

A WAVELET-BASED APPROACH TO THE DISCOVERY OF THEMES AND SECTIONS IN MONOPHONIC MELODIES

Gissel Velarde

Aalborg University
gv@create.aau.dk

David Meredith

Aalborg University
dave@create.aau.dk

ABSTRACT

We present the computational method submitted to the MIREX 2014 Discovery of Repeated Themes & Sections task, and the results on the monophonic version of the JKU Patterns Development Database. In the context of pattern discovery in monophonic music, the idea behind our method is that, with a good melodic structure in terms of segments, it should be possible to gather similar segments into clusters and rank their salience within the piece. We present an approach to this problem and how we address it. In general terms, we represent melodies either as raw 1D pitch signals or as these signals filtered with the continuous wavelet transform (CWT) using the Haar wavelet. We then segment the signal either into constant duration segments or at the resulting coefficients' modulus local maxima. Segments are concatenated based on their contiguous city-block distance. The concatenated segments are compared using city-block distance and clustered using an agglomerative hierarchical cluster tree. Finally, clusters are ranked according to the sum of the length of segments' occurrences. We present the results of our method on the JKU Patterns Development Database.

1. INTRODUCTION

We present the computational method¹ submitted to the MIREX 2014 Discovery of Repeated Themes & Sections task, and the results on the monophonic version of the JKU Patterns Development Database². In the context of pattern discovery of monophonic pieces, the idea behind our method is that, with a good melodic structure in terms of segments, it should be possible to gather together similar segments to rank their salience within the piece (See 'paradigmatic analysis' [3]). We also consider other aspects of the problem, in particular, representation, segmentation, measuring similarity, clustering of segments and ranking segments according to salience.

This document is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 License.
<http://creativecommons.org/licenses/by-nc-sa/3.0/>
© 2014 The Authors

¹The algorithm is implemented in MATLAB (R2013b, The Mathworks, Inc), using the following toolboxes: Signal Processing, Statistics, Symbolic Math, Wavelet, and the MIDI Toolbox (Eerola & Toiviainen, 2004).

²<https://dl.dropbox.com/u/11997856/JKU/JKUPDD-noAudio-Aug2013.zip>. Accessed 12 May 2014

In the context of this MIREX task, a good melodic structure is considered to be one that is closer to the ground truth analysis, which specifies certain patterns identified by expert analysts as being important or noticeable. These patterns may be nested or hierarchically related (see [1]). We use an agglomerative technique to cluster segments by similarity. Clusters are then ranked according to a perceptually motivated criterion.

2. METHOD

The method follows and extends our previously reported approach to melodic segmentation and classification based on filtering with the Haar wavelet [4] and uses some ideas from a generic motif discovery algorithm for sequential data [2]. It follows [4] in terms of representation and segmentation, extending the segmentation method. As [2] is very generic, we use the idea of computing a similarity matrix for "window connectivity information" as described in section 2.2.4.

2.1 Representation

As in [4], we represent melodies either as raw 1D pitch signals or as these signals filtered with the continuous wavelet transform (CWT) using the Haar wavelet at a single time scale. The melodic contour of a melody is sampled using chromatic MIDI pitch information at a defined sampling rate. In the case of pitch signal representation, after segmentation, melodic segments are normalized by subtracting the average pitch.

2.2 Segmentation

2.2.1 First stage segmentation

We use some of the segmentation methods described in [4] and additionally use modulus maxima segmentation. The segmentation methods are:

- constant segmentation, i.e., segmentation into segments of constant length, or
- modulus maxima, where segmentation points are set at local modulus maxima of the wavelet coefficients.

2.2.2 Segment length normalization

The segments obtained using these methods generally have different lengths. In order to normalize their length for the purpose of measuring their city-block distances,

and therefore have segments of equal length we define a maximal length for all segments and pad shorter segments as necessary with zeros at the end.

2.2.3 Comparison

Segments are compared by building a distance matrix giving all pair-wise distances between segments in terms of normalized city-block distance. The normalization consists of dividing the pairwise distance by the length of the smallest segment before segment length normalization by zero padding.

