# "AN APPROACH FOR MELODY EXTRACTION FROM POLYPHONIC AUDIO: USING PERCEPTUAL PRINCIPLES AND MELODIC SMOOTHNESS"

**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

In this research work the problem of melody extraction from polyphonic audio is addressed. A multi-stage approach is followed, inspired on principles from perceptual theory and musical practice. Physiological models and perceptual cues of sound organization are incorporated into the method, mimicking the behavior of the human auditory system to some extent. Moreover, musicological principles are applied, in order to support the identification of the musical notes that convey the main melodic line.

The system comprises three main modules, where a number of rule-based procedures are proposed: i) pitch detection, where an auditory model-based pitch detector is employed for selecting multiple pitches in each analysis frame; ii) determination of musical notes (with precise temporal boundaries and pitches); and iii) identification of melodic notes, based on two core assumptions that we designate as the salience principle and the melodic smoothness principle.

Experimental results were conducted, showing that the method performs satisfactorily under the specified assumptions, namely when the notes comprising the melody are in general more intense than the accompanying instruments. However, additional difficulties are encountered in song excerpts where the intensity of the melody in comparison to the surrounding accompaniment is not so favorable.

## 1 INTRODUCTION

This paper outlines an algorithm for melody detection in polyphonic audio signals. The proposed system comprises three main 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].

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.

Unlike most other melody-extraction systems, we attempt to explicitly distinguish individual musical notes (in terms of their pitches, timings, and intensity levels). This is the goal of the second stage of the algorithm (Determination of Musical Notes, in Figure 1). Here, we first create pitch tracks by connecting pitch candidates with similar frequency values in consecutive frames (the pitch trajectory construction, or PTC, step). The resulting pitch tracks may contain more than one note and should, therefore, be segmented in time. This is performed in two phases, namely frequency-based segmentation and salience-based segmentation.

In the last stage, our goal is to identify the final set of notes representing the melody of the song under analysis. To this end, ghost harmonically-related notes are first eliminated based on perceptual sound organization principles such as harmonicity and common fate. Then, we select the notes with highest pitch salience at each moment. The melodic contour is then smoothed out, based on the fact that pitch intervals between consecutive are usually small in tonal melodies.

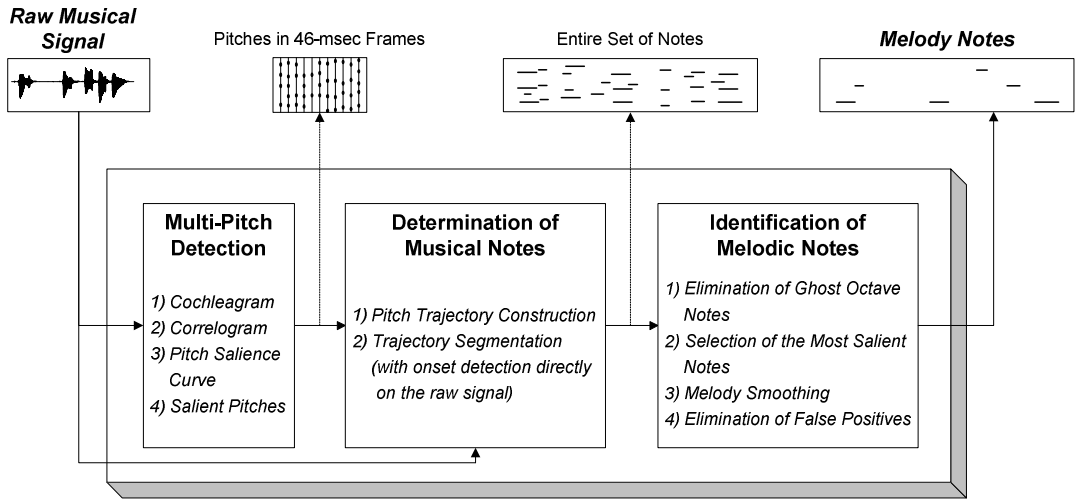


**Figure 1**. Melody detection system overview.

Each of the modules will be described in the next sections.

## 2 MULTI-PITCH DETECTION (MPD)

In the first stage of the algorithm, Multi-Pitch Detection (MPD) is conducted, with the objective of capturing the most salient pitch candidates in each time frame that constitute the pool of possible future notes.

