

# AN ALGORITHM FOR MELODY DETECTION IN POLYPHONIC RECORDINGS

Rui Pedro Paiva

CISUC – Centre for Informatics and Systems of the University of Coimbra  
Department of Informatics Engineering, Pólo II – Pinhal de Marrocos  
P 3030 – 290 Coimbra, Portugal  
ruipedro@dei.uc.pt

## ABSTRACT

This paper describes an algorithm for melody detection in polyphonic recordings. 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, we extract the notes comprising the melody by selecting the most salient ones at each moment, exploiting melodic smoothness and removing spurious notes that correspond to abrupt drops in note saliences or durations.

**Keywords:** Melody extraction, auditory modelling, multi-pitch detection, trajectory creation and segmentation, perceptual rules of sound organization, melodic smoothness.

## 1 INTRODUCTION

This paper outlines an algorithm for melody detection in polyphonic audio signals. The proposed system comprises five stages, as illustrated in Figure 1. Different parts of the system were described in greater detail detailed in other publications, e.g., [1, 2, 3, 4].

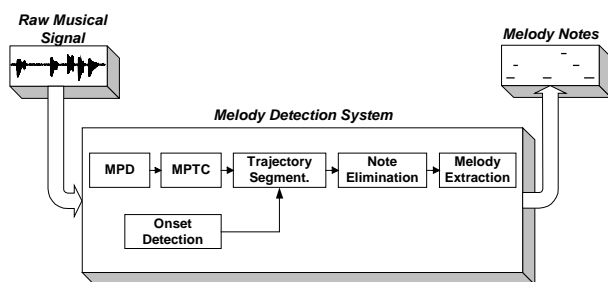


Figure 1. Melody detection system overview.

In the Multi-Pitch Detection (MPD) stage, the objective is to capture the most salient pitch candidates, which constitute the basis of possible future notes.

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.

Since the tracks resulting from the MPTC stage may contain more than one single note, they have to be segmented. This segmentation is performed 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. Onset detection is carried out to support this task.

In the fourth stage, irrelevant note candidates are eliminated, based on their saliences, durations and on the analysis of harmonic relations.

In the last stage, our goal is to obtain the final set of notes comprising the melody of the song under analysis. In fact, although a significant amount of irrelevant notes are eliminated in the previous stage, many notes are still present.

Each of the presented modules will be briefly described in the following sections.

## 2 MULTI-PITCH DETECTION (MPD)

In the first stage of the algorithm, Multi-Pitch Detection (MPD) is conducted, with the objective of capturing a set of candidate pitches that constitute the basis of possible future musical notes.

Pitch detection is carried out with recourse to an auditory model, in a frame-based analysis with a 46.44 ms frame length and a hop size of 5.8 ms. Our approach is based on Slaney and Lyon's auditory model [5].

This analysis comprises four stages:

i) Conversion of the sound waveform into auditory nerve responses for each frequency channel, using a model of the ear, with particular emphasis on the cochlea, obtaining a so-called cochleagram;

ii) Detection of the main periodicities in each frequency channel using auto-correlation, from which a correlogram results;

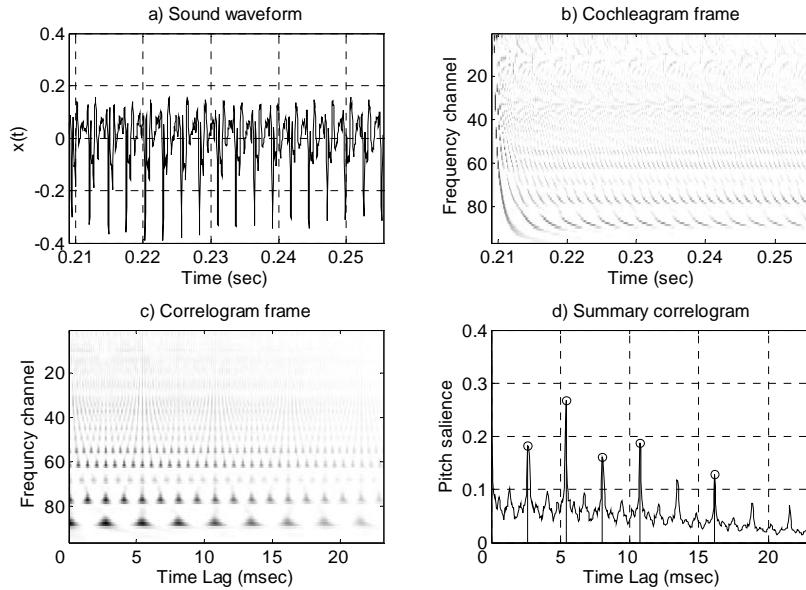
iii) Detection of the global periodicities in the sound waveform by calculation of a summary correlogram (SC);

