

EXPLOITING MELODIC SMOOTHNESS FOR MELODY DETECTION IN POLYPHONIC AUDIO

Rui Pedro Paiva

Teresa Mendes

Amílcar Cardoso

CISUC – Centre for Informatics and Systems of the University of Coimbra
Department of Informatics Engineering, University of Coimbra
e-mail: {ruipedro, tmendes, amilcar}@dei.uc.pt

ABSTRACT

This paper describes a method for melody detection in polyphonic musical signals. Our approach starts by obtaining a set of pitch candidates for each time frame, with recourse to an auditory model. Trajectories of the most salient pitches are then constructed. Next, note candidates are obtained by trajectory segmentation (in terms of frequency and pitch salience variations). Too short, low-salience and harmonically-related notes are then eliminated. Finally, the notes comprising the melody are extracted. Comparing to our previous work, we extend it by making use of melodic smoothness for the definition of the final melody notes. We tested our method with excerpts from 21 songs encompassing several genres and obtained an average detection accuracy of 82%. Melody smoothing was responsible for an improvement of 11.8% in the overall accuracy.

1. INTRODUCTION

Query-by-humming (QBH) is a particularly intuitive way of searching for a musical piece, since melody humming is a natural habit of humans. This is an important research topic in an emergent and promising field called Music Information Retrieval (MIR). Several techniques have been proposed in order to attain that goal in the MIDI domain, e.g., [1]. However, querying “real-world” polyphonic recorded musical pieces requires the analysis of polyphonic musical waveforms. This is a rather complex task since many types of instruments can be playing at the same time, with severe spectral interference between each other.

Previous work concerned with obtaining symbolic representations from musical audio has concentrated especially on the problem of full music transcription, which requires accurate multi-pitch estimation for the extraction of all fundamental frequencies present in a song under analysis, e.g., [5]. However, the present solutions are neither sufficiently general nor accurate, often imposing several constraints on the music material.

Only little work has been carried out in the particular problem of melody detection in “real-world” songs, e.g., [3, 4, 6, 8]. Additionally, most of the work is only concerned with the extraction of melodic pitch lines, rather than melody notes.

In our approach we put the focus on the melody, no matter what other sources are present. Thus, we base our strategy in two main assumptions that we designate as the “salience principle” and the “melodic smoothness principle”. By the salience principle, we assume that the

notes comprising the melody are, in general, salient in the mixture. As for the melodic smoothness principle, we exploit the fact that note frequency intervals tend, generally, to be small.

2. MELODY DETECTION SYSTEM

Our melody detection algorithm comprises five stages, as illustrated in Figure 1. The general strategy was described previously, e.g., [8]. New improvements to the melody extraction stage are described in more detail.

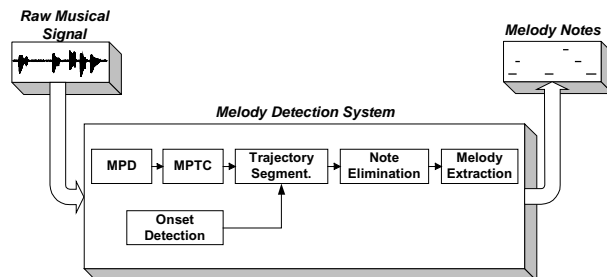


Figure 1. Melody detection system overview.

In the Multi-Pitch Detection (MPD) stage, the objective is to capture a set of pitch candidates, which constitute the basis of possible future notes. We perform pitch detection in a frame-based analysis, defining a 46.44 ms frame length and a hop size of 5.8 ms, based on an auditory model proposed by Slaney and Lyon [9]. For each obtained pitch, a pitch salience is computed, which is approximately equal to the energy of the corresponding fundamental frequency.

Multi-Pitch Trajectory Construction (MPTC), in the second stage, aims to create a set of pitch tracks, formed by connecting consecutive pitch candidates with similar frequency values. To this end, we based ourselves on the algorithm proposed by Serra [10].

Each trajectory from the MPTC stage may contain more than one note and, therefore, segmentation of tracks must be conducted in the third stage. This is carried out in two phases: frequency segmentation, aiming to separate notes with different MIDI values, and salience segmentation with the objective of dividing consecutive notes at the same MIDI note number.

In the fourth stage, irrelevant note candidates are eliminated, based on their saliences, durations and on the analysis of harmonic relations. We make use of perceptual rules of sound organization, namely “harmonicity” and “common fate” [2], where common frequency and amplitude modulation are exploited.

In the last stage, our goal is to obtain a final set of notes comprising the melody of the song under analysis. In fact, although a significant amount of irrelevant notes is eliminated in the previous stage, there are still many notes present. Therefore, we have to determine which are the ones that convey the main melodic line. This is the core topic of this paper and is described in the following section.

