UNDERSTANDING FEATURES AND DISTANCE FUNCTIONS FOR MUSIC SEQUENCE ALIGNMENT

Özgür İzmirli
Center for Arts and Technology
Computer Science Department
Connecticut College
oizm@conncoll.edu

Roger B. Dannenberg
School of Computer Science
Carnegie Mellon University
rbd@cs.cmu.edu

ABSTRACT

We investigate the problem of matching symbolic representations directly to audio based representations for applications that use data from both domains. One such application is score alignment, which aligns a sequence of frames based on features such as chroma vectors and distance functions such as Euclidean distance. Good representations are critical, yet current systems use ad hoc constructions such as the chromagram that have been shown to work quite well. We investigate ways to learn chromagram-like representations that optimize the classification of "matching" vs. "non-matching" frame pairs of audio and MIDI. New representations learned automatically from examples not only perform better than the chromagram representation but they also reveal interesting projection structures that differ distinctly from the traditional chromagram.

1. INTRODUCTION

Score alignment [4], score following [3], chord and key recognition [6, 7] chorus spotting [1, 8], audio-to-audio alignment [9, 13] and music structure analysis [2, 11] are all tasks where it is useful to compare two segments of music. A common representation for this is the chromagram [1], a sequence of chroma vectors, where each vector typically has 12 elements and each element represents the energy corresponding to one pitch class in the spectrum but not necessarily one pitch class in the score. Most algorithms use a distance function in conjunction with the chromagram representation to measure the similarity between frames. While it may be obvious, especially in hindsight, why the chromagram works well in many applications, it should be noted that the chromagram is a contrived representation, and there is no reason to believe it should be optimal. Very little research has been conducted on alternative ways to compare audio to audio let alone audio to symbolic representations. The existing approaches are generally domain specific. For example, in [12] the chromagram is made less timbre dependent by discarding the lower mel-frequency cepstral coefficients and then projecting the remaining coefficients onto the twelve chroma bins. Another example can be found in [15] in which a binary chroma similarity measure is used for alignment in the context of cover song detection. In this work, we explore various ways to derive good features and functions from real data. We specifically look at the problem of score alignment directly from a MIDI representation to audio without going through a synthesized version of the MIDI data. In this paper we give a formulation based on the score alignment task; however, results should be applicable to all other problems that require frame-based comparison. The goal of this work is to gain insight into why the chromagram works in practice and to learn what modifications might make it work even better. Our results suggest that there is room for at least some improvement.

2. THE SCORE ALIGNMENT TASK

Our work is aimed at optimizing score alignment: finding a mapping from a symbolic score or standard MIDI file to an audio recording. The basic algorithm transforms both the MIDI file and the audio file into chromagrams **A** and **B**, which are sequences of chroma vectors. We will denote the chroma vector corresponding to the i^{th} time frame (column) of **A** as $\mathbf{A}_i$. Then, construct a distance matrix $\mathbf{D}_{i,j} = f(\mathbf{A}_i, \mathbf{B}_j)$, where f is a distance function. The idea is that f is small when $\mathbf{A}_i$ is "similar" to $\mathbf{B}_j$ and large otherwise. Often, f is based on the cosine distance, correlation distance, or Euclidean distance from $\mathbf{A}_i$ to $\mathbf{B}_j$. The next step uses dynamic programming to find the lowest-cost path from $\mathbf{D}_{0,0}$ to $\mathbf{D}_{m-1,n-1}$, where m and n are the number of frames in **A** and **B** respectively.

Path smoothing or constraints may be useful to obtain even more accurate alignment. Experience has shown that the chromagram representation for audio, and a chromagram-like representation for MIDI data [9] results in a very robust score alignment algorithm. However, the chromagram is an arbitrary choice. There are many other possible features, including the spectrum and mel cepstrum, and even the chromagram has parameters including the range of spectral bins considered. How can we search for better representations and distance functions?