iv) Detection of the pitch candidates in each time frame by looking for the most salient peaks in the SC (maximum of five peaks selected). For each obtained pitch, a pitch salience is computed, which is approximately equal to the energy of the corresponding fundamental frequency.

The four steps described are graphically illustrated in Figure 2, for a simple monophonic saxophone riff. This algorithm is described in greater detail in [1].

## 3 MULTI-PITCH TRAJECTORY CONSTRUCTION (MPTC)

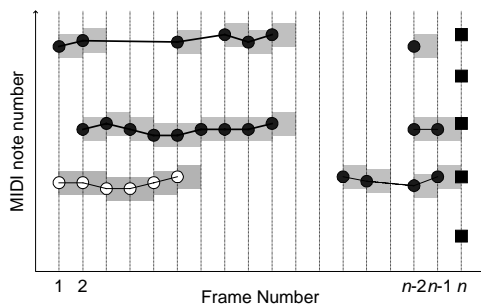
After multi-pitch detection, the goal is to quantise the temporal sequences of pitch estimates into note symbols characterized by precise timings and note values (e.g.,



**Figure 2.** Illustration of the four stages of the MPD algorithm.

MIDI note numbers). Therefore, the second stage, Multi-Pitch Trajectory Construction (MPTC), aims to create a set of pitch tracks, formed by connecting consecutive pitch candidates with similar frequencies. To this end, we based ourselves on the algorithm proposed by Serra [6]. The idea is to find regions of stable pitches, which indicate the presence of musical notes.

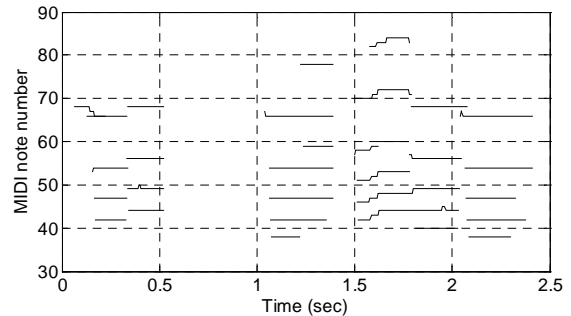
This algorithm is graphically illustrated in Figure 3. There, the black squares represent the candidate pitches in the current frame  $n$ . The black circles connected by thin continuous lines indicate the trajectories that have not been finished yet. The dashed lines denote peak continuation through sleeping frames. The black circles connected by bold lines stand for validated trajectories, whereas the white circles represent eliminated trajectories, due to too short lengths. Finally, the gray boxes indicate the maximum allowed frequency deviation for peak continuation in the corresponding frame.



**Figure 3.** Illustration of the MPTC algorithm.

In order not to lose information on the dynamic properties of musical notes, e.g., frequency modulations, glissandos, we had especial care in guaranteeing that such behaviours were kept within a single track. This is illustrated in Figure 4. There, we can see that some of the obtained trajectories comprise glissando regions. Also, some of the trajectories include more than one

note and should, therefore, be segmented in the third stage of our algorithm.



**Figure 4.** Results of the MPTC algorithm.

## 4 TRAJECTORY SEGMENTATION

The trajectories that result from the MPTC algorithm may contain more than one note and, therefore, must be segmented. This is the task of the third stage of the melody detection method.

Two types of segmentation are required: segmentation based on frequency and on pitch salience variations. Both are described followingly. Our proposed algorithm is described in greater detail in [2].

### 4.1 Frequency-Based Segmentation

In frequency-based segmentation, the goal is to split notes with different values that may be present in the same trajectory, taking into consideration the presence of glissandos and frequency modulation.

The main issue with frequency-based segmentation is to approximate the frequency curve by piecewise-constant functions (PCFs), as a basis for the definition of MIDI notes. However, this is often a complex task, since musical notes, besides containing regions of approximately stable frequency, also contain regions of

transition, where frequency evolves until (pseudo-)stability, e.g., glissando. Additionally, frequency modulation can also occur, where no stable frequency exists. Yet, an average stable fundamental frequency can be determined.

