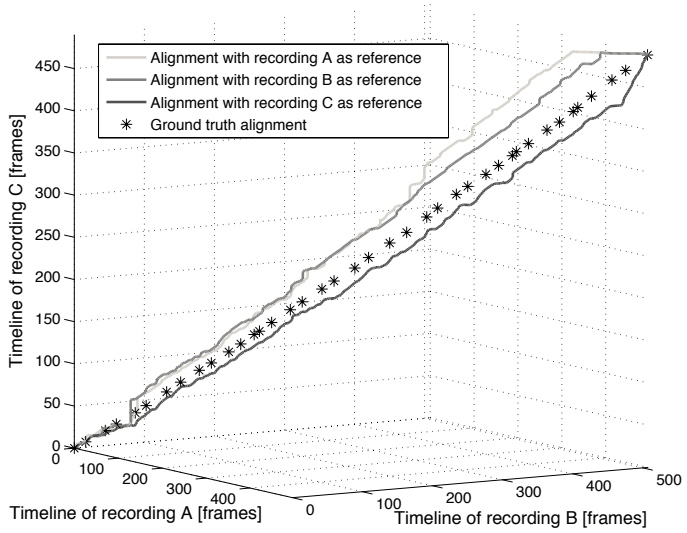

Fig. 4.1 The alignment paths through three different performances of Chopin's Ballade, as used in the evaluation. Each alignment path is calculated through the same three recordings, each using a different recording as the reference.

path resolution as that generated by DTW, the CPM's warping path is interpolated to fill in any gaps in the path larger than those of the DTW.

Results from the evaluation described in this chapter are presented in the next chapter.

5—Results and discussion

In this chapter, experimental results are presented and discussed, after a brief review of statistical tests in Section 5.1. In Section 5.2, preliminary evaluations compare the implementation of DTW for this thesis to previous results from the literature, and the choice of reduced dataset is investigated; in Section 5.3, pairwise alignment with the CPM is compared to pairwise alignment with DTW; and in Section 5.4, simultaneous multiple alignment with the CPM is compared to iterative pairwise alignment with DTW, on variously sized groups of recordings.

5.1 Review of statistical tests

This section briefly reviews the statistical tests used in this thesis. Section 5.1.1 introduces the tests used to investigate data distributions (e.g., the distribution of alignment deviations by a particular algorithm), which determine whether parametric or nonparametric analysis methods are required when comparing data from different experimental groups. Section 5.1.2 introduces the tests used to compare two or more experimental groups (e.g., alignment under different experimental conditions, such as different algorithms), which determine whether or not the data were drawn from the same population (i.e., whether the different algorithms yield different results). All tests performed in this thesis use a confidence interval of 95% (for significance level $\alpha = 0.05$). Note that the sample size, n, of an experimental group is the total number of deviations calculated for all alignments performed in the experiment. For example, data from six alignments of recordings containing eight score events have sample size $n = 6 \times 8 = 48$.

5.1.1 Investigating normality and variance

The classes of test used to compare experimental groups (parametric versus nonparametric) are based on assumptions about the distribution of data in those groups.

To determine which type of comparison tests to use, the data from each experiment are tested for distribution normality and homogeneity of variance. If each group has a normal distribution and there is homogeneity of variance across the groups, parametric tests are used. Otherwise, nonparametric tests must be used.

To determine if data are normally distributed, three tests are used to accept or reject the null hypothesis of the data belonging to a normally distributed population: the Kolmogorov-Smirnov (K-S) test, which compares the data to a normal distribution density;[1] the Lilliefors test (Lilliefors 1967), which is based on the KS test, but with a stochastically smaller probability distribution (i.e., greater sensitivity than the KS test);[2] and the Jarque-Bera (J-B) test (Jarque and Bera 1987), which tests the skewness (lack of symmetry) and kurtosis (flatness of the peak) of the data distribution as compared to those of a normal distribution.[3] To report the tests for normality, each test's statistic (D) is reported, along with the sample size (n) and significance (p), e.g., $D_{KS}(9240) = 0.5$, $p = 0.001$; $D_L = 0.34$, $p = 0.001$; $D_{JB} = 9.3 * 10^5$, $p = 0.001$ for a test of normality on a population of 9240 samples.

Variance is a measure of the spread of values in a dataset, calculated as the data's standard deviation squared. Homogeneity of variance is when experimental groups have a similar variance. Two tests are commonly used to determine if experimental groups have homogeneity of variance: Levene's test and the variance ratio. For data with large sample sizes the variance ratio is a more reliable measure (Field 2005, 98); this is because small differences in variance, such as the presence of extreme outliers, can unduly influence the significance of Levene's test when sample sizes are large. Since all tests in this thesis have large sample sizes ($n \geq 231$), the variance ratio is used to test for homogeneity of variance. To calculate the variance ratio, variance is first calculated for each group individually; the variance ratio is then the largest divided by the smallest variance of all groups. Homogeneity of variance is assumed when the variance ratio is less than two.

Folded distributions

As will be seen in the results, nonparametric tests were required in all experiments due to non-normality of the data. This non-normality was an anticipated consequence of the deviation metric. Deviation is the distance between a ground-truth point and an algorithmically generated warping path, and contains no information

[1] MATLAB function `kstest` (Statistics Toolbox).
[2] MATLAB function `lillietest` (Statistics Toolbox).
[3] MATLAB function `jbtest` (Statistics Toolbox).

about whether the point is above or below the line. The distribution of deviation data therefore has a lower bound at zero:

> Measurements are frequently recorded without their algebraic sign. As a consequence, the underlying distribution of measurements is replaced by a distribution of absolute measurements. ... The effect of dropping the sign is to add the otherwise negative values to the positive values. Geometrically this amounts to folding the negative side of the distribution onto the positive side. (Leone et al. 1961)

Non-normal distributions, requiring nonparametric comparison tests, are therefore expected for all experiments.

5.1.2 Comparing data groups

Depending on the number of experimental groups in each experiment, as well as the relative sample size of each group, we perform one of two nonparametric comparisons: the Wilcoxon signed-rank test or Kruskal-Wallis one-way analysis of variance by ranks. For two experimental groups with repeated measurements, comparison is performed with the Wilcoxon signed-rank test (Wilcoxon 1945),[4]. Repeated measurements are measurements performed on the same "subject;" in this thesis, a subject is an individual score event (e.g., the third note in the score). By definition, therefore, this test is only ever performed on two experimental groups with the same sample size (e.g., results of two different alignment algorithms used to align exactly the same score events). The Wilcoxon signed-rank test is reported as the test statistic T, significance, and effect size r, e.g., $T = 0$, $p = 0.01$, $r = -.57$. When significant, it indicates that the two groups have different medians.

To compare more than two experimental groups, or two groups with different sample sizes, we use the Kruskal-Wallis one-way analysis of variance by ranks method (Kruskal and Wallis 1952), a nonparametric version of the (parametric) one-way analysis of variance (ANOVA) test.[5] The Kruskal-Wallis test is reported as the test statistic H, along with its degrees of freedom (the number of groups minus one) and significance, e.g., $H(2) = 8.88$, $p = .07$ for a test with two degrees of freedom. When significant, it indicates that at least one of the groups has a different distribution from the rest.

[4] MATLAB function `signrank` (Statistics Toolbox).
[5] MATLAB function `kruskalwallis` (Statistics Toolbox).

The mean, maximum, and median for each experimental group will be reported in the tables of results. For both the Wilcoxon and Krsuskal-Wallis tests, significance indicates only that the groups have different distributions from one another. Since these tests rely on comparing the medians of experimental groups, and the groups are non-normal, median is more meaningful of the three statistics reported. Mean and maximum are reported here because they are used to compare alignments in DW05 and KL11.

5.2 Preliminary DTW investigation

The first set of evaluations investigates general choices regarding DTW implementation and evaluation methodology: First, results from the DTW implementation used in this thesis are compared to results previously published in the literature (Section 5.2.1). Second, two different choices of alignment path cost-path weighting for DTW are compared (Section 5.2.2). Next, the reduced Chopin dataset used in the remainder of experiments is compared to the original Chopin dataset (Section 5.2.3). Finally, two different distance measures for calculating alignment deviation, Euclidean and Manhattan, are compared (Section 5.2.4).

5.2.1 Comparison with the literature

This first evaluation compares the implementation of DTW in this thesis to similar implementations of DTW in the literature, to ensure comparable alignment outcomes despite slight differences in feature selection and feature frame size. Table 5.1 displays the results of DTW, as implemented with each of the two DTW cost-path weightings (one with a diagonal bias [♢] and one without [♣], as discussed in Section 4.3.2) on each the original Chopin dataset and the reduced Chopin dataset. The results of the two previous DTW alignment evaluations performed using the Chopin dataset, Dixon and Widmer (2005) (DW05) and Kirchhoff and Lerch (2011) (KL11), are also included in this table.[6]

As in KL11 and DW05, a Manhattan distance measure is used to calculate alignment deviation for all evaluations. Because KL11 and DW05 each implemented

[6] KL11 provided separate results for the Etude and the Ballade; in the table they have been combined into a single deviation measure to enable comparison of results across the different studies. The Ballade contains nearly half as many score events as the Etude, so the individual excerpt means were weighted when combined: $mean_{\text{Full}} = .33{*}mean_{\text{Ballade}} + .67{*}mean_{\text{Etude}}$.

different cost-path weightings (diagonally biased and diagonally unbiased, respectively) both were re-implemented for this thesis. It should be noted that while the window size (1024 samples) is the same for each of the six treatments, the features extracted in this thesis were calculated from downsampled audio, such that each feature frame spans twice the duration of the features in KL11 and DW05: 46 ms as compared to 23 ms. Additionally, the features themselves are not identical. DW05 used pitch onsets and KL11 combined chroma features and onset features, while this thesis makes use of a basic chroma feature.

Table 5.1 Deviation of DTW alignment as compared to the literature

	Literature$_{[frame = 23\ ms]}$		*Thesis*$_{[frame = 46\ ms]}$			
Dataset:	FULL$^{\clubsuit}_{DW05}$	FULL$^{\diamond}_{KL11}$	FULL$^{\clubsuit}$	FULL$^{\diamond}$	REDUCED$^{\clubsuit}$	REDUCED$^{\diamond}$
Mean [frames]	1	1.1	1.32 ± 0.04	1.10 ± 0.02	1.15 ± 0.07	0.79 ± 0.02
Mean [ms]	23	26.0	61.1 ± 1.8	51.0 ± 1.1	53.4 ± 3.3	36.8 ± 1.1
Median [frames]	1	1	1	1	0	1
Median [ms]	23	23	46	46	0	46
Max [frames]	NA	152	44	44	29	25
Max [s]	NA	3.66	2.04	2.04	1.35	1.16

DW05 is Dixon and Widmer (2005) and KL11 is Kirchhoff and Lerch (2011); neither reported uncertainty. Uncertainty for the thesis deviation is the standard deviation of the mean. ♢ indicates a cost-path weighting favoring diagonally biased DTW; ♣ indicates an unbiased weighting. Manhattan distance was used to calculate deviation for all treatments.

Discussion

As seen in the table, the DTW mean and median alignment deviations calculated here are comparable to those published in the literature for units of frames. (Given the differences in audio sampling rate, the deviations span half as much time in the literature as they do in this thesis.)

The biggest differences across alignments can be seen in the measure of maximum deviation: the results published in the literature are considerably larger than found here (152 versus 44 frames, and 3.7 versus 2.0 seconds). While this discrepancy could be due to the different features used in the alignment, it is likely due to choice of score event: as discussed in the previous chapter, simultaneous notes are treated as a single score event, even if the onset of each note in the score event is played

independently from one another. In the MIDI score for the Chopin dataset, grace notes have the same official onset time as the note they precede. In this evaluation, grace notes were treated as independent score events, as their timing is known to precede that of their following notes. Depending on how DW05 and KL11 calculated their score events, however, the grace notes might have been included as part of the following score event, introducing systematic error into the calculation of alignment deviation (but not actually affecting the alignment itself—just the evaluation of the alignment).

5.2.2 Choice of DTW cost-path weighting

DW05 and KL11 used different DTW cost-path weightings: DW05 used an unbiased weighting, while KL11 used a diagonally biased weighting, which preferences diagonal steps in the warping path. To investigate the effect of cost-path weighting on DTW alignment of this dataset, both weightings were implemented in DTW, on both the original and the reduced datasets.