3. THE LOG-FREQUENCY SPECTRUM OR "SEMIGRAM" REPRESENTATION

Although we are interested in learning better representations and distance functions, it would be difficult to learn a relationship between audio and symbolic representations starting from raw signal frames and raw MIDI data. To simplify the representation, we use a magnitude spectrum with bins logarithmically spaced by semitones (12 bins per octave). The input audio is downsampled to a sampling rate of 11025 Hz. A frame duration of 93 ms with 50% overlap is used. This representation is able to resolve semitone differences in frequency across the spectrum with far less data than the standard magnitude spectrum where each bin has a constant bandwidth. By analogy to the spectrogram, we call this representation the *semigram* **S**: a matrix where each column is a *semi vector* and each semi vector element represents the magnitude associated with the frequency range of one semitone. We note that the traditional 12-element chromagram can be understood as an octave-folded version of the semigram.

For MIDI data we construct a similar representation, a matrix **R** (also called a semigram), where each column represents a time window and each row represents a pitch (key number). If only one note is sounding in the time window at a given pitch, the matrix element is the note's MIDI velocity. If the note is not on during the entire time window, the velocity is weighted by the fraction of time the note is on. If there is more than one note on at the given pitch, the maximum of the weighted velocities is used.

4. TRAINING DATA FOR LEARNING PROJECTIONS

One way to search for good distance functions is simply to attempt alignment with various parameter settings, but this kind of evaluation is difficult. How do we score alignments? And if the chromagram is already robust, then it might take a huge number of examples to find enough failure cases for another method to show improvement.

Another possibility is to change the task. In our study, we use a classification task that labels frame pairs as "matching" or "non-matching." We assume that optimizing performance on this task will also be very good for the alignment task. We derive labeled training data (for supervised machine learning) from aligned scores, using 7 orchestra and wind ensemble recordings from one collection and 2 sets of 20 pieces from the RWC classical collection, as listed in Table 1. CLA1 consists of mostly symphonic pieces whereas CLA2 is a random selection of pieces with different combinations of instruments.

The alignment for the orchestra and wind ensemble recordings was done using chromagrams, but post processed with some spline fitting and smoothing techniques that generally improved the perceptual alignment. The alignment for the RWC pieces are taken from alignment data provided by Ewert, Müller, and Grosche [5]. The alignments of the corresponding scores were verified to be acceptable by listening to the MIDI synthesized versions simultaneously with the original audio.

From the aligned data, it is simple to extract all matching frames. To increase the number of matching frames and reduce overfitting to specific keys, we transpose the matching frames up to +6 and –5 semitone steps, thus covering all 12 chromatic degrees. To obtain non-matching frames, we select a random frame from audio for each frame from the MIDI data. These randomly selected pairs will run the gamut from very similar to very different, but for training purposes, we consider them all to be examples of "non-matching." For the training, the number of "non-matching" frames is equal to the number of "matching" frames including transpositions. All audio listed in the table was used for training, resulting in about 10^6 matching and the same number of non-matching frame pairs after transposition.

Table 1. The training data.

Recording	ID	Duration(secs.)
Tarantella from Incidental Suite, C. T. Smith	TAR	127
Nocturne from Incidental Suite, C. T. Smith	NOC	351
The Music of Disneyland, arr. by J. Brubaker	DIS	499
Medieval Legend, M. Story	LEG	248
The Travelin' Hat Rag, D. Bobrowitz	HAT	162
The Thunderer, J. Sousa	THU	148
Rondo from Incidental Suite, C. T. Smith	RON	168
RWC Classical Music Collection (20 pieces)	CLA1	1182
RWC Classical Music Collection (20 pieces)	CLA2	1192

5. LEARNING A FEATURE VECTOR

As a preliminary study, to find a good distance function for alignment, we trained a multi-layer perceptron neural network to classify semi vector pairs as "matching" or "non-matching." The inputs were midi and spectral vectors and the output was trained to be 0 or 1 based on whether the vectors were matching or not. 20 hidden nodes were used. We trained this on a particular set of three pieces: HAT, LEG and RON. These yielded 92.3% accuracy on the training data. Testing individually we obtained HAT: 89.5%, LEG: 93.0%, RON: 90.5%, TAR: 83.8%, NOC: 85.3%, DIS: 87.7 % and THU: 86.9%. We also trained the neural network separately on the 20-piece RWC sets and tested on the remaining 36 pieces in that set. We obtained 94.0% and 86.4% accuracy for CLA1, and 90.2% and 89.0% accuracy for CLA2 on the training and test data respectively. These results showed us that a model of this nature could generalize a matched-unmatched classification quite well with the given input representations. To reiterate our aim in this work, we are interested in understanding why chroma vectors work so well, whether they work better than a trained neural net, and whether variations can work even better. After all, the chroma vector is basically one particular projection from the semi vector. We can use machine learning to explore the space of projections and visualize the results to gain better understanding of the nature of the projections that work better.