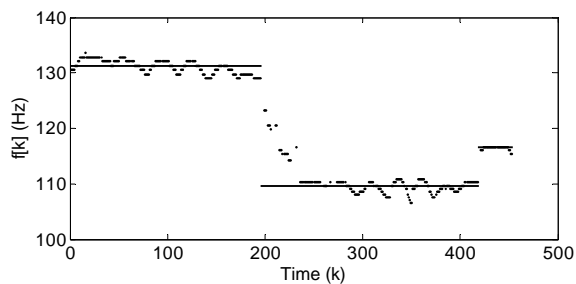
Our problem, could, thus, be characterized as one of finding a set of piecewise-constant/linear functions that best approximates the original frequency curve. As unknown variables we have the number of functions, their respective parameters (slope and bias – null slope if PCFs are used), and start and end points.

In short words, our algorithm first quantises the frequency values present in each track to the closest MIDI note numbers, thus obtaining a set of initial PCFs. Then, in order to cope with glissandos and oscillations resulting from vibrato, as well as frequency jitter and errors in the MPD stage, several stages of filtering are applied in order to merge relevant PCFs.

After filtering, the precise timings for the starting end ending points of each PCF are adjusted. We define the start of the transition as the point of maximum derivative of the frequency curve, after it starts to move towards the next note, i.e., the point of maximum derivative after the last occurrence of the median value.

Finally, we assign a definitive MIDI note number to each of the obtained PCFs for each track. In order to increase the robustness of the assignment procedure, we deal with ambiguous situations where it is not totally clear which is the correct MIDI value. This happens, for instance, when the median frequency is close to the frequency border of two MIDI notes.

The frequency-based segmentation algorithm is illustrated in Figure 5, for a pitch track from Mambo King’s “Bella Maria de Mi Alma”, where the continuous lines represent the obtained PCFs that approximate the original frequency curve.



**Figure 5.** Illustration of the frequency-based segmentation algorithm.

## 4.2 Saliency-Based Segmentation

As for saliency-based segmentation, the objective is to separate consecutive notes that have the same fundamental frequencies, which the MPTC algorithm may have interpreted as forming one single note. This requires segmentation based on pitch saliency minima, which mark the limits of each note.

In fact, the saliency value depends on the evidence of pitch for that particular frequency, which is strongly correlated, though not exactly equal, to the energy of the fundamental frequency under consideration. Consequently, the envelope of the saliency curve is similar to an amplitude envelope: it grows at the note onset, has then a more steady region and decreases at the offset. In

this way, notes can be segmented by detecting clear minima in the pitch saliency curve.

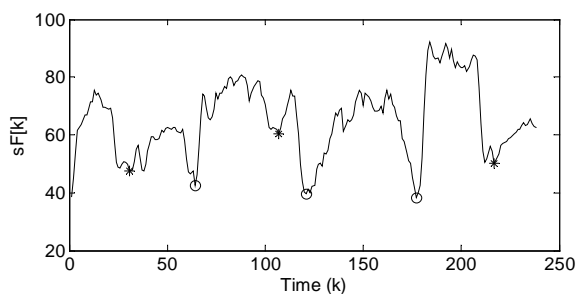
In a first attempt for performing saliency-based segmentation, we developed a prominent valley detection algorithm, which iteratively looks for all clear local minima and maxima of the saliency curve.

To this end, first, all local minima and maxima are found. Then, only clear minima are selected. This is accomplished in a recursive procedure that starts by finding the global minimum of the saliency curve. Next, the set of all local maxima is divided into two subsets, one to the left and another to the right of the global minimum. The global maximum for each subset is then obtained. After that, the global minimum is selected as a clear minima if its prominence, i.e., the minimum distance from its amplitude and that of both the left and right global maxima, is above the defined minimum peak-valley distance,  $minPvd$ .

Finally, the set of all local minima is also divided into two new intervals, to the left and right of the global minimum. The described procedure is then recursively repeated for each of the new subsets until all clear minima and respective prominences are found.