Our pitch detector is based on Slaney and Lyon's auditory model [5], using 46.44-msec frames with a hop size of 5.8 msec. 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 (F0).

The four steps described are graphically illustrated in Figure 3, for a simple monophonic saxophone riff. The algorithm is described in greater detail in [3].

## 3 DETERMINATION OF MUSICAL NOTES

After multi-pitch detection, the goal is to quantize the temporal sequences of pitch estimates into note symbols characterized by precise timings and pitches (e.g., MIDI note numbers). This is carried out in three steps: pitch trajectory construction, frequency-based segmentation and salience-based segmentation (with onset detection directly on the raw signal).

### 3.1 Pitch Trajectory Construction (PTC)

In the Pitch Trajectory Construction (PTC), we first create pitch tracks by connecting pitch candidates with similar frequency values in consecutive frames. We based our approach on the algorithm proposed by Xavier Serra [6]. The general idea is to find regions of stable pitches that indicate the presence of musical notes.

This algorithm is graphically illustrated in Figure 2. 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.

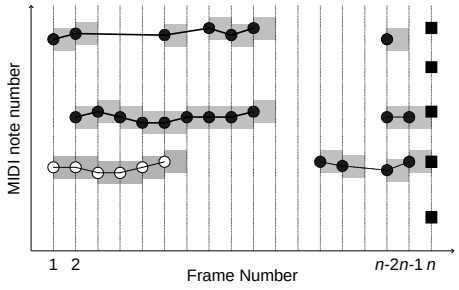


**Figure 2.** Illustration of the PTC algorithm.

To avoid losing information on the dynamic properties of musical notes, we took special care to keep phenomena such as vibrato and glissando within a single track. This is illustrated in Figure 4.

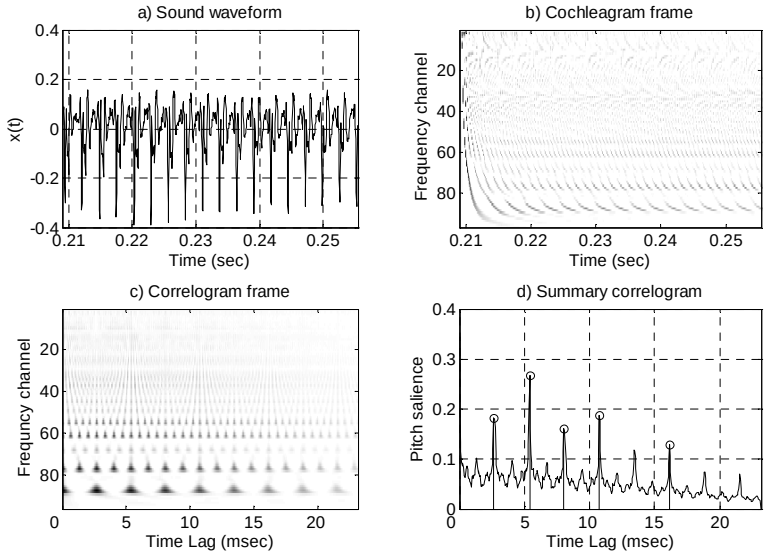


**Figure 3**. Illustration of the four stages of the MPD algorithm.

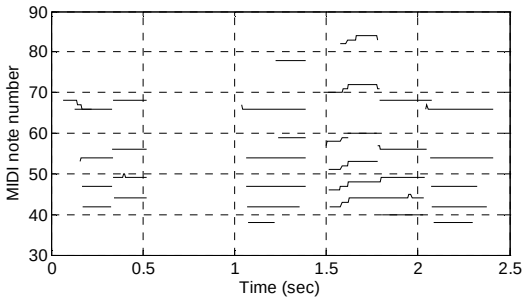


**Figure 4.** Results of the PTC algorithm.

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.

### 3.2 Frequency-based Segmentation

In frequency-based segmentation, the goal is to separate all notes of different pitches that might be present in the same trajectory. This is accomplished by approximating the pitch sequence in each track by a set of piecewise constant functions (PCFs), handling glissando, legato, vibrato, and frequency modulation in general. Each detected function will then correspond to a MIDI note. Despite this quantization effect, the original pitch sequences are still kept so that the information on note dynamics is not lost.