Results

All four test conditions (full dataset with diagonal weighting, full dataset with no weighting, reduced dataset with diagonal weighting, and reduced dataset with no weighting) yielded non-normal distribution of deviations ($p < 0.05$), as seen in Figure 5.1. In addition, the reduced dataset violates homogeneity of variance (variance ratio = 3.13); the full dataset has homogeneity of variance (variance ratio = 1.64). Since normality and homogeneity of variance are violated, and more than two experimental groups of varying sample sizes were compared, the Wilcoxon signed-rank test was used.

A significant difference due to slope weighting was found for both the full dataset ($T = 6.6{*}10^7$, $p < 0.01$, $r = -0.011$) and the reduced dataset ($T = 1.6{*}10^6$, $p < 0.01$, $r = -0.043$). For the full dataset, the diagonally biased weighting had a greater mean and equal median ($mean = 1.32 \pm 3$ frames, $median = 1$ frame) compared to the non-diagonally biased weighting ($mean = 1.10 \pm 2.3$ frames, $median = 1$ frame). For the reduced dataset, the diagonally biased weighting yielded a greater mean but lesser median ($mean = 1.15 \pm 2.6$ frames, $median = 0$ frames) than the non-diagonally biased weighting ($mean = 0.79 \pm 1.5$ frames, $median = 1$ frame). (Both mean and median are reported because mean was used to compare results in the literature, while median is a more reliable statistic for comparing these non-normally distributed data.)

Discussion

The median alignment deviations for diagonally biased weighting are comparable to those with a non-diagonally biased weighting, Because the diagonally biased slope weighting slightly outperforms the unbiased weighting in the reduced dataset (it has a slightly lower median), we chose to use it as the slope weighting for all subsequent DTW alignments.

This result also highlights the importance of parameter choice when implementing an algorithm such as DTW: it is possible that the out-performance of DW05 by KL11 was due to the difference in slope weighting implemented by each, rather than to the careful feature selection credited with the improved alignment. Their results are especially ambiguous since mean and maximum values were used to compare the studies. As can be seen here, using mean rather than median to compare the weightings flips the outcome, indicating that a non-diagonally biased weighting yields better alignment performance.

5.2.3 Reduced versus original dataset

To investigate the effect of the reduced dataset, DTW alignment of the reduced dataset was compared to DTW of the original dataset, for both biased and the unbiased DTW slope weightings.

Results

Due to the non-normality of the samples in all four test conditions (determined in the previous section), the violation of homogeneity of variance in the alignments with unbiased slope weighting (variance ratio = 2.52),[7] and the comparison of more than two experimental groups, the Kruskal-Wallis one-way analysis of variance by ranks test was used.

Both the biased and unbiased slope weightings showed a significantly larger mean deviation for the full compared to the reduced dataset ($H(1) = 68, p < 0.01$and $H(1) = 2 * 10^2, p < 0.01$, respectively). For the diagonally biased cost path, the full dataset has a larger mean deviation ($mean = 1.32 \pm 3$ frames, $median = 1$ frame) than the reduced dataset ($mean = 1.15 \pm 2.6$ frames, $median = 0$ frames). For the unbiased cost path, the full dataset has a larger mean deviation ($mean =$

[7] The alignments with biased slope weighting maintained homogeneity of variance (variance ratio = 1.32).

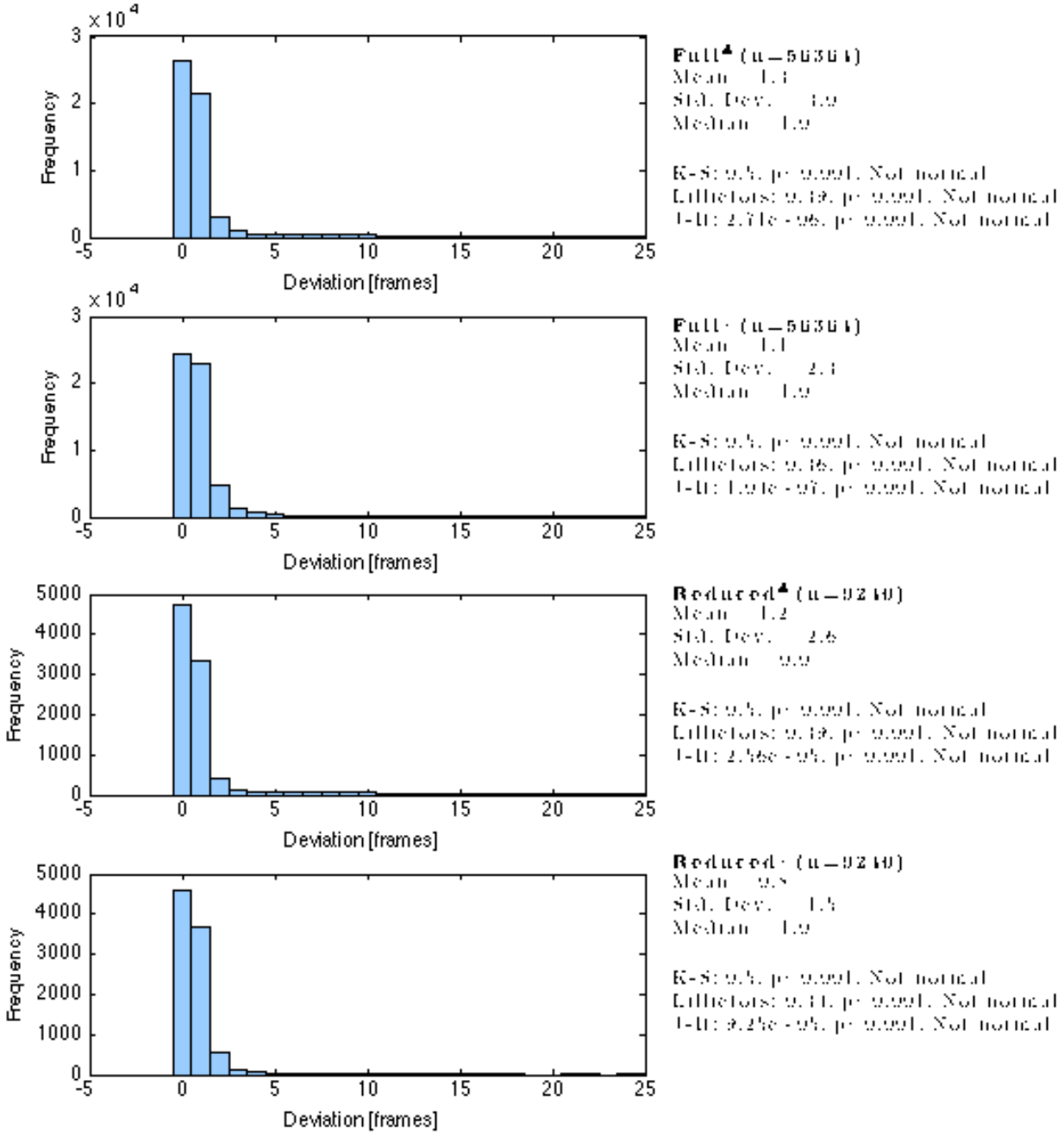

Fig. 5.1 Histogram of alignment deviations for pairwise alignment by DTW on the original and reduced datasets, with both diagonally biased and diagonally unbiased cost-path weightings.

1.10 ± 2.3 frames, *median* = 1 frame) than the reduced dataset (*mean* = 0.79 ± 1.5 frames, *median* = 1 frame).

Discussion

The difference in alignment performance (median) between the full and reduced datasets are less than or equal to one frame. As the resolution of the alignment algorithms is on the order of a single frame, by definition of DTW, the difference in performance between the full and reduced dataset falls below the resolution threshold and can be discounted. For the remainder of evaluations in this thesis, the reduced dataset is used instead of the full dataset and the results can be compared to those of other studies that perform alignment of the full Chopin Ballade excerpt.

5.2.4 Deviation distance measure

Finally, the choice of distance measure for calculation of the deviation metric, Euclidean versus Manhattan, is investigated.

Results

The results of alignment performed by each of the distance measures are non-normally distributed, as seen in Figure 5.2. Thus, despite homogeneity of variance (variance ratio = 1.44) a nonparametric comparison test was required. Since the two sets of results are repeated measures, the Wilcoxon signed-rank test was used. A significant difference ($T = 0$, $p < 0.01$, $r = -0.189$) was found between the Manhattan (*mean* = 0.79 ± 1.5 frames, *median* = 1 frame) and Euclidean (*mean* = 0.71 ± 1.2 frames, *median* = 1 frame) distance measures.

Discussion

Despite the Manhattan and Euclidean distance measures yielding significantly different deviation distributions, they have the same median. This renders the choice of distance measure in the deviation calculation metric trivial, as long as the same measure is used for all alignments being compared.

Manhattan distance was used for the preceding alignments in this section as it was used in the literature on pairwise alignment. In the remainder of this thesis, however, Euclidean distance will be used, as Euclidean distance is a common distance measure in multi-dimensional space, which makes it a more natural choice

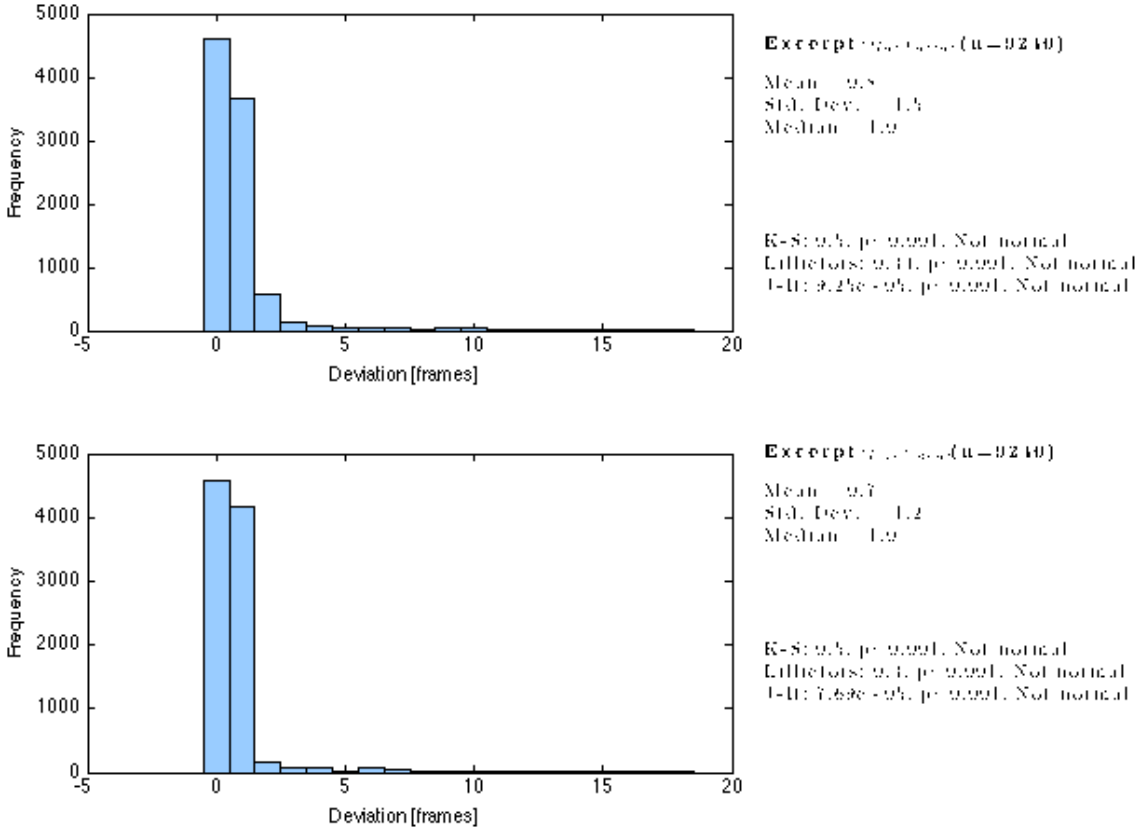

Fig. 5.2 Histogram of alignment deviations for pairwise alignment by DTW, as evaluated with Manhattan and Euclidean distances.

for the multi-dimensional evaluation metric. To maintain consistency throughout the evaluation, it is similarly used for the pairwise evaluation metric.

5.3 Pairwise alignment

This evaluation compares pairwise alignment by the CPM to pairwise alignment by DTW. It should be noted that for this and all following experiments the reduced dataset was used, along with the Euclidean distance metric for deviation calculation and the diagonally biased DTW cost-path weighting.