Let us first write the chroma vector computation as a projection. For MIDI data, we have the $p \times m$ semigram

R. We define an $r \times p$ matrix **L** (r = 12) that projects each semi vector (column) of **R** to a chroma vector. Similarly, we define an $r \times q$ matrix **M** to project the audio q x m semigram **S** to chroma vectors:

$$\mathbf{A} = \mathbf{LR} \quad (2)$$

$$\mathbf{B} = \mathbf{MS} \quad (3)$$

Matrices A and B consist of pairs of feature vectors resulting from the respective projections in L and M. We first use this framework with fixed projections and then generalize the approach by training these projections to better understand their nature and compare them with the commonly used ones. Figure 1 illustrates the standard form of M (and similarly for L), which collapses octaves in the semigram to form a 12-element chroma vector. Note that the frequency ranges corresponding to the audio semigram (the horizontal axis) are labeled with midi numbers.

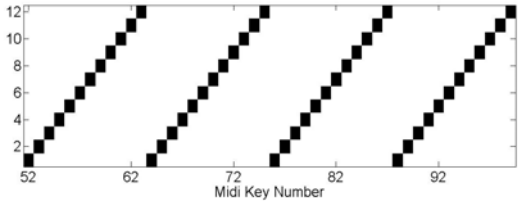

Figure 1. The conventional projection from semigram (log-frequency discrete magnitude spectrogram) to chromagram.

Next, a standard distance is taken between two corresponding feature vectors to obtain a measure of similarity. Hence, the required distance for the score alignment algorithm in terms of two input semi vectors $\mathbf{R}_i$ and $\mathbf{S}_j$ is given by $f(\mathbf{A}_i, \mathbf{B}_j)=\boldsymbol{C}(\mathbf{LR}_i, \mathbf{MS}_j)$, where f represents the desired distance and $\boldsymbol{C}$ is the centered cosine distance (found by first removing the means of the vectors and then calculating the cosine distance). To obtain a binary output ("matched" or "non-matched"), the distance is compared to a fixed threshold. This result is used for evaluation, but for training, we use the continuous real value as the output and try to train the system to output a zero (0) or one (1) value.

Now, suppose we generalize the chromagram to allow *any* projection. Although this is not a neural network, the back-propagation algorithm can be used to learn weights for the matrices **L** and **M**. The basic idea is to evaluate the partial derivative of the output with respect to each element of each matrix. Then, for each training example, the partial derivative for each coefficient is scaled by the output error, multiplied by a small rate parameter, and subtracted from the coefficient, thus adjusting each coefficient in a direction that would move the output closer to the correct value. This update is applied to all elements in the **M** and **L** matrices for all training frame pairs, and this process is iterated many times until the output converges. Given a large enough dimension r, this gradient descent algorithm will normally converge to a local optimum.

We can write $\boldsymbol{C}(\mathbf{LR}_i, \mathbf{MS}_j)$ as $D(x_k, \mathbf{R}, \mathbf{S})$ where x_k is some element of **M** or **L**, letting the remainder of **M** and **L** be constants for the moment. We can then evaluate $D(x_k + eps, \mathbf{R}, \mathbf{S}) - D(x_k, \mathbf{R}, \mathbf{S})$ to estimate the partial derivative of D with respect to x_k. The learning algorithm is as follows:

```
while convergence criterion not met
   for all pairs R and S
      for each parameter indexed by k
         delta_k = D(x_k + eps, R, S) - D(x_k, R, S)
         error_k = D(x_k, R, S) - GT
         new x_k = x_k - alpha * error_k * delta_k
```

In this algorithm, eps is a small number used to calculate the derivative, GT is the ground truth and it has a value of 0 when S matches R and 1 otherwise. The constant alpha is the learning rate.