One difficulty of the proposed approach is its lack of robustness. In fact, the best value for  $minPvd$  was found to vary from track to track, along different song excerpts. In fact, a unique value for that parameter leads to both missing and extra segmentation points. Also, it is sometimes difficult to distinguish between note endings and amplitude modulation in some performances. Therefore, we improved our method by performing onset detection and matching the obtained onsets with the candidate segmentation points that resulted from our prominent valley detection algorithm. Onset detection was performed based on Scheirer [7] and Klapuri [8].

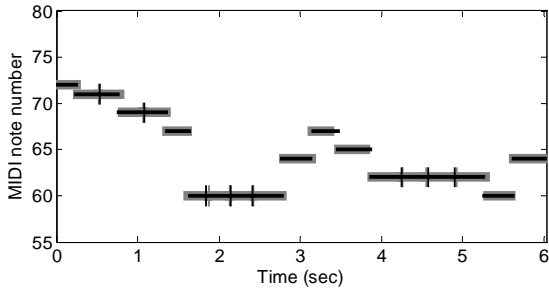
Figure 6 illustrates our algorithm for detection of candidate segmentation points. There, the pitch saliency curve of a trajectory from Claudio Roditi’s performance of “Rua Dona Margarida” is presented, where ‘o’ represent correct segmentation candidates and ‘\*’ denote extra segmentation points. Only the correct segmentation candidates should be validated based on the found onsets.



**Figure 6.** Illustration of the saliency-based segmentation algorithm: initial candidate points.

The results of the saliency-based segmentation algorithm for an excerpt from Claudio Roditi’s “Rua Dona Margarida” are presented in Figure 7. The gray horizontal lines represent the original annotated notes, whereas the black lines denote the extracted notes. The small gray vertical lines stand for the correct segmentation points and the black vertical ones are the obtained results of our algorithm. It can be seen that there is an al-

most perfect match when this solution is followed. However, in some excerpts extra segmentation occurs, especially in those excerpts with strong amplitude modulation.



**Figure 7.** Results of the salience-based segmentation algorithm.

## 5 NOTE ELIMINATION

The objective of the fourth stage of the melody detection algorithm is to delete irrelevant note candidates, based on their saliences, durations and on the analysis of harmonic relations.

First, low-salience notes are eliminated. Next, all the notes that are too short are also deleted. Finally, harmonically-related notes are discarded, based on the fact that some of the obtained pitch candidates are sub or super-harmonics of real pitches in the sound wave. Hence, the perceptual rules of sound organization designated as “harmonicity” and “common fate” are exploited [9].

In the “harmonicity” rule, if two notes have approximately the same onset times and are harmonically related, it is possible that they have come from the same source.

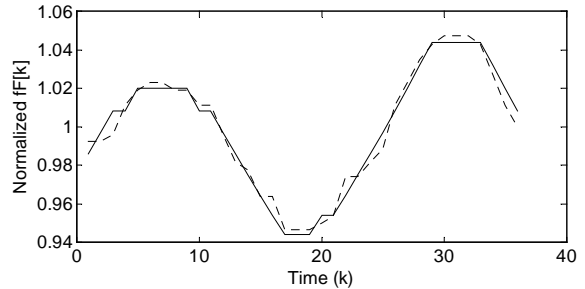
As for the “common fate” rule, harmonically-related notes can be grouped by taking advantage of aspects such as common modulation, both in frequency and in amplitude. In fact, components coming from the same source tend to have synchronized and parallel changes in frequency and intensity (here represented by pitch salience).

Thus, we measure the distance between frequency and salience curves for harmonically-related notes with common onsets, common offsets or inclusions. Formally, the distance between frequency curves is calculated as in (1), similarly to Virtanen [10]:

$$d_f(i, j) = \frac{1}{t_2 - t_1 + 1} \sum_{t=t_1}^{t_2} \left( \frac{f_i(t)}{\text{avg}(f_i(t))} - \frac{f_j(t)}{\text{avg}(f_j(t))} \right)^2 \quad (1)$$

where  $d_f$  represents the distance between two frequency trajectories,  $f_i(t)$  and  $f_j(t)$ , during the time interval  $[t_1, t_2]$  where they both exist. The idea of Eq. (1) is to scale the amplitude of each curve by its average, thus, normalizing it. An identical procedure is performed for the salience curves.

This procedure is illustrated in Figure 8 for two harmonically-related notes from an opera excerpt with strong vibrato. We can see that the normalized frequency curves are very similar, which provide good evidence that the notes originated from the same source.