3. EXTRACTION OF MELODY NOTES

The definition of the notes comprising the melody of a song under analysis, being probably the most important task of any melody detection algorithm, is also the most difficult one to carry out. In fact, many aspects of auditory organization influence the perception of melody by humans, for instance in terms of the role played by pitch, timbre and intensity content of the sound signal.

In our approach, we do not tackle the problem of source separation. Instead, we base our strategy on the assumptions that i) the main melodic line often stands out in the mixture (salience principle) and that ii) melodies are usually smooth in terms of the note frequency intervals, which tend to be small (melodic smoothness principle).

3.1. Selecting the Most Salient Notes

In the first step of the melody extraction stage, we select the most salient notes at each time as initial melody candidates. The criteria used for comparing the salience between notes as well as algorithmic details were described previously, e.g., [8]. In the implemented algorithm, some of the selected notes were truncated, since melody notes are not allowed to overlap in time.

The results of melody extraction after selecting the most salient notes are illustrated in Figure 3, for an excerpt from Pachelbel’s Kanon. There, the correct notes are depicted in gray and the black continuous lines denote the obtained melody notes. The dashed lines stand for the notes that result from the note elimination stage. We can see that some erroneous notes are extracted, whereas true melody notes are excluded. Namely, some octave errors occur.

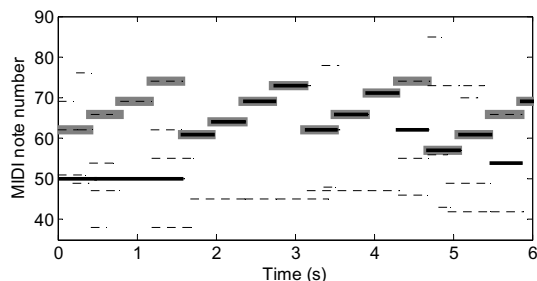


Figure 3. Extraction of the most salient notes for an excerpt from “Pachelbel’s Kanon”.

In fact, one of the limitations of only taking into consideration note saliences is that the notes comprising the

melody are not always the most salient ones. In this situation, wrong notes may be selected as belonging to the melody, whereas true notes are left out. This is particularly clear when abrupt transitions between notes are found, as can be seen in Figure 3. Hence, we improved our method by smoothing the melody contour.

3.2. Exploiting Melodic Smoothness

As referred above, abrupt transitions between notes give strong evidence that wrong notes were selected. In fact, small frequency transitions favor melody coherence, since smaller steps in pitch hang together better [2]. In an attempt to demonstrate that musicians generally prefer to use smaller note steps, the psychologist Otto Ortmann counted the number of sequential intervals in several songs by classical composers, having found that the smallest ones occur more frequently and that their respective number roughly decreases in inverse proportion to the size of the interval [2]. So being, we improved the melody extraction stage by taking advantage of this melodic smoothness principle.

We started to improve the initial melody by performing octave correction. In fact, in the note elimination stage not all harmonically-related notes are eliminated and, thus, some octave errors occur when sub or superharmonic notes are more salient than the right notes. In order to correct octave errors, we select all notes for which no octaves (either above or below) are found and compute their average MIDI values. Then, we analyze all notes that have octaves with common onsets: if the octave is closer to the computed average, the original note is replaced by the corresponding octave. This simple first step already improves the final melody significantly. However, some octave errors, as well as abrupt transitions, are still kept, which will be worked out in the following stages.

In the second step, we analyze the obtained notes and look for regions of smoothness, i.e., regions where there are no abrupt transitions between consecutive notes. Here, we define a transition as being abrupt if the intervals between consecutive notes are above a fifth, i.e., seven semitones, as illustrated in Figure 4. There, the bold notes (a_1 , a_2 and a_3) are marked as abrupt. In the same example, four initial regions of smoothness are detected (R_1 , R_2 , R_3 and R_4).

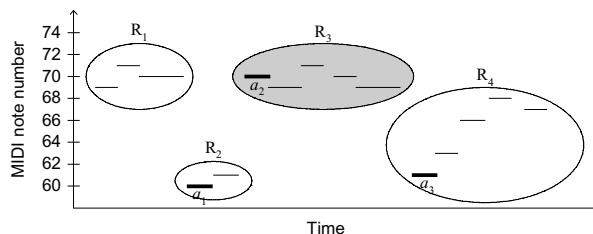


Figure 4. Regions of smoothness.

Then, we select the longest region as a correct region (region R_3 , in Figure 4, filled in gray) and define the allowed note range for its adjacent regions (R_2 and R_4).