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. The procedures conducted towards this goal are described in detail in [4].

In short words, our algorithm first quantizes 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, a situation that might result from imperfect tuning. 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 a female opera excerpt with strong vibrato. There, dots denote the F0 sequence under analysis, grey lines are the reference segmentations, dashed lines denote the results attained prior to time correction and final note labelling and solid lines stand for the final achieved results. It can be seen that the segmentation methodology works quite well in these examples, despite some minor timing errors that may have even derived from annotation inaccuracies.

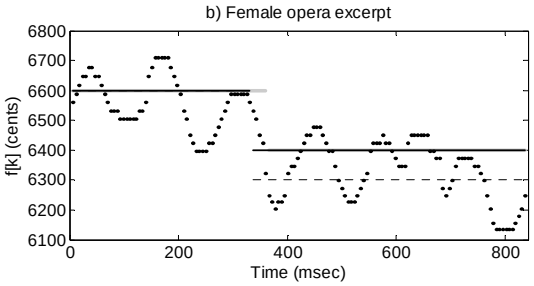


**Figure 5.** Illustration of the frequency-based segmentation algorithm.

The algorithm for frequency segmentation is based on a minimum note duration of 125 msec. This threshold was set based on the typical note durations in Western music. As Albert Bregman points out, "Western music tends to have notes that are rarely shorter than 150 msec in duration" [7, p. 462]. We experimented with a range between 60 and 150 msec, but the defined threshold of 125 msec led to the best results. It is noteworthy that this value is close to the one mentioned by Bregman.

### 3.3 Salience-Based Segmentation

With segmentation based on pitch salience variations, the objective is to separate consecutive notes at the same pitch that the PTC algorithm may have mistakenly interpreted as forming only one note. This requires trajectory segmentation based on pitch-salience minima, which mark the temporal boundaries of each note. To increase the robustness of the algorithm, note onsets are detected directly from the audio signal and used to validate the candidate salience minima found in each pitch track.

In fact, the salience 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 salience curve is similar to an amplitude envelope: it grows at the note onset, has then a steadier region and decreases at the offset. In this way, notes can be segmented by detecting clear minima in the pitch salience curve.

In a first attempt for performing salience-based segmentation, we developed a prominent valley detection algorithm, which iteratively looks for all clear local minima and maxima of the salience 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 salience 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 [8] and Klapuri [9].

Figure 6 illustrates our algorithm for detection of candidate segmentation points. There, the pitch salience 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.

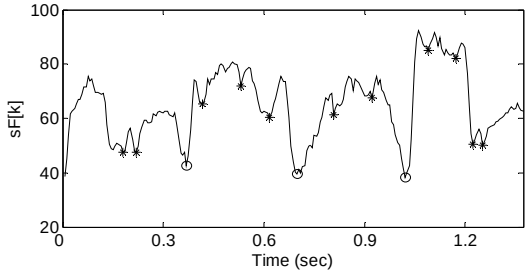


**Figure 6.** Illustration of the salience-based segmentation algorithm: initial candidate points.

The results of the salience-based segmentation algorithm for an excerpt from Claudio Roditi's "Rua Dona Margarida" are presented in Figure 7.

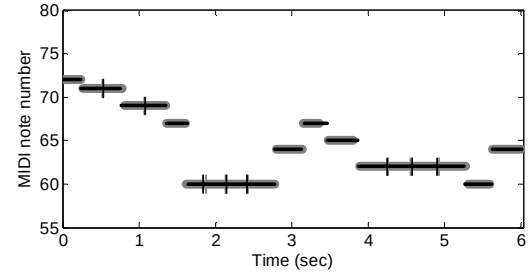


**Figure 7.** Results of the salience-based segmentation algorithm.

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

The procedures carried out for salience-based segmentation are described in greater detail in [4].

# 4 IDENTIFICATION OF MELODIC NOTES

After the first two stages of our system (see Figure 1), several notes from each of the different instruments present in the piece under analysis are obtained, among which the main melody must be identified. The separation of the melodic notes in a musical ensemble is not a trivial task. In fact, many aspects of auditory organization influence the perception of the main melody by humans, for instance in terms of the pitch, timbre, and intensity content of the instrumental lines in the sonic mixture. We start this stage by disposing of ghost octave notes

## 4.1 Elimination of Ghost Octave Notes

The set of candidate notes resulting from trajectory segmentation typically contains several ghost octave notes. The partials in each such note are actually multiples of the true note's harmonics (if the ghost octave note is higher than the true note) or submultiples (if it is lower). Therefore, the objective of this step is to discard such notes.