5.3.1 Results

The results of pairwise alignment by each DTW and the CPM are shown in Table 5.2, Figure 5.3, and Figure 5.4. While the figures present the data more comprehensively, the table mirrors the format of the results presented in DW05 and KL11. In the top portion of the table, cumulative deviation is displayed (cumulative deviation is the frequency distribution of deviations, presented as a percentage of deviations shorter than or equal to the time listed in the leftmost columns). In the bottom portion of the table, the mean, median, and maximum deviation of each set of alignments is listed, both in frames and milliseconds.

Due to the non-normality of the samples in these two test conditions (calculated and displayed in Figure 5.4) and their repeated measures, the nonparametric Wilcoxon signed-rank test was used to compare them. A significant difference was found between DTW and CPM for pairwise alignment ($T = 1.5 * 10^7$, $p < 0.01$, $r = -0.183$).

Table 5.2 Pairwise alignment: DTW vs. CPM

DEVIATION ≤		CUMULATIVE DEVIATION (%)	
Frames	*Seconds*	DTW	CPM
0	0	49.7	1.3
1	0.046	89.4	68
2	0.093	95.4	78.1
3	0.139	96.8	79.1
5	0.232	97.9	80.7
10	0.464	99.7	84.2
25	1.161	100	90.7
50	2.322	100	99
Mean [frames]		0.79±0.02	5.48±1.42
Mean [ms]		36.79±1.06	254.51±66.12
Median [frames]		1.0	0.5
Median [ms]		46.0	23.0
Max [frames]		25	89.15
Max [s]		1.16	4.14

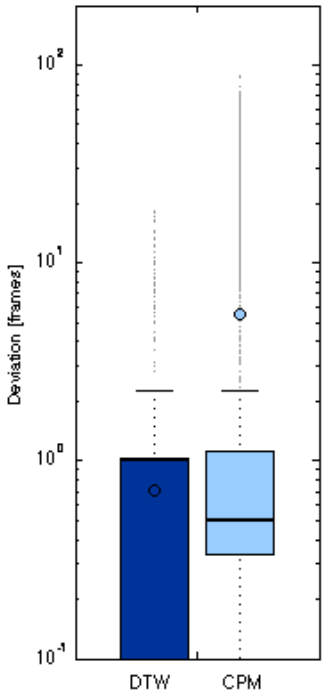

Fig. 5.3 A comparison of pairwise alignment by DTW and CPM. Each box spans the interquartile range (the 25^{th} to 75^{th} percentile) of the deviation distribution. The whiskers form the boundaries between extreme deviation values and outliers. The dark horizontal line marks the median, the circle marks the mean. Due to the logarithmic scaling of the y–axis, deviation values of zero are not able to be displayed for the lower whiskers and the lower bound of the DTW box.

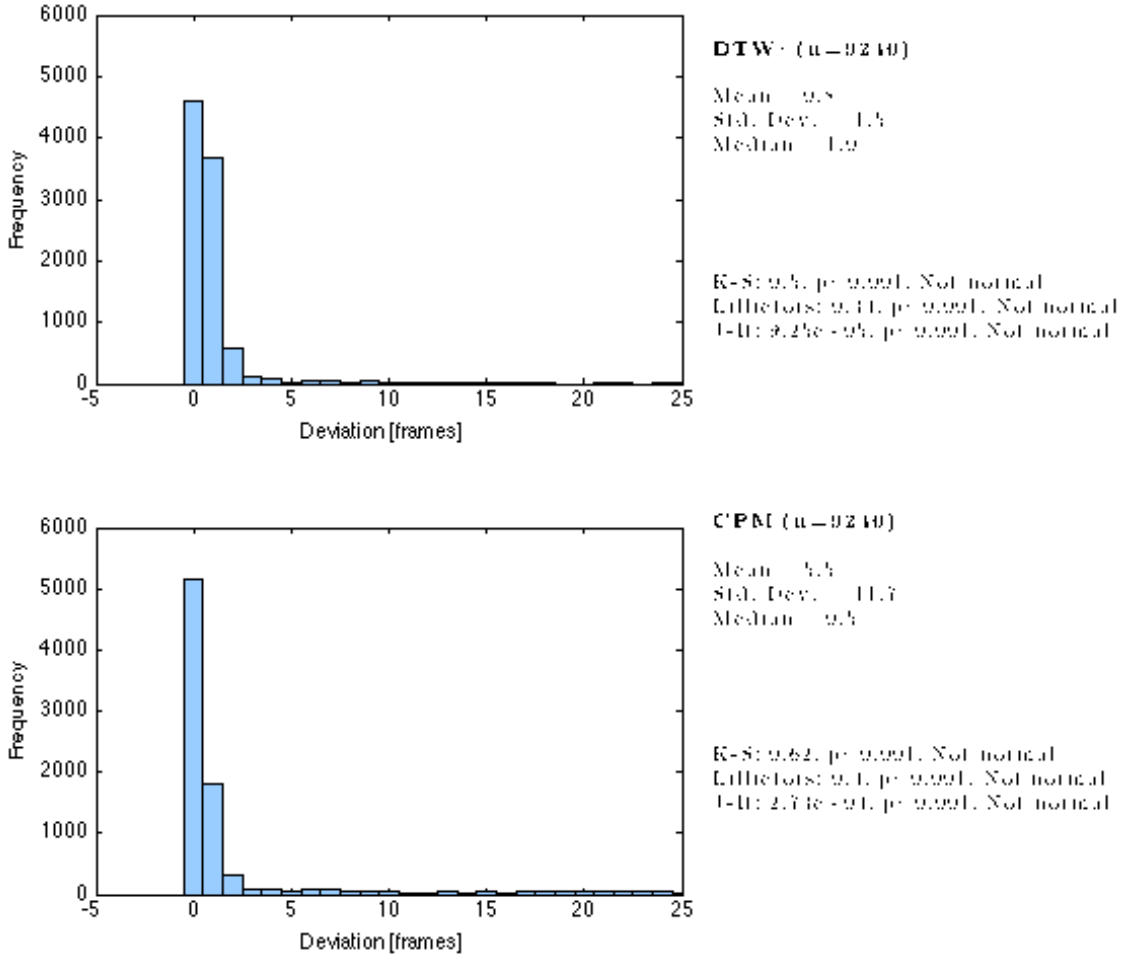

Fig. 5.4 Histogram of alignment deviations for pairwise alignment by DTW and CPM. The x–axes do not span the whole range of deviations, but the full range (including outliers) can be seen in Figure 5.3.

5.3.2 Discussion

It is clear from Figures 5.3 and 5.4 that the CPM is as successful as DTW for pairwise audio-to-audio alignment of music, at least for these particular recordings and audio features. The deviations for each algorithm have the same order of magnitude, with the upper bound on their interquartile ranges (two frames). Additionally, the CPM has a lower median deviation.

The especially large deviations can most likely be attributed to poor alignment of entire pairs of recordings. For the evaluation methodology used in this thesis, each alignment contributes a set of deviation values to the overall deviation data. By definition of the algorithmic constrains of global alignment, the first and last score events must have a low deviation from ground truth because boundary restrictions force the beginning and ending of each recording to map to the beginnings and endings of the other recordings.[8] Additionally, because global alignment is monotonic (the order of events is preserved such that an alignment always progresses forward in time through all recordings), the deviation of a score event is related to the deviation of preceding and following score events.

As it is impossible for an alignment to suddenly stray and then snap back (or vice versa) without violating the monotonicity imposed by both DTW and CPM, large score event deviations do not occur in isolation: large deviations in the midst of smaller ones (or vice versa) are impossible unless the duration between score events is large compared to the feature resolution (sample rate), such that the score events do not adequately represent the alignment path.[9] As the score events in this thesis adequately represent the alignment path, the presence of larger CPM deviations suggests that alignment of several of the pairs of recordings strayed substantially from the ground truth.

In general, these results underscore the importance of reporting alignment results other than the mean and maximum deviation values. Since the data are non-normally distributed the median is a more meaningful statistic than mean, and is the value tested by the nonparametric Wilcoxon signed-rank test.[10] In this particular evaluation, since DTW has a lower mean than CPM and the maximum CPM deviation is nearly four times larger than that of DTW, reporting only the mean and

[8] In the Chopin dataset the first score event is the first frame of each recording, so the first deviation of each alignment by DTW will always be zero.

[9] This underrepresentation of the path is similar to signal aliasing.

[10] Note that when the Wilcoxon test finds a significant difference between experimental groups, no conclusion can be drawn other than that the groups have different distributions.

maximum deviations would logically imply that CPM performs pairwise alignment considerably worse than DTW. By looking at Figures 5.3 and 5.4, however, it is clear that the deviations of each algorithm are on the same order of magnitude, and the median CPM deviation is actually lower than the median DTW deviation. The reason for this mean versus median switch is made clear by the box plot: there are a much greater number of large CPM deviations than for DTW, and these large deviations inflate the mean.

As a side note, the distribution of CPM deviations smaller than one but larger than zero is noteworthy, and likely due to creation of and alignment with the latent trace. Unlike DTW, in which the initial sample rate of the recordings are maintained during alignment, the sample rate of the latent trace has no meaningful relation to those of the input recordings. Because interpolation is involved when the recordings are pairwise aligned back to the latent trace, deviations smaller than a single frame are possible and likely. When the CPM deviations are rounded to the nearest frame when grouping them, their distribution closer to zero more clearly matches that of DTW. This is apparent in the histogram, where the bin size of one effects rounding each deviation to the nearest frame.

5.4 Multiple alignment

This evaluation investigates the two multiple alignment approaches—-iterative pairwise alignment (using DTW) and simultaneous alignment (by the CPM). As mentioned in the previous chapter and illustrated in Figure 4.1, the choice of reference recording affects the overall group alignment. As seen in that figure, the resultant alignment path differs depending on which recording is used as the reference. For this reason, two different DTW alignment results are reported for each group alignment: DTW_{All} and DTW_{Best}. Each group of recordings was aligned by iterative DTW as many times as the number of recordings it contained, with each recording used once as the reference recording, and deviation measurements were calculated for each of these alignments. DTW_{All} consisted of all deviation measurements for all reference recordings of the group (i.e., all possible choices), while DTW_{Best} contains only the deviation measurements from the single most successful alignment of the group (i.e., alignment with the reference recording found to perform best for that group).

For example, alignment of recordings A, B, and C by iterative DTW is performed three times: once with A as the reference recording, once with B as the reference recording, and once with C as the reference recording. If the smallest deviations are found to occur when using B as the reference, DTW_{Best} is DTW_B, while DTW_{All} contains all deviation measurements from DTW_A, DTW_B, and DTW_C combined. This means that while CPM and DTW_{Best} each have sample sizes n = [number of unique alignment groups] × [number of score events in one alignment], DTW_{All} has sample size n = [number of unique alignment groups] × [number of score events in one alignment] × [number of recordings aligned in each group].

Multiple alignment was calculated for alignment of groups of three, four, eight, twelve, and sixteen recordings. 231 unique groups of recordings were aligned for groups of size three and four; 160 unique groups of recordings were aligned for groups of size eight, twelve, and sixteen.

5.4.1 Results

Normality testing for each of these group sizes are shown in Figures 5.5 to 5.9, respectively. As deviation distributions for each of these group sizes was non-normal, all comparison testing was nonparametric. The Kruskal-Wallis one-way analysis of variance by ranks method was used to test for significance between DTW_{All} and DTW_{Best} and between DTW_{All} and CPM, since each pair of groups has two different sample sizes. The Wilcoxon signed-rank test was used to test for significance between DTW_{Best} and CPM, as they have the same sample size and consist of repeated measurements. For all multiple alignments, the CPM performed significantly better ($p<0.05$) than both iterative DTW alignment with each recording used as a reference once (DTW_{All}) and iterative DTW alignment using the best-performing reference (DTW_{Best}). Results from these comparisons are shown in Table 5.3.

Additionally, the performance of each of the three treatments (DTW_{All}, DTW_{Best}, CPM) for each of the different group sizes is plotted in Figure 5.10. Because it is difficult to see the values of CPM from this figure, these same CPM alignments are replotted in Figure 5.11, on a logarithmic scale. A significant difference was found across all tested group sizes for each of these three treatments: for DTW_{All}, $H(5) = 6.9 * 10^4$, $p < 0.01$; for DTW_{Best}, $H(5) = 2.4 * 10^4$, $p < 0.01$); and for CPM, $H(5) = 1.7 * 10^4$, $p < 0.01$.