**Figure 8.** Illustration of similarity analysis of frequency curves.

Additionally, we found it advantageous to also measure the distance between the salience and frequency derivatives since it sometimes happens that curves can have high absolute distances having, however, the same trends. By computing the distance between derivatives, those curves can also be considered similar.

Finally, we compare the saliences of pairs of harmonically-related notes that satisfy the common fate requirement in order to take a decision: if the salience of one of the notes is much lower than the other’s, the least salient note is eliminated.

## 6 MELODY EXTRACTION

Finally, in the melody extraction stage the objective 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.

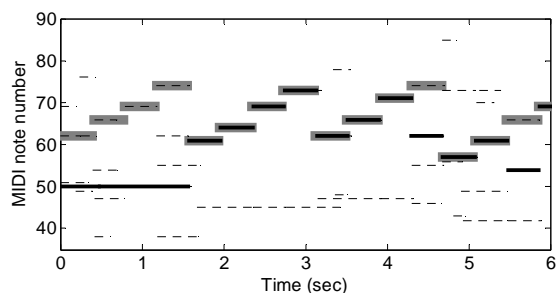
In the present approach, we do not tackle the problem of source separation. Instead, the strategy is based on two assumptions, designated as the “salience principle” and the “melodic smoothness principle”. Furthermore, we try to eliminate false positives, i.e., notes that are output by the algorithm whenever the melody is silent, which belong to the accompaniment parts. This algorithm is described in greater detail in [1, 3, 4].

### 6.1 Selection of the Most Salient Notes

The salience principle makes use of the fact that the main melodic line often stands out in the mixture. Thus, in the first step of the melody extraction stage, the most salient notes at each time are selected as initial melody note candidates. Details of this analysis are provided in [1].

The results of the implemented procedures are illustrated in Figure 9, for an excerpt from Pachelbel’s Canon. 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.

One of the limitations of only taking into consideration pitch salience is that the notes comprising the melody are not always the most salient ones. In this situation, erroneous 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 illustrated in Figure 9.



**Figure 9.** Results of the algorithm for extraction of salient notes.

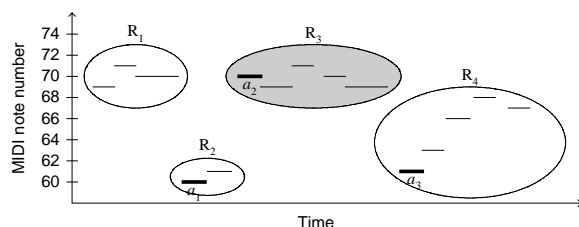
In fact, small frequency intervals favour melody coherence, since smaller steps in pitch result in melodies more likely to be perceived as single 'streams'. Hence, we improved our method by smoothing out the melody contour, as follows.

## 6.2 Melody Smoothing

As referred above, taking into consideration only the most salient notes has the limitation that, frequently, non-melodic notes are more salient than melodic ones. As a consequence, erroneous notes are often picked up, whereas true notes are excluded. Particularly, abrupt transitions between notes give strong evidence that wrong notes were selected. In fact, small frequency transitions favour melody coherence, since smaller steps in pitch hang together better [9].

Briefly, our algorithm starts with an octave correction stage, which aims to tackle some of the octave errors that appear as a consequence of the fact that not all harmonically-related notes are deleted at the note elimination stage.

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 10. 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  $R_4$ ).

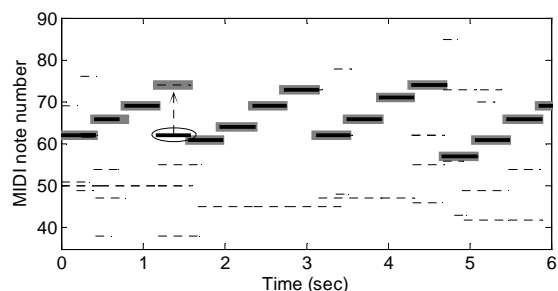


**Figure 10.** Regions of smoothness.

Then, we analyse the regions of smooth, deleting or substituting notes corresponding to abrupt transitions, as described in detail in [3].

The results of the implemented procedures are illustrated in Figure 11, for the same excerpt from Pachelbel's Kanon presented before. We can see that only one erroneous note resulted (signaled by an ellipse), which corresponds to an octave error. This example is particu-

larly challenging to our melody-smoothing algorithm due to the periodic abrupt transitions present. Yet, the performance was very good.