Regarding the left region, we define its allowed range based on the first note of the correct region, e.g., MIDI value 70 in this example. Keeping in mind the importance of the perfect fifth, the allowed range for the left region is 70 ± 7 , i.e., [63, 77]. As region R_2 contains no note in the allowed range, this region is a candidate for elimination. However, before deletion, we first check if each of its notes contains an octave in the allowed range. If so, the corresponding notes are substituted by the found octaves. If at least one octave is found, no note is deleted in this iteration. On the contrary, if no octave is found, all the notes are eliminated.

For the right region we proceed likewise. Hence, we define the allowed range based on the last note of the correct region, e.g., 69 in this example, resulting the range [62, 76]. Since region R_4 contains notes in the allowed range, its first note, i.e., note a_3 , is marked as non-abrupt. However, we still look for an octave of the referred note in the allowed range. In case it is found, the abrupt note is substituted, as before.

In short words, regions that correspond to sudden movements to different registers are interpreted as being incoherent and are, consequently, eliminated. However, abrupt transitions are allowed if adjacent regions are both coherent in melodic terms, as happens in Figure 4 for regions R_3 and R_4 . This situation occurs in some musical pieces as, for example, Pachelbel’s Kanon, as can be seen in Figures 3 and 5.

If no notes are substituted/deleted for the current region, the following regions are analyzed in the same way, in descending length order. If no change at all is performed for all regions, the algorithm stops. Otherwise, whenever a change is performed, the procedure for definition of regions of smoothness, analysis of neighbors and deletion/substitution is repeated until no change is done. In the successive iterations, regions of smoothness are defined taking into consideration notes previously marked as non-abrupt, e.g., the notes in region R_4 in the above descriptions. Thus, in a following iteration, regions R_3 and R_4 will not be divided.

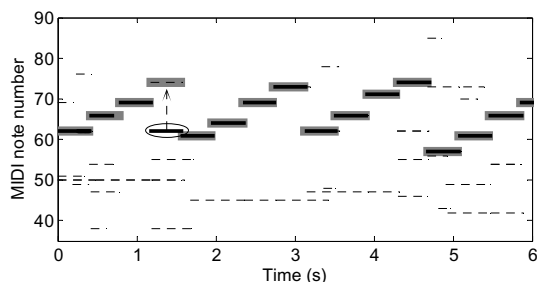


Figure 5. Extracted melody for an excerpt from “Pachelbel’s Kanon”.

Finally, since some regions are eliminated, their notes need to be substituted by other notes that are more likely to belong to the melody, according to the smoothness principle. Thus, we fill each gap in the melody with the most salient note candidates that are in the allowed range for that region.

The results of the implemented procedures are illustrated in Figure 5, for the same excerpt from Pachelbel’s Kanon. We can see that only one erroneous note resulted (signaled by an ellipse), which corresponds to an octave error. This example is particularly challenging to our melody-smoothing algorithm due to the periodic abrupt transitions present. Yet, the performance was very good.

4. EXPERIMENTAL RESULTS

We evaluated the proposed algorithms with a database generated for the Melody Extraction Contest (MEC-04) as part of the ISMIR 2004 Audio Description Contest [7] and a test-bed we had previously created [8].

For accuracy computation, the detected melody notes were compared with the correct notes. Then, we used the pitched accuracy metric defined in [7], with some adaptations. Namely, the target frequency values for each time frame were defined as the reference frequencies of the corresponding MIDI values. In the same way, the extracted frequencies were defined from the reference frequencies corresponding to the extracted melody notes. The accuracy was calculated as the percentage of correctly identified frames. In the original metric defined in [7], exact frequency values were used, which seem more relevant in a problem of predominant-pitch detection, rather than in our melody detection problem.

Four evaluations were performed: i) using only note saliences, without the allowance of octave errors (Sa); ii) only note saliences, permitting octave errors ($Sa+Oc$); iii) note saliences and melodic smoothness (Sm); and iv) note saliences and melodic smoothness, with the allowance of octave errors ($Sm+Oc$). From these, we are naturally more interested in the Sm evaluation.

The obtained results are summarized in Table 1, where the top 11 lines correspond to our test-bed and the next 10 refer to the MEC-04 database.

We can see that good results were achieved for the Sm evaluation. There, an average accuracy of 82% was attained. Also, in several excerpts the system achieved almost 100% accuracy. Without melody smoothing, the average accuracy was 70.2% (Sa evaluation) and so our implementation of the melodic smoothness principle amounts for an average improvement of 11.8%.

As for the MEC-04 database, the results were also good, except for the opera excerpts. These samples seem to pose additional difficulties to the pitch detection algorithm, in the first stage of our system. We plan to address this issue in the near future.

