A Quantitative Comparison of Different Approaches for Melody Extraction from Polyphonic Audio Recordings

Emilia Gómez[1], Sebastian Streich[1], Beesuan Ong[1], Rui Pedro Paiva[2], Sven Tappert[3], Jan-Mark Batke[3], Graham Poliner[4], Dan Ellis[4], Juan Pablo Bello[5]
[1]Universitat Pompeu Fabra, [2]University of Coimbra, [3]Berlin Technical University,[4]Columbia University, [5]Queen Mary University of London

MTG-TR-2006-01
April 6, 2006

Abstract: This paper provides an overview of current state-of-the-art approaches for melody extraction from polyphonic audio recordings, and it proposes a methodology for the quantitative evaluation of melody extraction algorithms. We first define a general architecture for melody extraction systems and discuss the difficulties of the problem in hand; then, we review different approaches for melody extraction which represent the current state-of-the-art in this area. We propose and discuss a methodology for evaluating the different approaches, and we finally present some results and conclusions of the comparison.

A Quantitative Comparison of Different Approaches for Melody Extraction from Polyphonic Audio Recordings

Emilia Gómez[1], Sebastian Streich[1], Beesuan Ong[1], Rui Pedro Paiva[2], Sven Tappert[3], Jan-Mark Batke[3], Graham Poliner[4], Dan Ellis[4], Juan Pablo Bello[5]

[1]Universitat Pompeu Fabra, [2]University of Coimbra, [3]Berlin Technical University, [4]Columbia University, [5]Queen Mary University of London

Abstract: This paper provides an overview of current state-of-the-art approaches for melody extraction from polyphonic audio recordings, and it proposes a methodology for the quantitative evaluation of melody extraction algorithms. We first define a general architecture for melody extraction systems and discuss the difficulties of the problem in hand; then, we review different approaches for melody extraction which represent the current state-of-the-art in this area. We propose and discuss a methodology for evaluating the different approaches, and we finally present some results and conclusions of the comparison.

Index Terms—Melody Extraction, Music Information Retrieval, Evaluation

1. Introduction

Music Content Processing has recently become an active and important research area, largely due to the great amount of audio material that has become accessible to the home user through networks and other media. The need for easy and meaningful interaction with this data has prompted research into techniques for the automatic description and handling of audio data. This area touches many disciplines including signal processing, musicology, psychoacoustics, computer music, statistics, and information retrieval.

In this context, melody plays a major role. Selfridge-Field (1998, p.4) states that: "It is melody that enables us to distinguish one work from another. It is melody that human beings are innately able to reproduce by singing, humming, and whistling". The importance of melody for music perception and understanding reveals that beneath the concept of melody there are many aspects to consider, as it carries

implicit information regarding harmony and rhythm. This fact complicates its automatic representation, extraction and manipulation. In fact, automatic melody extraction from polyphonic and multi-instrumental music recordings is an issue that has received far less research attention than similar music content analysis problems, such as the estimation of tempo and meter.

This is changing in recent years with the proposal of a number of new approaches (Goto 2000, Eggink 2004, Marolt 2004, Paiva 2004), which is not surprising given the usefulness of melody extraction for a number of applications including: content-based navigation within music collections (query by humming or melodic retrieval), music analysis, performance analysis in terms of expressivity, automatic transcription and content-based transformation. As with any emergent area of research, there is little agreement on how to compare the different approaches, so as to provide developers with a guide to the best methods for their particular application.

This paper aims to provide an overview of current state-of-the-art approaches for melody extraction from polyphonic audio recordings, and to propose a methodology for the quantitative evaluation of melody extraction systems. Both this objectives were pursued under the context of the ISMIR 2004 Melody Extraction Contest (2004), organized by the Music Technology group of the Universitat Pompeu Fabra.

The rest of this paper is organized as follows: In Section 2 we propose a general architecture for melody extraction systems and discuss the difficulties of the problem in hand; in Section 3 we present an overview of the participants' approaches to the ISMIR 2004 melody extraction contest, hence reflecting the current state-of-the-art in this area; Section 4 proposes a methodology for the evaluation of melody extraction approaches; results of this evaluation on the reviewed approaches are presented and discussed in Section 5; and finally, section 6 presents the conclusions of our study and proposes directions for the future.

2. Melody Extraction

Figure 1 roughly illustrates the procedure employed in the majority of melody extraction algorithms: a feature set is derived from the original music signal, usually describing the signal's frequency behavior.

Then, fundamental frequencies are estimated before finally segregating signal components from the mixture to form melodic lines or notes. These are, in turn, used to create a transcription of the melody.

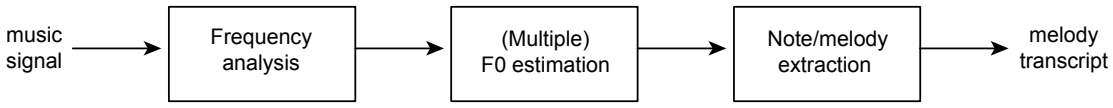

Figure 1: Overview of melody extraction system

Fundamental frequency is then the main low-level signal feature to be considered. It is important in both speech and music analysis, and is intimately related to the more subjective concept of pitch, or tonal height, of a sound. Although there has been much research devoted to pitch estimation, it is still an unsolved problem even for monophonic signals (as reviewed by Gomez et al. 2003).