**Figure 11.** Results of the melody-smoothing algorithm.

## 6.3 Elimination of Spurious Notes

As referred, when pauses between melody notes are fairly long, spurious notes, resulting either from noise or background instruments, may be included in the melody. We observed that, usually, such notes have lower saliences and shorter durations, leading to clear minima in the pitch salience and duration contours.

Regarding the pitch salience contour, we start by computing the average pitch salience of each note in the extracted melody and, then, look for deep valleys in the pitch salience sequence. As with salience-based segmentation, we detect clear minima in the salience contour and delete notes in deep valleys of the pitch salience contour.

Regarding the duration contour, we proceeded likewise. However, we observed that duration variations are much more common than pitch salience variations. In this way, we decided to eliminate only isolated abrupt duration transitions, i.e., isolated notes delimited by much longer notes. Additionally, in order not to inadvertently delete short ornamental notes, a minimum difference of two semi-tones was defined.

This algorithm is described with more detail in [4].

## 7 EXPERIMENTAL RESULTS

In this paper, we present the results achieved for the MIREX 2005 Audio Melody Extraction Contest [11]. A collection of excerpts from genres such as Rock, R&B, Jazz and Opera, of around 30 seconds each, were used.

We participated with the described algorithm, as well as a simpler version where only one pitch candidate was selected in each frame, rather than a multi-pitch approach.

As for the obtained results, an average of 57.3% correctly transcribed voiced and unvoiced portions was achieved. Contrariwise to our previous results, the single-pitch approach achieved a surprisingly higher accuracy: 60.7%.

Taking into consideration only the voiced portions, the accuracy was 62.2% for the multi-pitch (MP) method and 58.2% for the SP method (raw pitched accuracy metric).

Disregarding octave errors, the MP algorithm achieved 66.2%, whereas the SP attained 61.6% (raw chroma accuracy metric).

Other results are listed below (SP and MP):

- Voicing detection: 68.0%, 82.4%;
- Voicing false alarm: 23.2% 55.8%;
- Voicing d-prime: 1.20, 0.78;

Finally, execution time is the main limitation of our proposed algorithm: the whole data set took around 45000s to be processed. This very long computing time results mostly from the implementation of the auditory model: more or less 90% of the execution time is devoted to the MPD stage! Therefore, a more efficient front-end should be considered. Additionally, the Matlab implementation is also responsible for the resulting long computing time.

## ACKNOWLEDGEMENTS

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

## REFERENCES

- [1] Paiva, R. P., Mendes, T., and Cardoso, A. "An auditory model based approach for melody detection in polyphonic musical recordings". In Wil, U. K. (ed.), *Computer Music Modelling and Retrieval - CMMR 2004, Lecture Notes in Computer Science*, Vol. 3310, 2005.
- [2] Paiva, R. P., Mendes, T., and Cardoso, A. "Segmentation of Pitch Tracks for Melody Detection in Polyphonic Audio", *Proceedings of the European signal Processing Conference (EUSIPCO)*, 2005.
- [3] Paiva, R. P., Mendes, T., and Cardoso, A. "Exploiting Melodic Smoothness for Melody Detection in Polyphonic Audio", *Proceedings of the International Conference on Computer Music (ICMC)*, 2005.
- [4] Paiva, R. P., Mendes, T., and Cardoso, A. "On the Detection of Melody Notes in Polyphonic Audio", *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2005.
- [5] 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.
- [6] Serra, X. "Musical sound modeling with sinusoids plus noise". In Roads, C., Pope, S., Picialli, A., De Poli, G. (eds.), *Musical signal processing*, 1998.
- [7] Scheirer, E. D. "Tempo and beat analysis of acoustic musical signals", *Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–601, 1998.
- [8] Klapuri, A. "Sound onset detection by applying psychoacoustic knowledge", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1999.
- [9] Bregman, A. S. *Auditory scene analysis: the perceptual organization of sound*. MIT Press, 1990.
- [10] Virtanen, T. and Klapuri, A. "Separation of harmonic sound sources using sinusoidal modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2000.
- [11] MIREX, "Music Information Retrieval exchange", <http://www.music-ir.org/mirexwiki/index.php/>, 2005.