The training can be performed in different ways. Normally, both matrices co-learn but it is also possible to fix the weights of one matrix and learn the other. The initial values for both matrices can be assigned to chroma mappings or assigned random values. In addition to this, different learning rates for the matrices can be set.

One advantage of using a linear projection (multiplication by **L** and **M**) to obtain paired feature vectors is that the matrix can be visualized to give some insight as to what features are being used by the system. For example, if the chromagram representation were optimal, we would expect **L** and **M** to maintain their projections shown in Figure 1 during training. In contrast, Figure 2 shows the actual result of learning matrix **M** starting from a chroma mapping. In this particular case, the learned weights are systematically different.

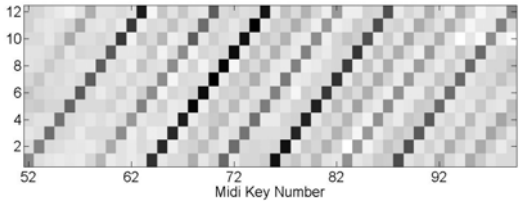

Figure 2. The trained matrix M with initial chroma pattern and fixed L with chroma projection (as in Figure 1).

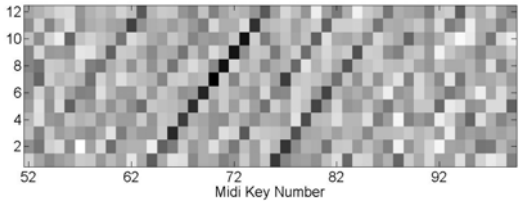

Figure 3. The trained M matrix with random initial values and initial chroma pattern for L.

In comparison to the preceding two figures, Figure 3 shows a trained **M** where initial weights were random. The **L** matrix had initial values for the chroma projection and was allowed to co-learn with **M**. The matrix in Figure 3 is similar to the chromagram in that each row

corresponds to the detection of a different pitch class or chroma. In at least some of the rows, there is a clear pattern of high weights separated by octaves.

The matrix in Figure 2 differs from the chroma mapping in several ways. First, the matrix is not symmetric, but this would be expected from the asymmetry of the training data and the nature of the training algorithm. Second, the rows are not just selecting octaves and pitch classes. It has been noted that the chromagram does not really compute the strength of 12 pitch classes because harmonics of the fundamental will generally include energy in bins that are mapped to other pitch classes. Here, we see that learned rows are selecting not only octave-related frequencies, but some fifths, thirds and other relationships. In fact, the rows are quite similar, at least for the octaves, fifths and major thirds, to pitch histograms for diatonic scales and the Krumhansl template [10] for key finding. This relationship has been studied in [16]. In this study, the empirical profiles have been found to present statistically significant correlations with tonal profiles obtained from human judgments. They demonstrate this by extracting tonal profiles based on covariance analysis of chroma features computed from western tonal musical recordings. Similarly, in our case, we find that the rows contain effects of both pitch distributions and overtone strength distributions. It seems likely that all of these factors play a role in determining the optimal patterns. A third property we can observe in the learned matrix is that low and high frequencies seem to have less significance. There is more variation between rows in the middle frequencies. In this work, the note range for the midi semigram was chosen to be from E_1 to $D\#_7$ and the range for the audio semigram was chosen to be E_3 to $D\#_7$. The audio semigram has a shorter note span than the midi semigram because of the time-frequency trade-off for the given time window length, which is kept short in the interest of higher time resolution.

We have learned the matrices many times using different training data and different initial conditions. Ideally, the matrices would converge to a configuration where the 12 rows represent 12 unique transpositions of some underlying pattern. To test this, we can rotate each row left and right until the correlation with a commonly used pitch distribution, such as the Krumhansl template, is maximized. For this, the pitch-class Krumhansl template is unwrapped to span multiple octaves and is weighted by a Hann window. The choice of the type of pitch distribution is not critical because the purpose of the window is only to shift the elements in a row to line up with the other rows. Figure 4 shows the aligned matrix **M**, the averages over rows of **M** and the weighted Krumhansl template used in the alignment. We observe that there is usually a unique shift (modulo 12) for each row of a pattern that is somewhat similar between rows of both matrices. However, there are also some irregularities possibly due to registral pitch effects and co-learning dynamics of the matrices.