Table 5.3 Multiple alignment: Significance testing on different group sizes

Size	DTW$_{All}$ vs. DTW$_{best}$	DTW$_{All}$ vs. CPM	DTW$_{Best}$ vs. CPM
Three	$H(1) = 1.2 * 10^3$, $p < 0.01$	$H(1) = 1.5 * 10^4$, $p < 0.01$	$T = 1.7 * 10^6$, $p < 0.01$, $r = -0.56$
Four	$H(1) = 1.3 * 10^3$, $p < 0.01$	$H(1) = 1.8 * 10^4$, $p < 0.01$	$T = 1.4 * 10^6$, $p < 0.01$, $r = -0.57$
Eight	$H(1) = 1.3 * 10^3$, $p < 0.01$	$H(1) = 1.4 * 10^4$, $p < 0.01$	$T = 3.9 * 10^5$, $p < 0.01$, $r = -0.60$
Twelve	$H(1) = 1.6 * 10^3$, $p < 0.01$	$H(1) = 1.5 * 10^4$, $p < 0.01$	$T = 2.7 * 10^5$, $p < 0.01$, $r = -0.60$
Sixteen	$H(1) = 2.0 * 10^3$, $p < 0.01$	$H(1) = 1.6 * 10^4$, $p < 0.01$	$T = 1.0 * 10^5$, $p < 0.01$, $r = -0.61$

Note: Significance testing between DTW$_{All}$ and DTW$_{Best}$ and between DTW$_{All}$ and CPM was performed with the Kruskal-Wallis one-way analysis of variance by ranks test; significance testing between DTW$_{Best}$ and CPM was performed with the Wilcoxon signed-rank test.

5.4.2 Discussion

The results very clearly demonstrate the success of CPM for alignment of this dataset—especially as compared to iterative pairwise alignment with DTW. Even as compared to the best-case DTW scenario (DTW$_{Best}$), the CPM performed either better-than or comparably to DTW for every alignment group size, as judged by both mean and consistency (size of interquartile range).

Additionally, the CPM was considerably more reliable than DTW, regardless of the number of recordings in a group. As seen in Figure 5.11, the size of CPM deviations increased only slightly as the number of recordings to align increased. In contrast, the size of DTW errors steadily increases with the increase in alignment group size. That said, the contribution of each recording to the overall DTW alignment decreases as more recordings are used in an alignment, as can be seen in the bottom plot of Figure 5.10.

As can be seen in all the histograms, the deviation distributions for the two algorithms have distinctive shapes that are maintained regardless of the number of recordings in an alignment group. All distributions have peaks near zero, while DTWdistributions (both DTW$_{Best}$ and DTW$_{All}$ variants) have an additional bell-shaped curve. The presence of a peak near zero for all evaluations is due both to good alignments and, for DTW, to the zero-valued deviation of the first score event of all alignments.[11] Further analysis is required to tease apart these two sources, but is beyond the scope of this thesis.

[11] The first score event occurs in the first feature frame in the Chopin dataset; as global DTW maps the first frames to one another, deviation of the first score event is always zero.

The additional curve in the deviation distributions can likely be attributed to poor alignment of individual groups of recordings. As introduced in the previous section in the context of large deviation values, extreme deviations do not exist in isolation; the presence of the second peak suggests a sizable number of recordings with less than ideal alignment. This intuition is supported by the fact that the shape of the DTW_{All} deviation distribution mirrors that of DTW_{Best}. For all group sizes, the lower (left-hand) slope of DTW_{All} matches DTW_{Best} but then extends notably higher, as it contains a greater proportion of poorer alignments (alignments with greater deviations).

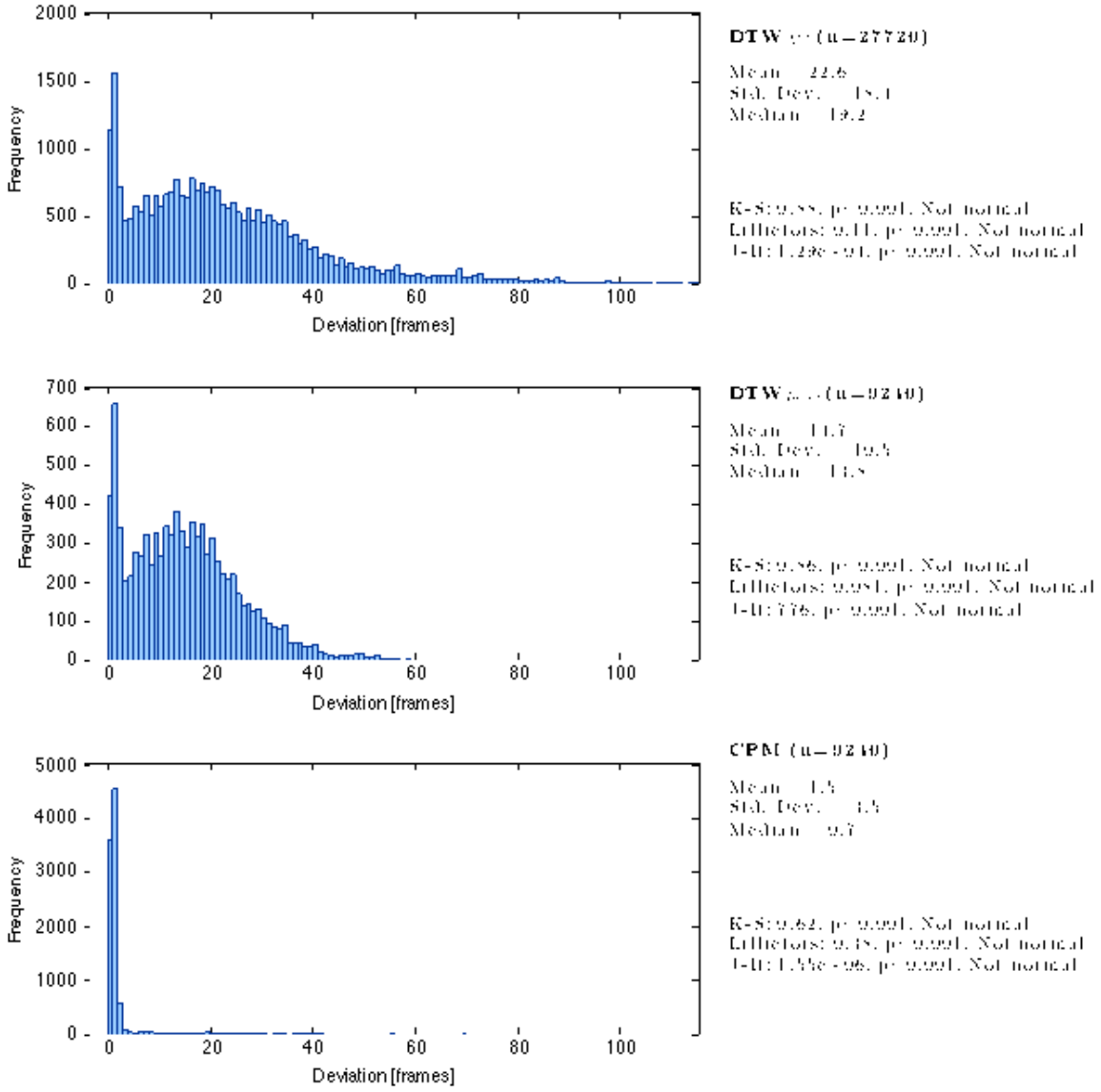

Fig. 5.5 Histogram of alignment deviations for alignments of three recordings.

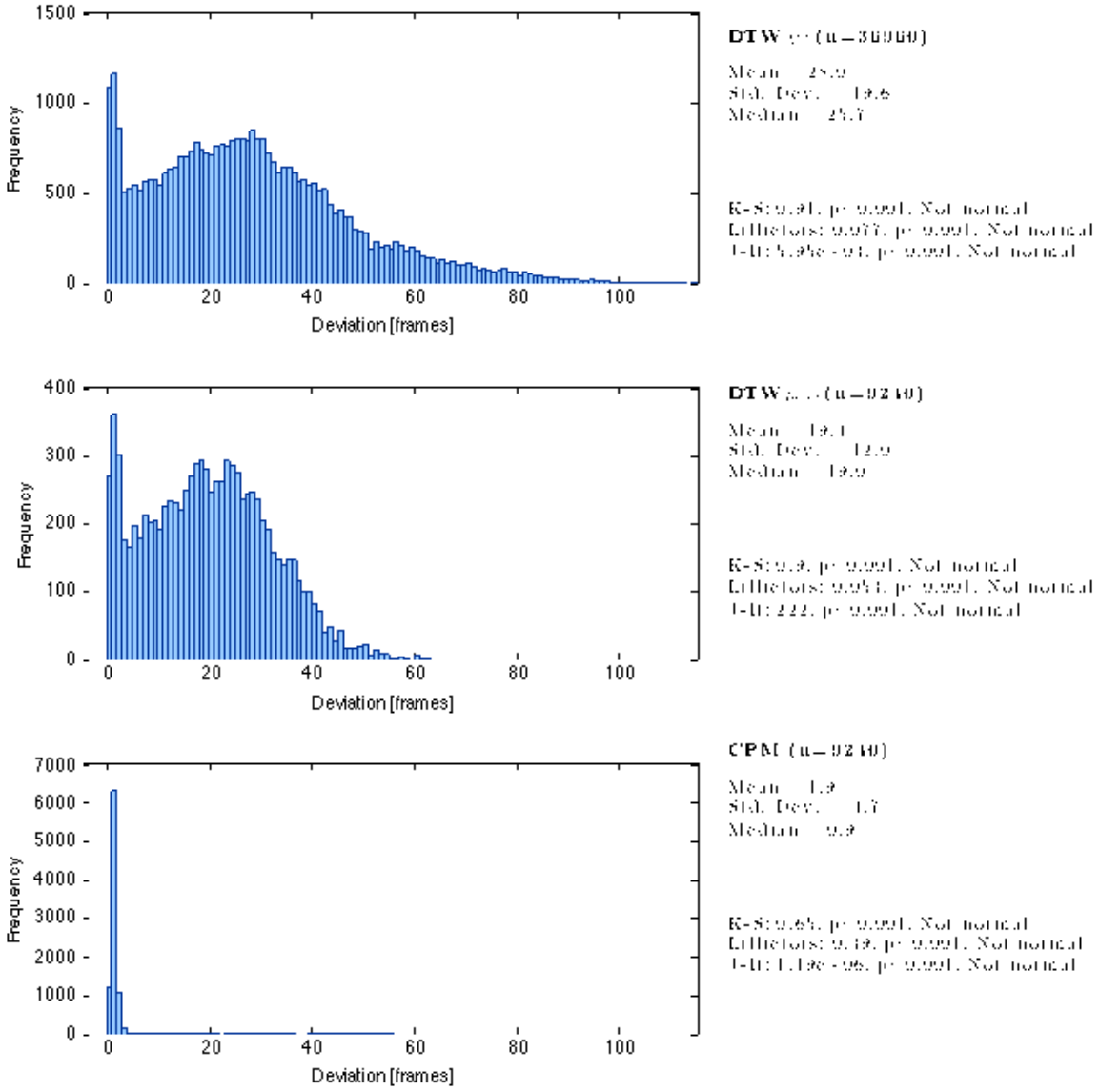

Fig. 5.6 Histogram of alignment deviations for alignments of four recordings.