Another interesting fact is that the proposed approach is almost immune to octave errors, as can be seen by comparing the Sm and $Sm+Oc$ columns in Table 1: their average accuracy differs by only 1.1%.

5. CONCLUSIONS

We propose a system for melody detection in polyphonic musical signals. This is a main issue for MIR applications, such as QBH in “real-world” music data-

<i>Song Title</i>	<i>Genre</i>	<i>Sa</i>	<i>Sa+Oc</i>	<i>Sm</i>	<i>Sm+Oc</i>
Pachelbel's Kanon	Classical	59.3	85.1	89.5	96.0
Handel's Hallelujah	Choral	60.1	68.1	81.4	83.2
Enya - Only Time	Neo-Classical	82.7	82.7	89.4	89.4
Dido - Thank You	Pop	94.3	94.3	94.3	94.3
Ricky Martin - Private Emotion	Pop	69.0	79.8	80.4	80.4
Avril Lavigne - Complicated	Pop / Rock	58.2	77.7	92.8	92.8
Claudio Roditi - Rua Dona Margarida	Jazz / Easy	89.0	98.3	98.3	98.3
Mambo Kings - Bella Maria de Mi Alma	Bolero	88.2	88.2	91.3	91.3
Compay Segundo - Chan Chan	Son Cubano	70.6	78.5	70.6	78.5
Juan Luis Guerra - Palomita Blanca	Bachata	71.5	71.5	80.2	80.2
Battlefield Band - Snow on the Hills	Scottish Folk	46.4	85.8	94.6	94.6
daisy2	Synthesized singing voice	85.1	86.6	88.1	88.1
daisy3	Synthesized singing voice	66.4	71.1	78.3	78.3
jazz2	Saxophone phrases	62.0	68.1	70.6	70.6
jazz3	Saxophone phrases	75.8	79.1	86.1	86.1
midi1	MIDI synthesized	69.8	86.0	80.9	87.8
midi2	MIDI synthesized	97.6	97.6	97.6	97.6
opera_fem2	Opera singing	47.9	59.0	64.5	64.5
opera_male3	Opera singing	40.5	46.7	45.1	45.1
pop1	Pop singing	63.7	64.4	70.3	70.3
pop4	Pop singing	75.7	77.0	77.6	77.6
<i>Average accuracy</i>		70.2	78.4	82.0	83.1

Table 1. Results of the melody detection system.

bases. The work conducted in this field is presently restricted to the MIDI domain, and so we guess we make an interesting contribution to the area, with some encouraging results. Furthermore, we explicitly define musical notes, with precise onsets and offsets, something that is not addressed in most approaches for melody detection. Regarding future work, we plan to further work out some of the described limitations, as well as addressing the problem of false positive notes.

6. ACKNOWLEDGEMENTS

This work was partially supported by the Portuguese Ministry of Science and Technology, under the program PRAXIS XXI.

7. REFERENCES

- [1] Bainbridge, D., Nevill-Manning, C., Witten, I., Smith, L. and McNab, R. "Towards a digital library of popular music", *Proceedings of the ACM International Conference on Digital Libraries*, 1999.
- [2] Bregman, A. S. *Auditory scene analysis: the perceptual organization of sound*, MIT Press, 1990.
- [3] Eggink, J., and Brown, G. J. "Extracting melody lines from complex audio", *Proceedings of the International Conference on Music Information Retrieval*, 2004.
- [4] Goto, M. "A predominant-F0 estimation method for CD recordings: MAP estimation using EM algorithm for adaptive tone models", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001.
- [5] Klapuri, A. *Signal processing methods for the automatic transcription of music*. PhD Thesis, Tampere University of Technology, 2004.
- [6] Marolt, M. "On finding melodic lines in audio recordings", *Proceedings of the International Conference on Digital Audio Effects*, 2004.
- [7] MTG - UPF. "ISMIR 2004 Audio Description Contest", *International Conference on Music Information Retrieval*, 2004. http://ismir2004.ismir.net/ISMIR_Contest.html
- [8] Paiva, R. P., Mendes, T., and Cardoso, A. "An auditory model based approach for melody detection in polyphonic musical recordings", In Wiil, U. K. (ed.), *Computer Music Modelling and Retrieval - CMMR 2004*, Lecture Notes in Computer Science, Vol. 3310, 2005.
- [9] Slaney, M., and Lyon, R. F. "On the importance of time - a temporal representation of sound", In Cooke, Beet and Crawford (eds.), *Visual representations of speech signals*, 1993.
- [10] Serra, X. "Musical sound modeling with sinusoids plus noise". In Roads, C., Pope, S., Piccilli, A., and De Poli, G. (eds.), *Musical signal processing*, 1998.