It is reasonable to assume that there is some underlying "ideal" pattern that is learned in 12 different transpositions. Next, we test this assumption by forcing all 12 rows to contain the same basic pattern, shifted by 12 different offsets. First, we average the aligned matrix over all rows to find the estimated "ideal" pattern as shown in the middle plot of Figure 4. We then form a new matrix by copying the "ideal" pattern into every row and then un-rotating the rows according to the rotations performed to obtain the aligned matrix. The effect is to force the matrix to a more symmetric configuration and perhaps eliminate any overfitting due to the many degrees of freedom offered by an unconstrained matrix. Figure 5 shows the resulting un-rotated matrix. We can then re-evaluate the test data with the new matrices and compare the performance to the trained versions. This shows us how well the single pattern captures the essential information. The evaluations have been carried out with this process applied to **L** and **M** separately.

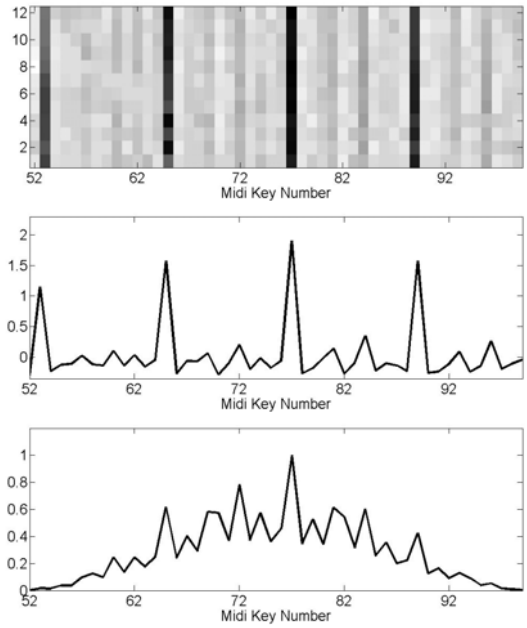

Figure 4. Matrix M aligned (upper plot). Average over rows of the aligned matrix (middle plot) and the weighted multi-octave Krumhansl template (lower plot). The sub-peaks in the middle plot represent major 3rds and perfect 5ths or perhaps 5th and 3rd harmonics.

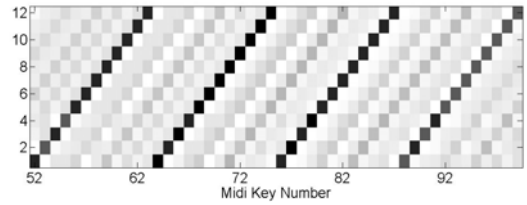

Figure 5. Un-rotated M after averaging over rows of the aligned matrix.

Learning can alternatively be started from random weights in both matrices. In this case, comparable classification accuracy is achieved, however, neither

matrix exhibits an easily visualizable structure similar to those seen in the preceding figures.

6. EVALUATION

Several different evaluations have been carried out on the data set. The first evaluation used a group consisting of 4 pieces (TAR, NOC, DIS, RON) for training and the remaining three pieces (LEG, HAT, THU) for testing. For training, eleven unique transpositions were added to the original aligned MIDI semigram - audio semigram pairs. The aligned pairs were followed by the same length of random pairs. For testing, four transpositions of the test pieces and a fresh set of random pairs were added to the aligned test set to assess its generalization capability. The classification accuracy of this test is given in the top row of Table 2 and is abbreviated TNDR. The table is divided into two groups of 3 columns with the first group showing the accuracy of the model run on the training data itself and the second group showing the accuracy for the test data. Within each group the column labeled 'LE' shows the results for the learned matrices, 'LU' for the aligned, averaged and un-rotated matrices and 'CH' for the matrices in standard chroma form (as shown in Figure 1 for **M**). A similar evaluation was performed by inter-changing the test and training sets in the evaluation mentioned above. The results are given in the second row of Table 2 with the abbreviation LHT. We also tried using Krumhansl templates (rotated to 12 transpositions) as the rows of the projection matrix, but this did not work as well as the standard chromagram. Although we omit those results here, as a summary, they performed about 3% less than the chroma mapping.