In short, we look for harmonic relations between all notes, based on the fact that some of the obtained pitch candidates are actually harmonics or sub-harmonics of true fundamental frequencies in the sound wave. Therefore, we make use of the perceptual rules of sound organization designated as harmonicity and common fate [7]. Namely, we look for pairs of octave-related notes with common onsets or endings and with common modulation, i.e., whose frequency and salience sequences change in parallel. We then delete the least-salient note if the ratio of its salience to the salience of the other note is below a defined threshold.

Regarding common fate analysis, we exploit the fact that frequency sequences belonging to the same note tend to have synchronized and parallel changes in frequency and intensity (here represented by pitch salience). Thus, we measure the distance between frequency curves for pairs of octave-related note candidates. Similarly, we measure the distance between their salience curves. Formally, the distance between frequency curves is calculated according to Eq. 1, based on [10]:

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

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 of vibrato. We can see that the normalized frequency curves are very similar, which provide good evidence that the notes originated from the same source.

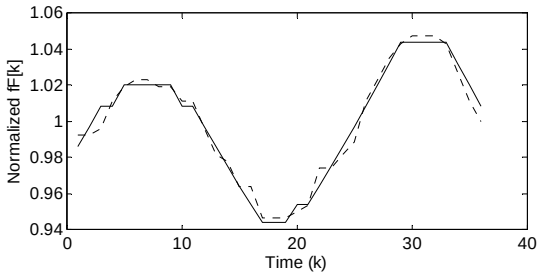


**Figure 8.** Illustration of similarity analysis of frequency curves.

Additionally, we found it advantageous to measure the distance between the normalized derivatives of frequency curves (and, likewise, the derivatives of salience curves). In fact, it is common that these curves have high absolute distances despite exhibiting the same trends. The distance between derivatives is used as another measure of curve similarity.

To conclude the common modulation analysis, we assume that the two candidate notes have parallel changes if any of the four computed distances (i.e., in frequency, salience, or their derivatives) are below a threshold of 0.04. Finally, we eliminate one of the notes if its salience is less than 40% of the most salient note if they differ by one octave, 20% if they differ by two octaves, and so forth.

### 4.2. Selection of the Most Salient Notes

As previously mentioned, intensity is an important cue in melody identification. Therefore, we select the most salient notes as an initial attempt at melody identification.

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, 2].

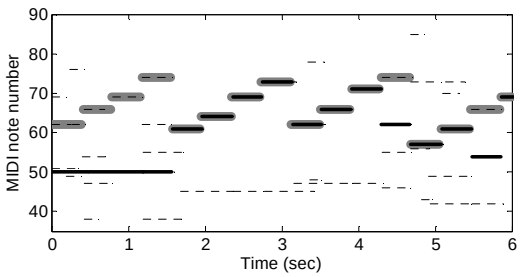


**Figure 9.** Results of the algorithm for extraction of salient notes.

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

In fact, small frequency intervals favor 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.

### 4.3 Melody Smoothing

As referred to 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 favor melody coherence, since smaller steps in pitch hang together better [7].

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 $R_4$).

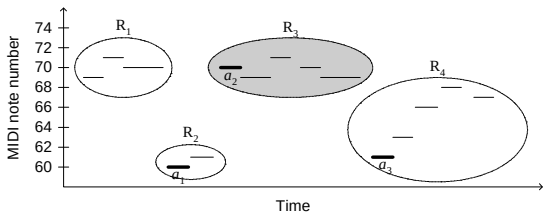


**Figure 10.** Regions of smoothness.

Then, we analyze the regions of smooth, deleting or substituting notes corresponding to abrupt transitions, as described in detail in [1, 2].

The results of the implemented procedures are illustrated in Figure 11, for the same excerpt from Pachelbel's Canon presented before. 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.