2.2.4 Concatenation of segments

We binarize the distance matrix setting a threshold: values lower than or equal to the threshold take the value of 1 or true, otherwise the value is 0 or false. We concatenate segments of contiguous true values of the diagonals, to form longer segments.

2.3 Comparison

This time we use the segments that have been concatenated as described in 2.2.4. The comparison is the same as in 2.2.3.

2.4 Clustering

The distance matrix obtained in 2.3 is used for clustering. We use agglomerative clusters from an agglomerative hierarchical cluster tree. Finally, clusters are ranked according to the sum of the length of segments' occurrences.

3. EXPERIMENTS

We tested the following parameter combinations:

- Melodies sampled at 16 samples per quarter note (qn)
- Representation: normalized pitch signal or wavelet coefficients filtered at the scale of 1 qn
- Segmentation: constant segmentation or modulus maxima
- Scale segmentation at 1 or 4 qn
- Threshold for concatenating segments: 0.1 or 1
- Distance for both comparisons: city-block
- Number of clusters: 7
- Ranking criterion: Sum of the length of occurrences

4. RESULTS

We used the evaluation metrics defined by Collins and Meredith in [1] and Collins' Matlab implementation to compute the results. The results are obtained applying our method on the JKU Patterns Development Database

monophonic version, which contains five melodies for training: Bach's Fugue BWV 889, Beethoven's Sonata Op. 2, No. 1, Movement 3, Chopin's Mazurka Op. 24, No. 4, Gibbons's Silver Swan, and Mozart's Sonata K.282, Movement 2. Table 1 and Table 2 present the results of our two submissions VM1 and VM2 respectively. In our experiments we have tested all combinations mentioned in section 3, and selected two configurations to submit to MIREX. VM1 differs from VM2 in the following parameters:

- Normalized pitch signal representation,
- Constant segmentation at the scale of 1 qn,
- Threshold for concatenation 0.1.

VM2 differs from VM1 in the following parameters:

- Wavelet coefficients representation filtered at the scale of 1 qn
- Modulus maxima segmentation at the scale of 4 qn
- Threshold for concatenation 1

According to Friedman's test ($\chi^2(1)=1.8$, $p=0.1797$) VM1 and VM2 show no significant difference in the results of the "three-layer" F1 score. However, for discovering exact occurrences, VM1 outperforms VM2, ($\chi^2(1)=4$, $p=0.045$). On the other hand, there is a statistically significant difference in the runtime, suggesting that VM2 should be preferable for fast computation, ($\chi^2(1)=5$, $p=0.0253$).

In general, recall values are slightly higher than precision values, and the standard deviation of the recall values are slightly lower than the standard deviation of the precision values. For standard precision, recall and F1 score, the standard deviation is highest, compared to the standard deviation of establishment and occurrence measures. These results suggest that VM1 and VM2 perform consistent on the training dataset over establishment and occurrence values, and VM1 performs less consistent on the standard measures.

5. CONCLUSIONS

We present a novel computational method for the discovery of repeated themes and sections in monophonic melodies and the results of our two submissions on the same task, considering that VM1 and VM2 perform similarly on the "three-layer" measures, but VM1 should be preferable for standard measures and VM2 should be preferable for runtime computation.

6. ACKNOWLEDGEMENT

Gissel Velarde is supported by the Department of Architecture, Design and Media Technology at Aalborg University. The contribution of David Meredith to the work reported here was made as part of the "Learning to Create" project (Lrn2Cre8) which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

7. REFERENCES

- [1] T. Collins. Mirex 2014 competition: Discovery of repeated themes and sections, 2014. http://www.music-ir.org/mirex/wiki/2014:Discovery_of_Repeated_Themes_and_Sections. Accessed on 12 May 2014.
- [2] K. Jensen, M. Styczynski, I. Rigoutsos and G. Stephanopoulos: “A generic motif discovery algorithm for sequential data”, *Bioinformatics*, 22:1, pp. 21-28, 2006.
- [3] R. Monelle: *Linguistics and Semiotics in Music*, Harwood Academic Publishers, Chur, 1992.
- [4] G. Velarde, T. Weyde and D. Meredith: “An approach to melodic segmentation and classification based on filtering with the Haar-wavelet”, *Journal of New Music Research*, 42:4, 325-345, 2013.