Another type of evaluation was carried out by training on each piece in Table 1 and then testing the alignment function using the remaining six pieces (hold out testing). The remaining seven rows of Table 2 show the results of this evaluation.

Table 2. Accuracy for group and hold out tests. LE: learned, LU: learned with averaging and un-rotating, CH: chroma.

	Training Data (%)			Test Data (%)		
Rec.	**LE**	**LU**	**CH**	**LE**	**LU**	**CH**
TNDR	89.6	88.7	85.4	90.5	89.6	87.4
LHT	91.3	90.5	87.3	88.9	88.5	85.3
TAR	90.0	86.8	83.4	88.2	87.8	86.2
NOC	87.6	87.0	83.5	88.9	89.1	86.5
DIS	90.8	90.6	87.4	89.0	88.8	85.3
LEG	91.9	88.0	88.0	88.1	84.3	85.6
HAT	91.4	90.0	83.9	88.9	88.6	86.1
THU	91.6	91.2	88.3	88.5	88.5	85.7
RON	91.5	89.4	85.1	88.8	88.9	86.2
CLA1	93.1	92.7	90.6	90.0	89.6	88.3
CLA2	92.0	91.4	89.2	91.1	90.8	89.1

Overall, for all the tests explained above, learned (LE) and learned averaged (LU) tests performed better than chroma (CH) with one exception in piece LEG where LU was lower than CH. This shows that averaging in this particular case did not work well and degraded the performance. In general, however, results suggest that an asymmetrical multi-octave chroma mapping is better than the commonly used octave independent symmetrical mapping as suggested by [7] and [14] and others.

CLA1 and CLA2 refer to the training data given in Table 1. Each of these were tested with the remaining 36 pieces (about 2000 seconds) from the RWC collection.

The accuracy numbers partially reflect the effects of some foreseeable factors in performing this evaluation: training alignments are not perfectly aligned at the frame level, the time variation of the spectral content in the audio is not reflected in the MIDI representation (timbre effects), audio contains percussion but the MIDI does not.

7. RESULTS

The most interesting result is that learned representations outperform chroma vectors on the task of discriminating aligned vs. unaligned audio frames. Perhaps this should not be too surprising since machine learning from large sets of training data often outperforms hand-tuned algorithms or features. Not only is there nothing "magic" about the chromagram, we see comparable performance from a neural network trained to answer the question "Does this MIDI frame align to that spectral frame?"

We also explored a particular model that maps the spectrum (and MIDI data) into 12-element vectors and computes similarity between these vectors using a standard distance function. Even though this certainly loses information, it allows us to study the representation, which can be viewed as a projection of the spectrum to a new space defined by a set of basis vectors. These vectors are particularly interesting. With chromagrams, the basis vectors are simply chroma (pitch classes), but with our learned projections, the basis vectors also show a remarkable similarity to pitch histograms obtained from music in a fixed key. Thus, even assuming the general form of the chromagram as a projection from the spectrum to a lower-dimensional space, we see room for improvement. This is evident from the fact that a learning system initialized with the chromagram projection will systematically adjust and improve to a new projection.

8. FUTURE WORK

We have limited our study to a 12-element vector representation for comparison to the chromagram. It would be interesting to study larger (and smaller) vectors. In particular, we wonder whether with additional dimensions, the learning algorithms would build different patterns for major, minor, and dominant tonalities, whether some patterns would reflect timbre or overtone characteristics, or whether other structures would be formed. The projection matrix formulation of the problem allows these potential structures to be observed. It should be noted that the nature of the **M** and **L** matrices is slightly different in that **M** incorporates the spectral structure of notes whereas **L** deals with notes alone.

The similarity of our learned patterns to the pitch histogram or Krumhansl template deserves further analysis.

Is this a coincidence? Are the learned patterns a reflection of the pitch distributions as well as average overtone strength distributions in our training data, or have pitch distributions in tonal music evolved to optimize the listener's ability to recognize music structures? Perhaps both forces are at work.