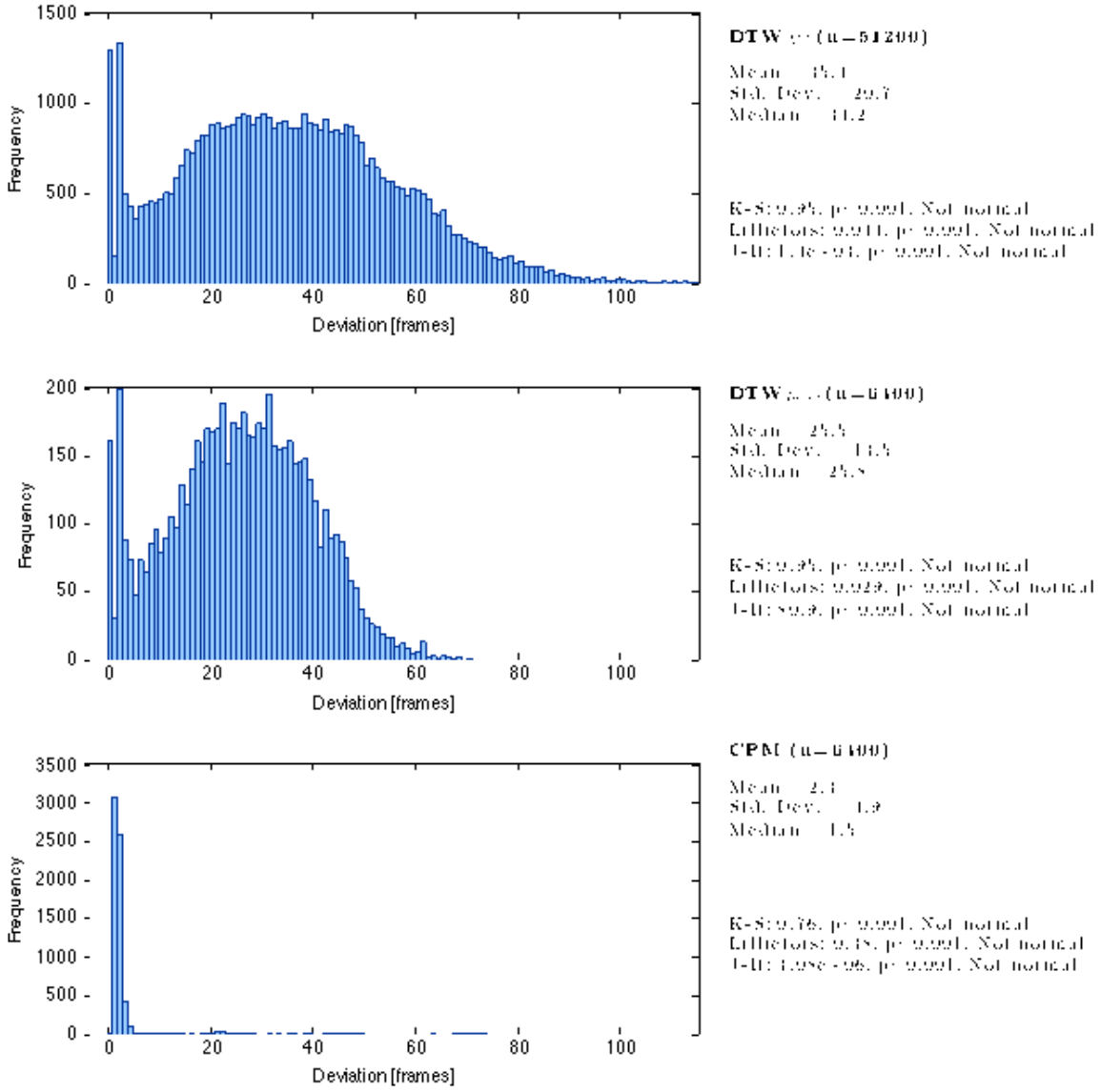

Fig. 5.7 Histogram of alignment deviations for alignments of eight recordings.

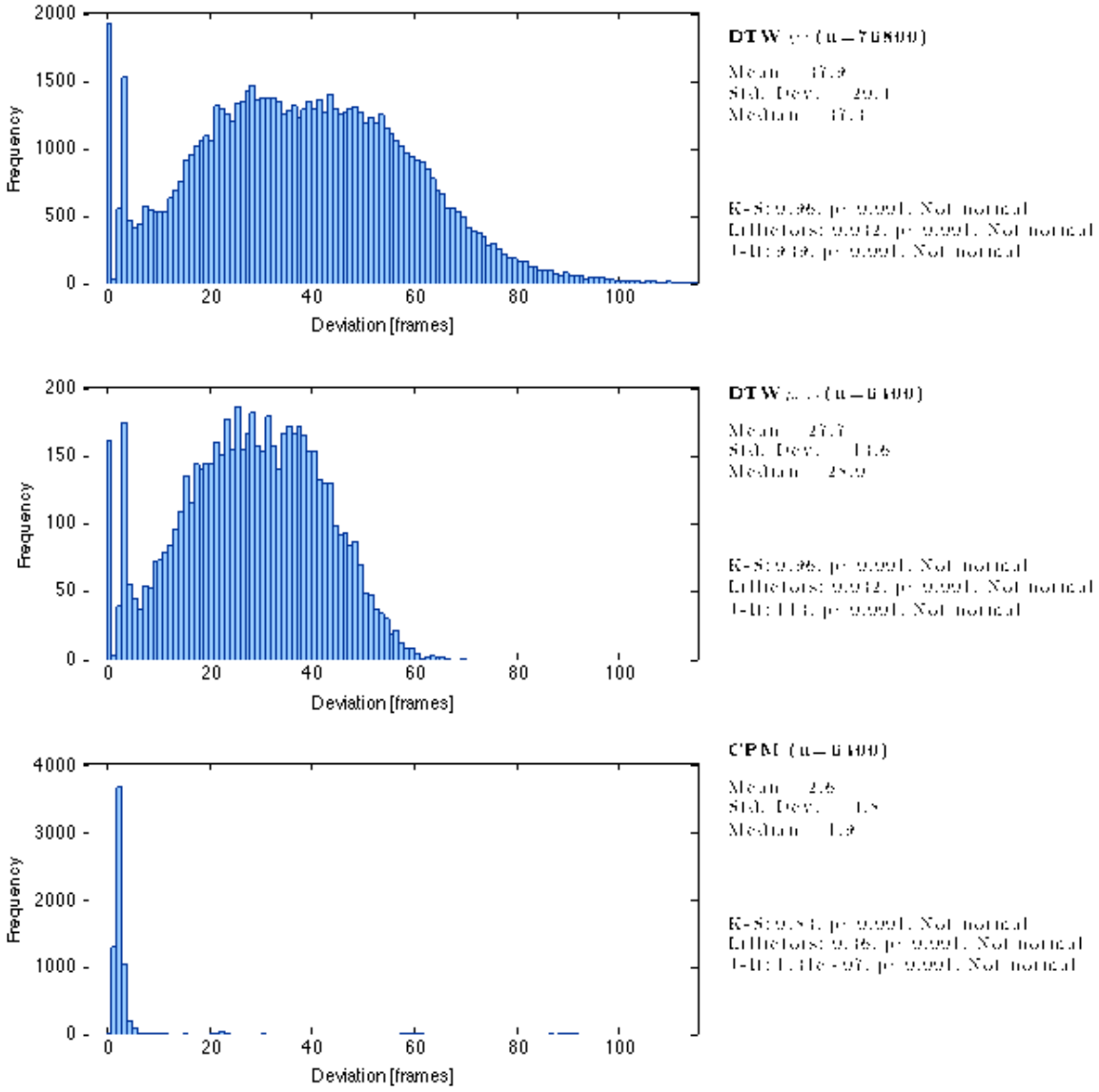

Fig. 5.8 Histogram of alignment deviations for alignments of twelve recordings.

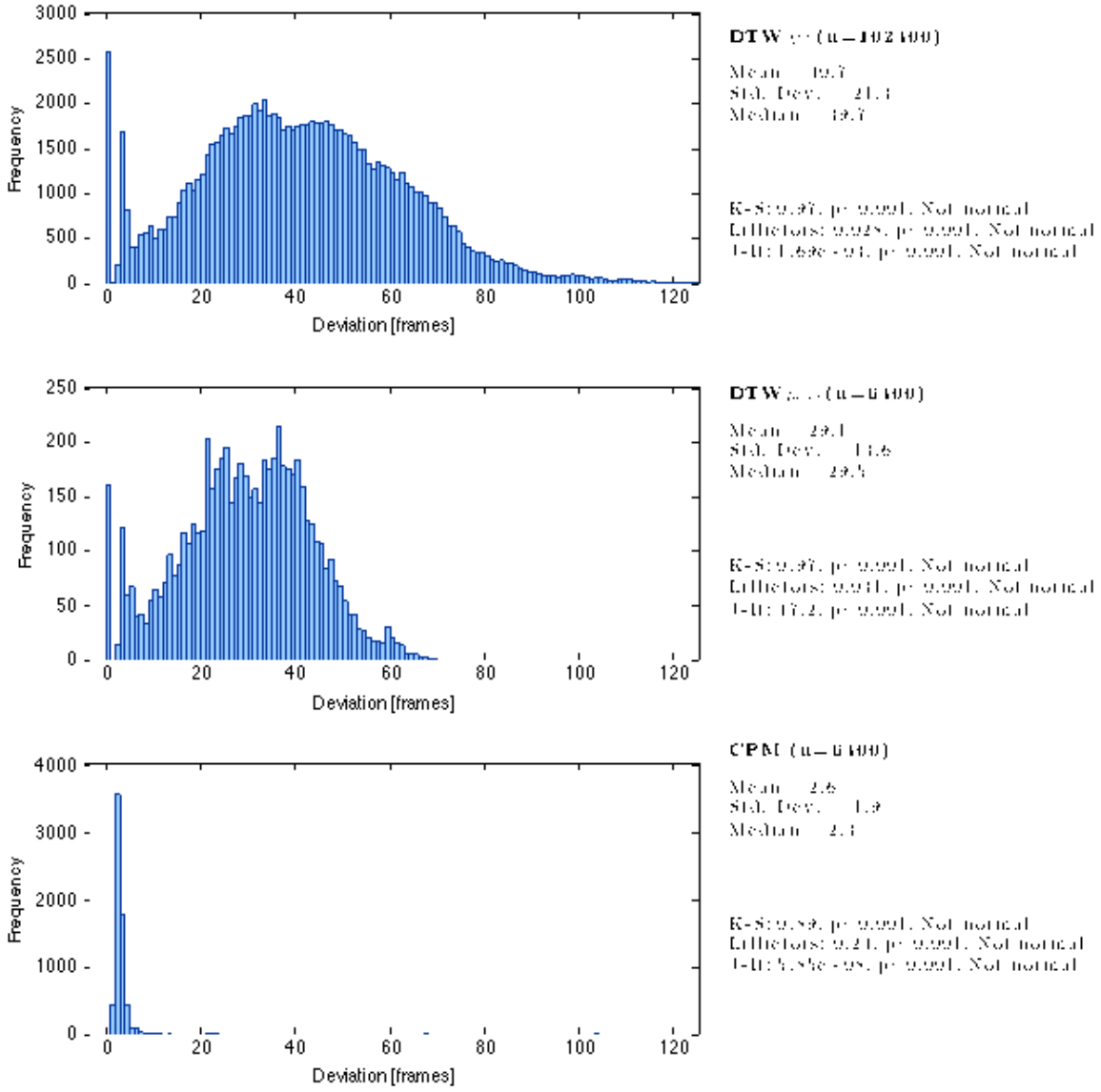

Fig. 5.9 Histogram of alignment deviations for alignments of sixteen recordings.