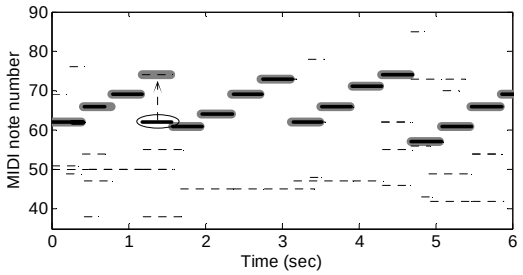


**Figure 11.** Results of the melody-smoothing algorithm.

### 4.4 Elimination of False Positives

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].

## 5 EXPERIMENTAL RESULTS

One difficulty regarding the evaluation of MIR systems comes from the lack of meaningful standard test collections and benchmark problems. This was partly solved through the creation of a set of evaluation databases for the ISMIR 2004 Melody Extraction Contest (MEC-04) and for MIREX 2005.

Thus, we evaluated the proposed algorithms with both the MEC-04 database and a small database we had previously created. Each of these databases were designed taking into consideration diversity and musical content. Therefore, the selected song excerpts contain a solo (either vocal or instrumental, corresponding to the main melody) and accompaniment parts (guitar, bass, percussion, other vocals, etc.). Additionally, in some excerpts, the solo is absent for some time. In our test bed, we collected excerpts of about 6 sec from 11 songs that were manually annotated with the correct notes. As for the MEC-04 database, 20 excerpts, each around 20 sec, were automatically annotated based on monophonic pitch estimation from multi-track recordings, as described in [11]. From these, we employed the defined training set, consisting of 10 excerpts.

Regarding multi-pitch detection, we achieved 81.0% average pitch accuracy (nearly the same, i.e., 81.2%, if octave errors are ignored).

As for note determination, pitch tracks were segmented with reasonable accuracy. In terms of frequency-based segmentation, average recall (i.e., the percentage of annotated segmentation points correctly identified, was 72%, and average precision (i.e., the percentage of identified segmentation points that corresponded to actual segmentation points) was 94.7%. Moreover, the average time error was 28.8 msec (which may be slightly distorted by annotation errors), and the average semitone error rate for the melodic notes was 0.03%.

Regarding salience-based segmentation, many false positives resulted, with a consequent decrease in average precision (41.2%), against 75.0% average recall.

As for the elimination of ghost notes, an average of 38.1% of notes from the note-determination stage were eliminated, among which only 0.3% of true melodic notes were inadvertently deleted.

Finally, in terms of melody identification, 84.4% average accuracy was attained considering only the melodic notes. The achieved performance decreases when we take also into account the regions where the main melody is absent. There, no notes should be output. Thus, in these "empty" frames we define a target F0 of 0Hz which should be matched against the generated melody. In this case the melody detection accuracy drops to 77%. In fact, our algorithm shows a limitation in disposing of false positives (i.e., accompaniment or noisy notes): 31.0% average recall and 52.8% average precision. This is a direct consequence of the fact that the algorithm is biased detecting the maximum of melodic notes, no matter if false positives are included. A pilot study employing note clustering was conducted to improve this limitation, which needs to be further elaborated.

We also evaluated our system in the MIREX 2005 database. There, the average accuracy dropped to 61.1% (considering both melodic and non-melodic frames). The main apparent cause for this decrease was that the signal to noise ratio in the used excerpts was not so favourable, i.e., the ratio of the energy of the melodic part against "all the rest" was not so high.

## 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. "Melody Detection in Polyphonic Musical Signals: Exploiting Perceptual Rules, Note Salience and Melodic Smoothness", Computer Music Journal, Vol. 30(4), pp. 80-98, 2006.

[2] 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.

[3] 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.

[4] Paiva, R. P., Mendes, T., and Cardoso, A. "On the Definition of Musical Notes from Pitch Tracks for Melody Detection in Polyphonic Recordings", Proceedings of the Internatiomal Conference on Digital Audio Effects – DAFx'05, 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] Bregman, A. S. Auditory scene analysis: the perceptual organization of sound. MIT Press, 1990.

[8] 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.

[9] Klapuri, A. "Sound onset detection by applying psychoacoustic knowledge", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1999.

[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] Gómez, E., et al. "A Quantitative Comparison of Different Approaches for Melody Extraction from Polyphonic Audio Recordings." Technical Report MTG-TR- 2006-01. Barcelona: University Pompeu Fabra, Music Technology Group, 2006.