9. CONCLUSIONS

We have described methods for learning features that are useful for score alignment and other comparative and similarity based tasks such as identification of repeating sections, subsequence searching and template based chord recognition. The learned features out-perform the chromagram representation at least in the task of discriminating aligned from non-aligned frames of music. Unlike the chromagram representation, which is a simple projection based on pitch classes, the learned representation uses a projection that appears to be based on pitch distributions as well as the harmonic series common to most pitched musical instruments. In addition, the middle frequencies and pitches receive the most weight in the patterns, indicating that high and low frequencies are less useful for alignment. Another advantage of such an approach in MIR is that an alignment function can be directly learned from and used with almost native representations in both spectral and symbolic domains, thus bridging the gap between audio and symbolic music collections. We believe this work represents a significant advance by suggesting better features for music audio analysis, particularly for alignment and discovering music structure.

10. ACKNOWLEDGEMENTS

This material is based on work partly supported by the National Science Foundation under Grant Nos. 0534370 and 0855958. We would like to thank Meinard Müller for providing alignment data for the RWC dataset.

11. REFERENCES

[1] M. Bartsch and Wakefield, G. H. "Audio Thumbnailing of Popular Music Using Chroma-based Representations," *IEEE Transactions on Multimedia*, vol. 7, pp. 96-104, Feb. 2005.

[2] Dannenberg, R. and Hu, N. "Pattern Discovery Techniques for Music Audio," *Int. Symposium on Music Information Retrieval (ISMIR)*, Paris: IRCAM, pp. 63-70, 2002.

[3] Dannenberg, R. and Raphael, C. "Music Score Alignment and Computer Accompaniment," *Commun. ACM*, 49(8) (August 2006), pp. 38-43.

[4] Dixon, S. and Widmer, G. "Match: A Music Alignment Tool Chest," *Int. Symposium on Music Information Retrieval (ISMIR)*, London: Queen Mary, Univ. of London and Goldsmiths College, Univ. of London, 2005.

[5] Ewert, S., Müller, M., and Grosche, P. "High Resolution Audio Synchronization Using Chroma Onset Features," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), Taipei, Taiwan, pp. 1869-1872, 2009.

[6] Fujishima, T. "Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music," *Proc. of the 1999 Int. Computer Music Conference (ICMC), pp. 464-467*, 1999.

[7] Gómez, E., "Tonal Description of Music and Audio Signals," Ph.D. dissertation, Barcelona: MTG, Universitat Pompeu Fabra, 2006.

[8] Goto, M. "A Chorus-Section Detection Method for Musical Audio Signals and Its Application to a Music Listening Station," *IEEE Trans. On Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1783-1794, Sep. 2006.

[9] Hu, N., Dannenberg, R., and Tzanetakis, G. "Polyphonic Audio Matching and Alignment for Music Retrieval," *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (WASPAA), New Paltz, USA, pp. 185-188, 2003.

[10] Krumhansl, C. *Cognitive Foundations of Musical Pitch.* New York: Oxford Univ. Press, 1990.

[11] Lu, L., Wang, M., and Zhang, H.-J. "Repeating Pattern Discovery and Structure Analysis from Acoustic Music Data." *Proc. of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval.* New York: Assoc. for Computing Machinery, pp. 275-282, 2004.

[12] Müller, M., Ewert, S., and Kreuzer, S. "Making Chroma Features more Robust to Timbre Changes," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), Taipei, Taiwan, pp. 1869-1872, 2009.

[13] Müller, M., Kurth, F., and Clausen, M. "Audio Matching via Chroma-Based Statistical Features," *Int. Symposium on Music Information Retrieval (ISMIR)*, pp. 144-149, Oct. 2006.

[14] Pauws, S. "Musical Key Extraction from Audio," *Int. Symposium on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004.

[15] Serrà, J., Gómez, E., Herrera, P. and Serra, X. "Chroma Binary Similarity and Local Alignment Applied to Cover Song Identification," *IEEE Transactions on Audio, Speech and Language Processing*, 16-6, pp. 1138-1152, August 2008.

[16] Serrà, J., Gómez, E., Herrera, P. and Serra, X. "Statistical Analysis of Chroma Features in Western Music Predicts Human Judgments of Tonality," *Journal of New Music Research*, 37-4, pp. 299-309, December 2008.