Piece	n_P	n_Q	P_est	R_est	F1_est	P_occ	R_occ	F1_occ	P_3	R_3	F1_3	Runtime	FFTP_	FFP	P_occ	R_occ	F1_occ	P	R	F1
						(c=.75)	(c=.75)	(c=.75)				(s)	est		(c=.5)	(c=.5)	(c=.5)			
Bach	3	7	0.87	0.95	0.91	0.63	0.72	0.67	0.51	0.65	0.57	8.50	0.95	0.60	0.63	0.72	0.67	0.14	0.33	0.20
Beethoven	7	7	0.92	0.92	0.92	0.98	0.98	0.98	0.86	0.91	0.88	31.00	0.76	0.80	0.89	0.93	0.91	0.57	0.57	0.57
Chopin	4	7	0.53	0.86	0.66	0.66	0.86	0.75	0.48	0.70	0.57	34.20	0.68	0.47	0.46	0.83	0.60	0.00	0.00	0.00
Gibbons	8	7	0.95	0.95	0.95	0.66	0.93	0.77	0.85	0.79	0.82	17.76	0.77	0.79	0.66	0.93	0.77	0.29	0.25	0.27
Mozart	9	7	0.92	0.79	0.85	0.82	0.96	0.88	0.79	0.69	0.73	23.61	0.67	0.73	0.72	0.92	0.81	0.57	0.44	0.50
mean	6.2	7	0.84	0.89	0.86	0.75	0.89	0.81	0.70	0.75	0.71	23.01	0.77	0.68	0.67	0.87	0.75	0.31	0.32	0.31
SD	2.59	0	0.17	0.07	0.12	0.15	0.11	0.12	0.19	0.10	0.14	10.34	0.11	0.14	0.15	0.09	0.12	0.26	0.22	0.23

Table 1. Results of VM1 on the JKU Patterns Development Database.

Piece	n_P	n_Q	P_est	R_est	F1_est	P_occ	R_occ	F1_occ	P_3	R_3	F1_3	Runtime	FFTP_	FFP	P_occ	R_occ	F1_occ	P	R	F1
						(c=.75)	(c=.75)	(c=.75)				(s)	est		(c=.5)	(c=.5)	(c=.5)			
Bach	3	7	0.56	0.65	0.60	0.89	0.43	0.58	0.39	0.41	0.40	5.07	0.59	0.37	0.56	0.46	0.50	0.00	0.00	0.00
Beethoven	7	7	0.90	0.90	0.90	0.79	0.89	0.84	0.82	0.86	0.84	5.54	0.67	0.75	0.83	0.90	0.86	0.00	0.00	0.00
Chopin	4	7	0.58	0.86	0.69	0.69	0.83	0.75	0.53	0.78	0.64	5.83	0.65	0.44	0.67	0.65	0.66	0.00	0.00	0.00
Gibbons	8	7	0.92	0.88	0.90	0.79	0.84	0.82	0.81	0.73	0.77	2.22	0.70	0.76	0.72	0.69	0.71	0.14	0.13	0.13
Mozart	9	7	0.83	0.71	0.77	0.93	0.93	0.93	0.77	0.63	0.69	5.70	0.56	0.68	0.84	0.88	0.86	0.00	0.00	0.00
mean	6.2	7	0.76	0.80	0.77	0.82	0.78	0.78	0.66	0.68	0.67	4.87	0.63	0.60	0.72	0.71	0.72	0.03	0.03	0.03
SD	2.59	0	0.17	0.11	0.13	0.09	0.20	0.13	0.19	0.17	0.17	1.51	0.06	0.18	0.12	0.18	0.15	0.06	0.06	0.06

Table 2. Results of VM2 on the JKU Patterns Development Database.