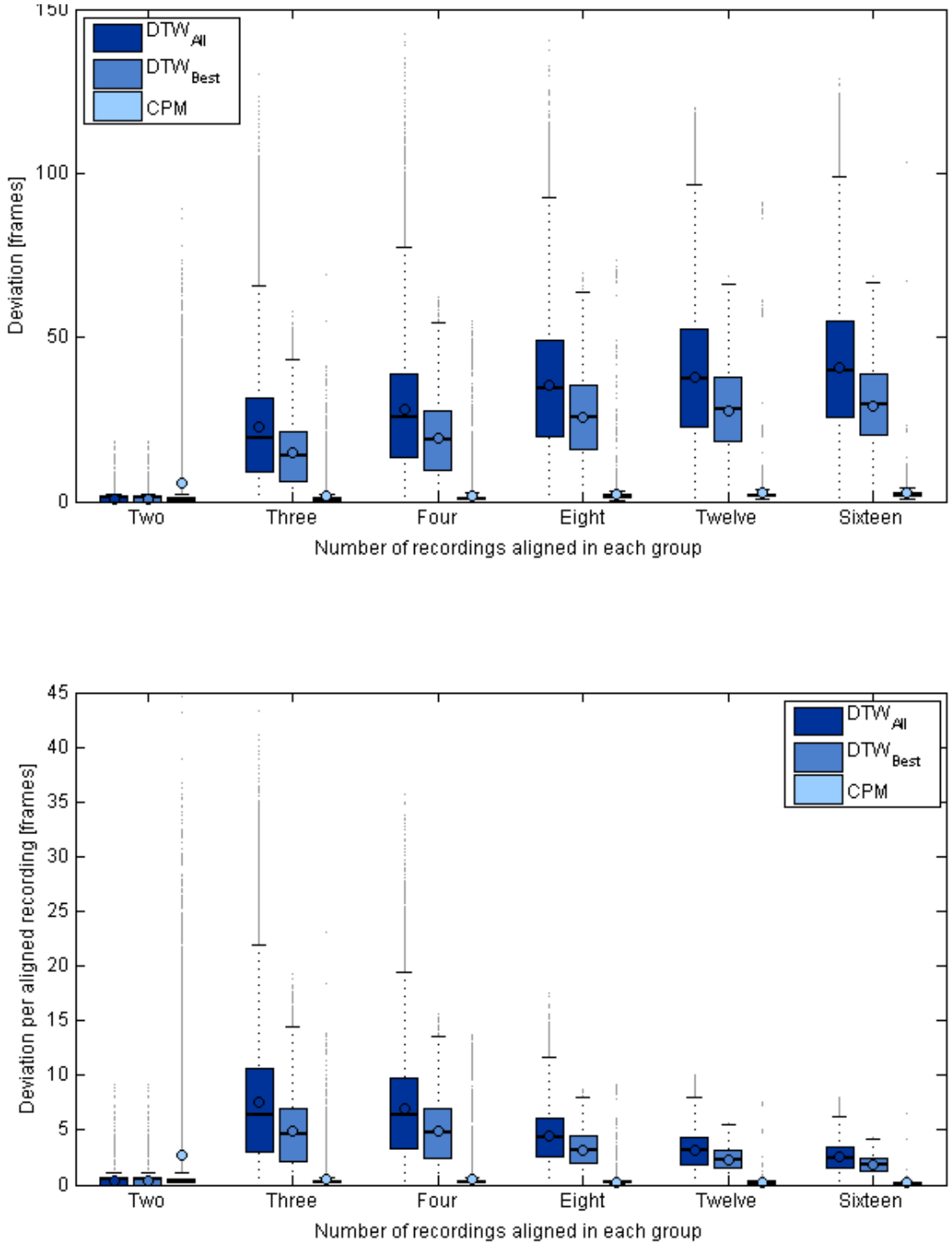

Fig. 5.10 A comparison of multiple alignment by iterative pairwise alignment with DTW and simultaneous alignment with the CPM, for alignment groups of various sizes. The top figure is plotted against deviation; the bottom figure is plotted against deviation-per-recording. Each box spans the interquartile range (the 25th to 75th percentile) of the deviation distribution. The whiskers form the boundaries between extreme deviation values and outliers. The dark horizontal line marks the median, the circle marks the mean. Lower whiskers for groups of size two, three, and four have a deviation of zero so are not shown.

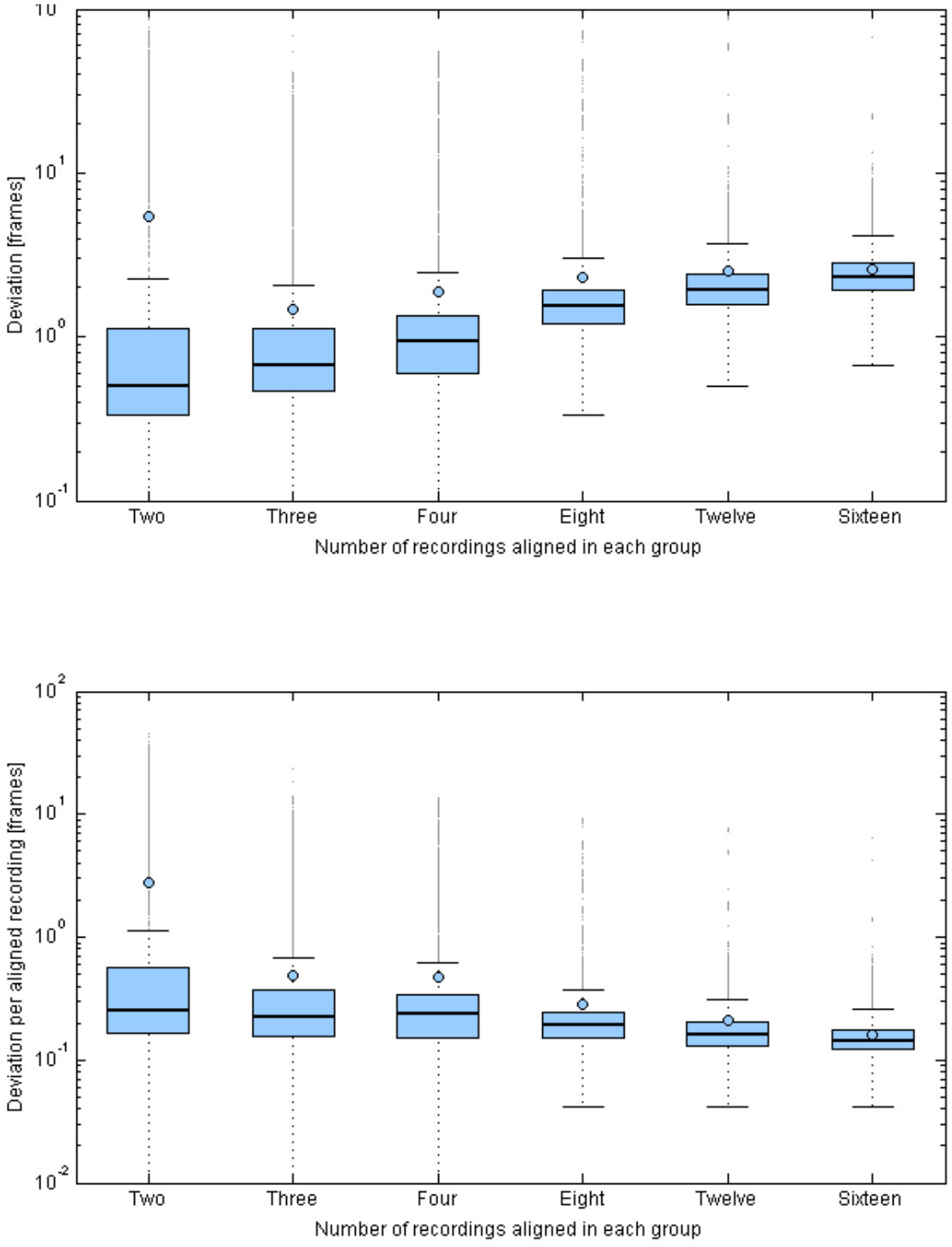

Fig. 5.11 Simultaneous multiple alignment by the CPM on alignment groups of various sizes, plotted on a logarithmic scale. The top figure is plotted against deviation; the bottom figure is plotted against deviation-per-recording. Each box spans the interquartile range (the 25^{th} to 75^{th} percentile) of the deviation distribution. The whiskers form the boundaries between extreme deviation values and outliers. The dark horizontal line marks the median, the circle marks the mean. Lower whiskers for groups of size two, three, and four have a deviation of zero so are not shown.

6—Conclusions

This thesis investigated audio-to-audio alignment with the CPM, an algorithm which had not previously been used to align musical audio. Chapter 1 introduced the concept of audio-to-audio alignment, as well as the algorithms and features commonly used to perform it. Chapter 2 presented a literature review that included a history of musical alignment, a summary of research applications, a discussion of feature and algorithm selection, and an overview of commercial and open-source alignment software. Chapter 3 explained the CPM algorithm and described how it was used to align musical audio in this thesis. Chapter 4 described the evaluation methodology, metrics, and implementation, as well as the choice of dataset.

Chapter 5 presented the experimental results. First, use of a reduced subset of the Chopin dataset was justified, as comparable DTW alignment results were obtained between the reduced dataset and the full Chopin dataset. Similarly, use of the Euclidean distance measure and a diagonally biased cost-path weighting were objectively justified. Next, it was found that the CPM successfully aligned pairs of recordings from the reduced Chopin dataset. Finally, it was found that alignment by the CPM considerably outperformed iterative pairwise alignment with DTW, for alignment of groups of three, four, eight, twelve, and sixteen recordings from the reduced Chopin dataset.

6.1 Summary of contributions

The first major contribution of this thesis is the demonstrated utility of the CPM as a tool for musical audio-to-audio alignment, both for pairwise alignment and for multiple alignment. This is especially significant given the dearth of multiple alignment algorithms available for aligning audio, and the accessibility of the CPM through the preexisting, open-source MATLAB toolbox.

The second major contribution of this thesis is the quantified comparison of two multiple alignment approaches, simultaneous alignment via the CPM versus iterative pairwise alignment via DTW. To the best of our knowledge, such a comparison of approaches to multiple audio-to-audio alignment has not previously been published. It was found that the simultaneous approach performed substantially better than the iterative pairwise approach. While it is possible that the iterative approach implemented here could be improved through a more strategic combination of recordings, these results indicate that care is needed when choosing an algorithmic approach to a multiple alignment task.

6.2 Future work

The results of this thesis suggest three directions for future work: further investigation of audio-to-audio alignment in general, fine-tuning the CPM for improved audio-to-audio alignment, and investigating additional CPM features.

6.2.1 Further audio-to-audio alignment research

One of the greatest limitations to investigating audio-to-audio alignment, both for this thesis and in the field in general, is a lack of ground truth corpora covering a variety of musical recordings. To understand how any alignment algorithm, pairwise or simultaneous, performs on audio with different types of variations, a greater variety of corpora are needed. New corpora need to include audio in a variety of genres, and with a variety in variation across the different recordings: variation that is both intentional, such as with melodic ornamentation or differing instrumentation; and variation in audio quality, such as recordings that contain different amounts of background noise or that have been recorded from performances in different environments. For this thesis, the Chopin dataset facilitated an effective preliminary study of alignment with the CPM, as it had been used for previous audio-to-audio alignment research using DTW. To obtain a better understanding of the intricacies and situational performance of different algorithms, however, a wider variety of datasets are needed.

In addition to a greater variety of corpora, a standardized audio-to-audio alignment evaluation framework, similar to the evaluation frameworks used by the Music Information Retrieval Evaluation eXchange (MIREX), would facilitate a standardized comparison of alignment performance for different audio features, algorithms, and

corpora. As mentioned briefly in Chapter 2, MIREX is an annual competition among cutting-edge solutions to music information retrieval (MIR) tasks (Downie 2008). A formal evaluation framework and metric for each task serves to standardize algorithm evaluation within the MIR community. The evaluation code developed for this thesis could potentially be generalized into such a framework for audio-to-audio alignment.

6.2.2 Improving audio-to-audio alignment by the CPM

It seemed that many of the larger alignment deviations of the CPM alignments occurred during the beginning and ending regions of the recordings. Forced endpoint boundary constraints would almost certainly improve the alignment of these sections, just as endpoint boundary constraints promote a global over partial alignment approach in DTW.

The CPM has computational limitations in keeping with other HMM-based models. Alignments of recordings with many frames (such as long recordings and recordings with high feature resolution) or feature vectors with high dimensionality or both require prohibitively long computation times. To work with longer recordings without shortening the input signals by increasing the frame resolution, a multiscale alignment approach could be performed. This implementation could be modeled after the multiscale DTW approach taken by Müller et al. (2006).

6.2.3 Beyond the basic CPM

The CPM was designed to perform both alignment and difference detection, although this thesis focused entirely on the alignment functionality. The difference detection functionality finds regions of high and low similarity across the aligned recordings, making it a potentially valuable tool for music research applications such as performance analysis. Since basic alignment of musical recordings with the CPM has been deemed successful, it is now worth examining the CPM's difference detection utility. Unlike alignment, little precedent has been set for a quantitative evaluation of difference detection.

Finally, the authors of the CPM extended it to create the hierarchical Bayesian continuous profile model (HB-CPM), which contains a classification functionality such that a set of recordings can be grouped into two or more classes during alignment.[1]

[1] This functionality is included in the CPM Toolbox for MATLAB.

Examples of potential classes could include groups for a set of recordings made in a recording hall versus a set made at an open-air music festival, or groups based on instrumentation—woodwind versus brass versus mallet, for a set of solo recordings. Class-based alignment would perhaps reduce overall misalignment when aligning a group of recordings with distinctive noise or timbre differences. Additionally, difference detection could then be performed across classes, rather than across all individual recordings.[2]

6.3 Coda

Given the proliferation of multiple recordings of the same piece (such as those taken by fans of a particular band at many of the band's different concerts, or from multiple recording takes in a studio) tools such as alignment algorithms are more relevant than ever for facilitating a large-scale, data-driven analysis of audio. It is hoped that this thesis has demonstrated the utility of the CPM as an effective, accessible tool for performing multiple alignment of musical recordings.

[2] In this latter case, each individual recording is considered its own class.

APPENDIX A—CHOPIN DATASET EXCERPTS

THE musical scores from the two works performed in the Chopin dataset are shown in Figures A.1 and A.2. Only the Ballade (Figure A.1) was used for evaluation in this thesis.

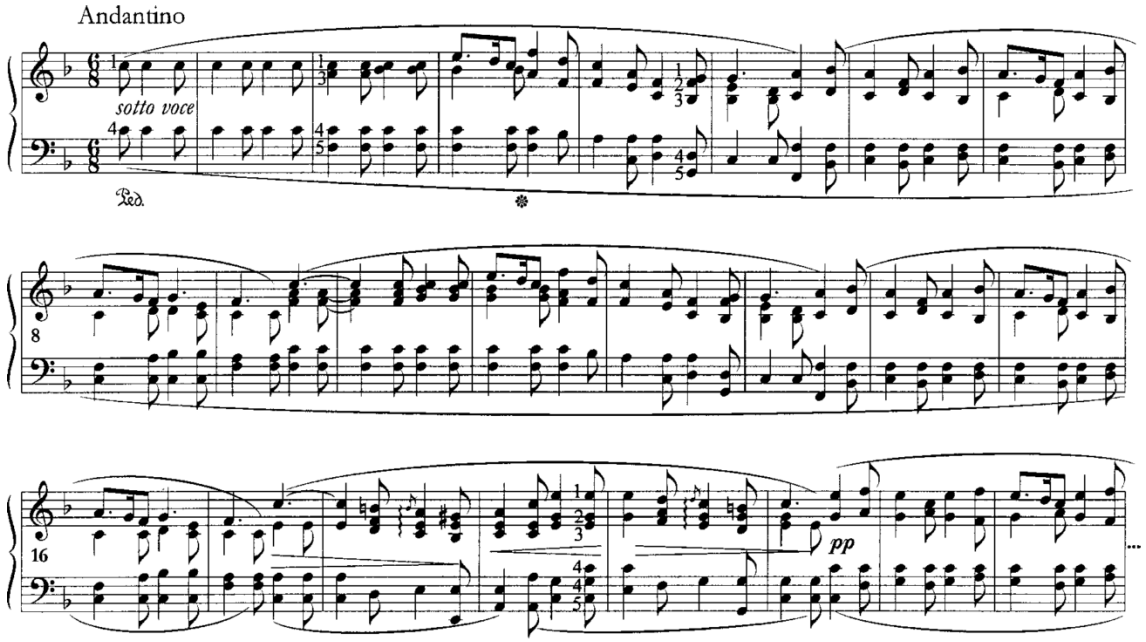

Fig. A.1 The Chopin dataset excerpt of Ballade in F major, op. 38, by Frédéric Chopin. Reprinted from Goebl (2001): "Score prepared with computer software following the Henle Urtext Edition." The reduced Chopin dataset used in this thesis spans the first 40 score events of this excerpt (approximately nine measures).

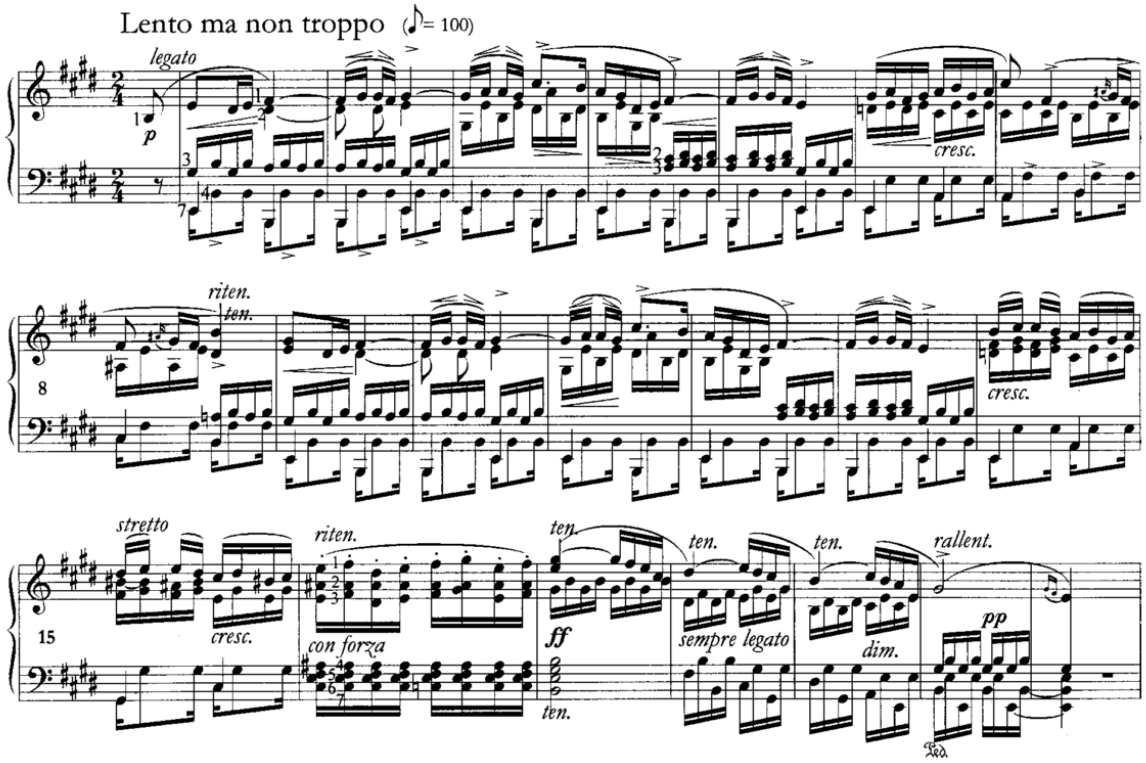

Fig. A.2 The Chopin dataset excerpt of Etude in E major, op. 10 No. 3, by Frédéric Chopin. Reprinted from Goebl (2001): "Score prepared with computer software following the Paderewski Edition."

References

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. 1990. Basic local alignment search tool. *Journal of Molecular Biology 215*: 403–10.

Amor, J. D., and C. J. James. 2010. Behavioral pattern detection from personalized ambient monitoring. In *Proceedings of the IEEE Conference on Engineering in Medicine and Biology Society*, Buenos Aires, Argentina, 5193–6.

Antonopoulos, I., A. Pikrakis, S. Tehodoridis, O. Cornelis, D. Moelants, and M. Leman. 2007. Music retrieval by rhythmic similarity applied on Greek and African traditional music. *Austrian Computer Society*: 6–9.

Basaran, D., A. T. Cemgil, and E. Anarim. 2011. Model based multiple audio sequence alignment. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 13–6.

Bellman, R. 1952. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences*: 716–9.

Bloom, P. 1984. Use of dynamic programming for automatic synchronization of two similar speech signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 67–72.

Bohak, C., and M. Marolt. 2012. Finding repeating stanzas in folk songs. In *Proceedings of the International Conference on Music Information Retrieval*, Porto, Portugal, 451–6.

Camarena-Ibarrola, A., and E. Chávez. 2006. On musical performances identification, entropy and string matching. In *Proceedings of the Mexican International Conference on Artificial Artificial Intelligence*, Apizaco, Mexico, 952–62.

Camarena-Ibarrola, A., and E. Chávez. 2010. Real time tracking of musical performances. *Advances in Soft Computing*: 138–48.

Cano, P., E. Batlle, H. Mayer, and H. Neuschmied. 2002. Robust sound modeling for song detection in broadcast audio. In *Proceedings of the Audio Engineering Society*, Munich, Germany.

Carabias, J. J., F. J. Rodriguez, P. Vera, P. Cabañas, F. J. Cañadas, and N. Ruiz. 2012. A real-time NMF-based score follower for MIREX 2012. *Music Information Retrieval Evaluation eXchange (MIREX)*.

Chan, S., A. Wong, and D. Chiu. 1992. A survey of multiple sequence comparison methods. *Bulletin of Mathematical Biology* 54 (4): 563–98.

Dannenberg, R. 2007. An intelligent multi-track audio editor. In *Proceedings of the International Computer Music Conference*, Copenhagen, Denmark, 89–94.

Dannenberg, R. B. 1984. An on-line algorithm for real-time accompaniment. In *Proceedings of the International Computer Music Conference*, Paris, France, 193–8.

Dannenberg, R. B., and N. Hu. 2003. Polyphonic audio matching for score following and intelligent audio editors. In *Proceedings of the International Computer Music Conference*, Singapore, 507–13.

Dempster, A., N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39 (1): 1–38.

Devaney, J. 2011. An empirical study of the influence of musical context on intonation practices in solo singers and SATB ensembles. PhD dissertation, McGill University.

Devaney, J., and D. Ellis. 2009. Handling asynchrony in audioscore alignment. In *Proceedings of the International Computer Music Conference*, Montreal, Canada, 29–32.

Devaney, J., M. I. Mandel, and D. Ellis. 2009. Improving MIDI-audio alignment with acoustic features. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 19–22.

Devaney, J., M. I. Mandel, D. P. W. Ellis, and I. Fujinaga. 2011. Automatically extracting performance data from recordings of trained singers. *Psychomusicology: Music, Mind and Brain* 21 (1-2): 108–36.

Dixon, S. 2005a. An on-line time warping algorithm for tracking musical performances. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Edinburgh, United Kingdom.

Dixon, S. 2005b. Live tracking of musical performances using on-line time warping. In *Proceedings of the International Conference on Digital Audio Effects*, Madrid, Spain, 1–6.

Dixon, S., and G. Widmer. 2005. MATCH: A music alignment tool chest. In *Proceedings of the International Conference on Music Information Retrieval*, London, United Kingdom, 492–7.

Downie, J. S. 2008. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology* 29 (4): 247–55.

Downie, J. S., and M. Bay. 2008. Audio cover song identification: MIREX 2006-2007 results and analyses. In *Proceedings of the International Conference on Music Information Retrieval*, Philadelphia, PA, 468–73.

Duan, Z., and B. Pardo. 2011. A state space model for online polyphonic audio-score alignment. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Prague, Czech Republic, 197–200.

Ellis, D. P. 2003. Dynamic time warp (DTW) in MATLAB, web resource. `www.ee.columbia.edu/~dpwe/resources/matlab/dtw/`.

Ellis, D. P. 2008. Aligning MIDI score to music audio, web resource. `www.labrosa.ee.columbia.edu/matlab/alignmidiwav/`.

Ellis, D. P. W., and C. Cotton. 2007. The 2007 LabROSA cover song detection system. *Music Information Retrieval Evaluation eXchange (MIREX)*.

Ellis, D. P. W., and G. E. Poliner. 2007. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honolulu, HI, 1429–32.

Ewert, S., and M. Müller. 2009. Refinement strategies for music synchronization. In *Proceedings of the Computer Music Modeling and Retrieval Conference on Genesis of Meaning in Sound and Music*, Copenhagen, Denmark, 147–65.

Ewert, S., and M. Müller. 2012. Using score-informed constraints for NMF-based source separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan.

Ewert, S., M. Müller, and R. B. Dannenberg. 2009. Towards reliable partial music alignments using multiple synchronization strategies. In *Proceedings of the International Conference on Adaptive Multimedia Retrieval*, Madrid, Spain, 35–48.

Ewert, S., M. Müller, and P. Grosche. 2009. High resoultion audio synchronization using chroma onset features. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, 2–5.

Ewert, S., M. Müller, D. Müllensiefen, M. Clausen, and G. Wiggins. 2009. Case study "Beatles Songs"—what can be learned from unreliable music alignments? In *Proceedings of the Seminar on Knowledge Representation for Intelligent Music Processing*, Dagstuhl, Germany.

Field, A. 2005. *Discovering statistics using SPSS*. London, United Kingdom: SAGE Publications.

Foote, J. 1999. Visualizing music and audio using self-similarity. In *Proceedings of the ACM International Conference on Multimedia*, Orlando, FL, 77–80.

Foote, J. 2000. Automatic Audio Segmentation Using A Measure of Audio Novelty. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Los Alamitos, CA, 452–5.

Foote, J., and M. Cooper. 2001. Visualizing musical structure and rhythm via self-similarity. In *Proceedings of the International Computer Music Conference*, Havana, Cuba.

Fremerey, C., M. Clausen, S. Ewert, and M. Müller. 2009. Sheet music-audio identification. In *Proceedings of the International Conference on Music Information Retrieval*, Kobe, Japan, 645–50.

Fremerey, C., M. Müller, F. Kurth, and M. Clausen. 2008. Automatic mapping of scanned sheet music to audio recordings. In *Proceedings of the International Conference on Music Information Retrieval*, Philadelphia, PA, 413–8.

Gerber, T., M. Dutasta, and L. Girin. 2012. Professionally-produced music separation guided by covers. In *Proceedings of the International Conference on Music Information Retrieval*, Porto, Portugal, 85–90.

Giorgino, T. 2009. Computing and visualizing dynamic time warping alignments in R: The `dtw` package. *Journal of Statistical Software* 31 (7): 1–24.

Goebl, W. 2001. Melody lead in piano performance: Expressive device or artifact? *The Journal of the Acoustical Society of America* 110 (1): 563–72.

Gómez, E. 2006. Tonal description of music audio signals. PhD dissertation, University Pompeu Fabra.

Grosche, P., M. Müller, and J. Serrà. 2012. Audio Content-Based Music Retrieval. *Multimodal Music Processing 3*: 157–74.

Harte, C. 2010. Towards automatic extraction of harmony information from music signals. PhD dissertation, University of London.

Haussler, D., A. Krogh, I. S. Mian, and K. Sjolander. 1993. Protein modeling using hidden Markov models: Analysis of globins. In *Proceedings of the IEEE Hawaii International Conference on System Sciences*, Los Alamitos, CA, 792–802.

Hu, N., R. Dannenberg, J. Hailpern, U. Kurokawa, G. Wakefield, and M. Bartsch. 2005. Scorealign. `www.cs.cmu.edu/~music/alignment`.

Hu, N., and R. B. Dannenberg. 2005. A bootstrap method for training an accurate audio segmenter. In *Proceedings of the International Conference on Music Information Retrieval*, London, United Kingdom, 223–9.

Hu, N., R. B. Dannenberg, and G. Tzanetakis. 2003. Polyphonic audio matching and alignment for music retrieval. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 185–8.

Itakura, F. 1975. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23 (1): 67–72.

Jarque, C., and A. Bera. 1987. A test for normality of observations and regression residuals. *International Statistical Review* 55 (2): 163–172.

Jehan, T. 2005. Creating music by listening. PhD dissertation, Massachusetts Institute of Technology.

Kirchhoff, H., and A. Lerch. 2011. Evaluation of features for audio-to-audio alignment. *Journal of New Music Research* 40 (1): 27–41.

Konz, V., and M. Müller. 2012. A cross-version approach for harmonic analysis of music recordings. *Multimodal Music Processing 3*: 53–72.

Kruskal, J., and M. Liberman. 1983. The symmetric time-warping problem: From continuous to discrete. In D. Sankoff and J. Kruskal (Eds.), *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, 125–59. Reading, MA: Addison-Wesley.

Kruskal, J., and D. Sankoff. 1983. An anthology of algorithms and concepts for sequence comparison. In D. Sankoff and J. Kruskal (Eds.), *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, 293–6. Reading, MA: Addison-Wesley.

Kruskal, W. H., and W. A. Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association* 47 (260): 583–621.

Kurth, F., and M. Müller. 2008. Efficient index-based audio matching. *IEEE Transactions on Audio, Speech and Language Processing* 16 (2): 382–95.

Kurth, F., M. Müller, D. Damm, C. Fremerey, A. Ribbrock, and M. Clausen. 2005. Syncplayer-an advanced system for multimodal music access. In *Proceedings of the International Conference on Music Information Retrieval*, London, United Kingdom, 381–8.

Kurth, F., M. Müller, and C. Fremerey. 2007. Automated synchronization of scanned sheet music with audio recordings. In *Proceedings of the International Conference on Music Information Retrieval*, Vienna, Austria, 261–6.

Leone, F. C., L. Nelson, and R. Nottingham. 1961. The folded normal distribution. *Technometrics* 3 (4): 543–550.

Lilliefors, H. 1967. On the Kolmogorov-âĂŞSmirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association 62*: 399–402.

Listgarten, J. 2007. Analysis of sibling time series data: Alignment and difference detection. PhD dissertation, University of Toronto.

Listgarten, J., R. Neal, S. Roweis, and A. Emili. 2005. Multiple alignment of continuous time series. *Advances in Neural Information Processing Systems 17*: 817–24.

Listgarten, J., R. M. Neal, S. T. Roweis, P. Wong, and A. Emili. 2006. Difference detection in LC-MS data for protein biomarker discovery. *Bioinformatics 23*: 198–204.

Martin, B., D. G. Brown, P. Hanna, and P. Ferraro. 2012. BLAST for audio sequences alignment: A fast scalable cover identification tool. In *Proceedings of the International Conference on Music Information Retrieval*, Porto, Portugal, 529–34.

MATLAB and MATLAB Signal Processing Toolbox. 2011. *Version 7.13.0.564 (R2011b)*. Natick, MA: The MathWorks Inc.

Meron, Y., and K. Hirose. 2001. Automatic alignment of a musical score to performed music. *Acoustical Science and Technology* 22 (3): 189–98.

Mongeau, M., and D. Sankoff. 1990. Comparison of musical sequences. *Computers and the Humanities* 24 (3): 161–75.

Montecchio, N., and A. Cont. 2011a. A unified approach to real time audio-to-score and audio-to-audio alignment using sequential Montecarlo inference techniques. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Prague, Czech Republic, 193–6.

Montecchio, N., and A. Cont. 2011b. Accelerating the mixing phase in studio recording productions by automatic audio alignment. In *Proceedings of the International Conference on Music Information Retrieval*, Miami, FL, 627–32.

Müller, M., and D. Appelt. 2008. Path-constrained partial music synchronization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, NV, 65–8.

Müller, M., and S. Ewert. 2008. Joint structure analysis with applications to music annotation and synchronization. In *Proceedings of the International Conference on Music Information Retrieval*, Philadelphia, PA, 389–94.

Müller, M., and S. Ewert. 2011. Chroma toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *Proceedings of the International Conference on Music Information Retrieval*, Miami, FL, 215–20.

Muller, M., S. Ewert, and S. Kreuzer. 2009. Making chroma features more robust to timbre changes. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, 1869–72.

Müller, M., P. Grosche, and F. Wiering. 2009. Robust segmentation and annotation of folk song recordings. In *Proceedings of the International Conference on Music Information Retrieval*, Kobe, Japan, 735–40.

Müller, M., P. Grosche, and F. Wiering. 2010. Automated analysis of performance variations in folk song recordings. In *Proceedings of the ACM International Conference on Multimedia*, Philadelphia, PA, 247–56.

Müller, M., and F. Kurth. 2006. Enhancing similarity matrices for music audio analysis. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, 437–40.

Müller, M., F. Kurth, and M. Clausen. 2005a. Audio matching via chroma-based statistical features. In *Proceedings of the International Conference on Music Information Retrieval*, London, United Kingdom, 288–95.

Müller, M., F. Kurth, and M. Clausen. 2005b. Chroma-based statistical audio features for audio matching. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 275–8.

Müller, M., H. Mattes, and F. Kurth. 2006. An efficient multiscale approach to audio synchronization. In *Proceedings of the International Conference on Music Information Retrieval*, Victoria, Canada, 192–7.

Nagano, H., K. Kashino, and H. Murase. 2002. Fast music retrieval using polyphonic binary feature vectors. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, 101–4.

Needleman, S. B., and C. D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48 (3): 443–53.

Niedermayer, B., and G. Widmer. 2010. A multi-pass algorithm for accurate audio-to-score alignment. In *Proceedings of the International Conference on Music Information Retrieval*, Utrecht, Netherlands, 417–22.

Niedermayer, B., G. Widmer, and C. Reuter. 2011. Version detection for historical musical automata. In *Proceedings of the Sound and Music Computing Conference*, Padova, Italy.

Orio, N., S. Lemouton, and D. Schwarz. 2003. Score following: State of the art and new developments. In *Proceedings of the Conference on New Interfaces for Musical Expression*, Montreal, Canada, 36–41.

Orio, N., and D. Schwarz. 2001. Alignment of monophonic and polyphonic music to a score. In *Proceedings of the International Computer Music Conference*, Havana, Cuba, 129–32.

Orio, N., and L. Zattra. 2007. Audio matching for the philological analysis of electroacoustic music. In *Proceedings of the International Computer Music Conference*, Copenhagen, Denmark, 157–64.

R Core Team. 2012. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. `www.R-project.org`.

Rabiner, L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77 (2): 257–87.

Ramona, M., and G. Peeters. 2011. Automatic alignment of audio occurrences: Application to the verification and synchronization of audio fingerprinting annotation. In *Proceedings of the International Conference on Digital Audio Effects*, Paris, France, 429–36.

Ross, J., T. Vinutha, and P. Rao. 2012. Detecting melodic motifs from audio for Hindustani classical music. In *Proceedings of the International Conference on Music Information Retrieval*, Porto, Portugal, 193–8.

Sakoe, H., and S. Chiba. 1971. A dynamic programming approach to continuous speech recognition. In *Proceedings of the International Congress on Acoustics*, Budapest, Hungary, 65–9.

Sakoe, H., and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1): 43–9.

Sakoe, H., and S. Chiba. 1990. Dynamic programming algorithm optimization for spoken word recognition. In A. Waibel and K.-F. Lee (Eds.), *Readings in Speech Recognition*, 159–65. San Mateo, California: Morgan Kaufmann Publishers, Inc.

Sanguansat, P. 2012. Multiple multidimensional sequence alignment using generalized dynamic time warping. *World Scientific and Engineering Academy and Society Transactions on Mathematics* 11 (8): 668–78.

Sapp, C. 2007. Comparative analysis of multiple musical performances. In *Proceedings of the International Conference on Music Information Retrieval*, Vienna, Austria, 497–500.

Schwarz, D. 2003. The Caterpillar system for data-driven concatenative sound synthesis. In *Proceedings of the International Conference on Digital Audio Effects*, London, United Kingdom, 135–40.

Serrà, J., and E. Gomez. 2007. A cover song identification system based on sequences of tonal descriptors. *Music Information Retrieval Evaluation eXchange (MIREX)*.

Serrà, J., E. Gómez, P. Herrera, and X. Serra. 2008. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech and Language Processing* 16 (6): 1138–51.

Serrà, J., X. Serra, and R. G. Andrzejak. 2009. Cross recurrence quantification for cover song identification. *New Journal of Physics* 11 (9).

Skopal, T., and B. Bustos. 2011. On nonmetric similarity search problems in complex domains. *Association for Computing Machinery Computing Surveys* 43 (4): 1–50.

Stammen, D. R., and B. Pennycook. 1993. Real-time recognition of melodic fragments using the dynamic timewarp algorithm. In *Proceedings of the International Computer Music Conference*, Tokyo, Japan, 232–5.

Stevens, S. S., J. Volkmann, and E. B. Newman. 1937. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America 8*: 185–90.

Tabus, I., V. Tabus, and J. Astola. 2012. Information theoretic methods for aligning audio signals using chromagram representations. In *Proceedings of the IEEE International Symposium on Communications Control and Signal Processing*, Roma, Italy, 2–4.

Thomas, V., S. Ewert, and M. Clausen. 2012. Fast intra-collection audio matching. In *Proceedings of the ACM International Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, New York, NY.

Turetsky, R. J., and D. P. W. Ellis. 2003. Ground-truth transcriptions of real music from force-aligned midi syntheses. In *Proceedings of the International Conference on Music Information Retrieval*, Baltimore, MD, 135–41.

Vercoe, B. 1984. The synthetic performer in the context of live performance. In *Proceedings of the International Computer Music Conference*, Paris, France, 199–200.

Viterbi, A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. 260–9.

Wang, L. 2012. Learning task-based robotic grasping with vision, haptics and proprioception. Master's thesis, Royal Institute of Technology.

Wilcoxon, F. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin* 1 (6): 80–3.

Xiong, B., and O. Izmirli. 2012. Audio-to-audio alignment using particle filters to handle small and large scale performance discrepancies. In *Proceedings of the International Computer Music Conference*, Ljubljana, Slovenia, 539–42.

Yang, C. 2001. Music database retrieval based on spectral similarity. In *Proceedings of the International Symposium on Music Information Retrieval*, Bloomington, IN, 37–8.

Zhou, F., and F. De la Torre. 2012. Generalized time warping for multi-modal alignment of human motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 1282–9.

Zwicker, E. 1961. Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *The Journal of the Acoustical Society of America* 33 (2): 248.