Several approaches have been proposed for polyphonic pitch estimation (Klapuri 2000, Klapuri 2004, Marolt 2004, Dixon 2000, Bello 2004), showing different degrees of success and mostly constrained to musical subsets, e.g. piano music, synthesized music, random mixtures of sound, etc.

However, melody extraction and proposed approaches to polyphonic pitch estimation differ on that, along with the estimation of the predominant pitch in the mixture, melody extraction requires the identification of the voice that defines the melody within the polyphony. This later task is closer to using the principles of human auditory organization for pitch analysis, as implemented by Kashino et al. (1995) by means of a Bayesian probability network, where bottom-up signal analysis could be integrated with temporal and musical predictions, and by Wamsley & Godsill (1999), that use the Bayesian probabilistic framework to estimate the harmonic model parameters jointly for a certain number of frames. An example specific to melody extraction is the system proposed by Goto (2000). This approach is able to detect melody and bass lines by making the assumption that these two are placed in different frequency regions, and by creating melodic tracks using a multi-agent architecture. Other relevant methods are listed in (Gomez et al. 2003) and (Klapuri 2004).

There are other, more musicological aspects that make the task of melody extraction very difficult (Nettheim 1992). In order to simplify the issue, one should try to detect note groupings. This would

provide heuristics that could be taken as hypothesis in the melody extraction task. For instance, experiments have been done on the way the listener achieves melodic groupings in order to separate the different voices (see (Mc Adams 1994) and (Scheirer 2000 p.131)).

Other approaches can also simplify the melody extraction task by making assumptions and restrictions on the type of music that is analyzed. Methods can be different according to the complexity of the music (monophonic or polyphonic music), the genre (classical with melodic ornamentations, jazz with singing voice, etc) or the representation of the music (audio, midi etc).

For many applications, it is also convenient to see the melody as a succession of pitched notes. This melodic representation accounts for rhythmic information as inherently linked to melody. This poses the added problem of delimitating the boundaries of notes, in order to be able to identify their sequences and extract the descriptors associated to the segments they define.

ID	***System***	***Frequency analysis***	***Feature computation***	***F0 estimation***	***Post-processing***
1	Paiva	Cochlear model	Autocorrelation	Peaks in summary auto-correlation	Peak tracking, Segmentation and filtering (smoothness, salience)
2	Tappert & Batke	Multirate filterbank	Quantization to logarithmic frequency resolution	EM fit of tone models within selected range	Tracking agents
3	Poliner & Ellis	Fourier transform	Energies of spectral lines <2kHz	Trained SVM classifier	
4	Bello	HP filtering; frame-based autocorrelation		Peak picking	Peak tracking and rule-based filtering

Table 1. Comparison of different approaches

3. Approaches to melody extraction

As mentioned before, a number of approaches have been recently proposed to tackle the problem of automatically extracting melodies from polyphonic audio. To capitalise on the interest of both researchers and users, the ISMIR 2004 Melody Extraction Contest was proposed, aiming to evaluate and

compare state-of-the-art algorithms for melody extraction. Following an open call for submissions, four algorithms were received and evaluated. The corresponding methods represent an interesting and broad range of approaches, as can be seen in Table 1.

3.1. Paiva

This approach comprises five modules, as illustrated in Figure 2. A detailed description of the method can be found in (Paiva et al. 2004, 2004b and 2004c).

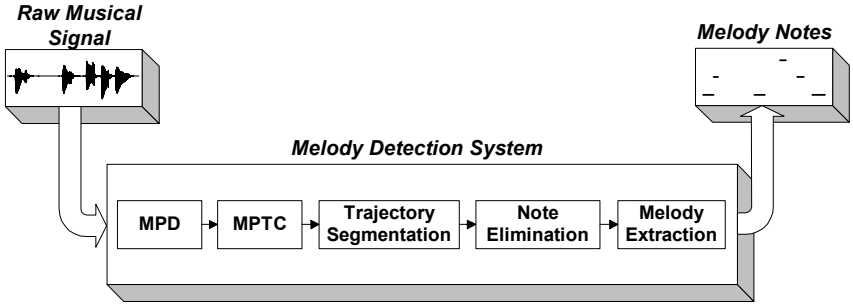

Figure 2. Overview of Paiva's melody detection system.

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, following Slaney and Lyon (1993). 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.

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. The idea is to find regions

of stable pitches, which indicate the presence of musical notes. This is based on Serra's peak continuation algorithm (Serra 1997). In order not to loose information on the dynamic properties of musical notes, e.g., frequency modulations, glissandos, this approach had especial care in guaranteeing that such behaviors were kept within a single track.

Thus, each trajectory that results from the MPTC algorithm may contain more than one note and should, therefore, be segmented in the third step. Such segmentation is performed in two phases: frequency and salience segmentation. Regarding frequency segmentation, the goal is to separate all the different frequency notes that are present in the same trajectory, taking into consideration the presence of glissandos and frequency modulation. As for pitch salience segmentation, it aims at separating consecutive notes with equal values, which the MPTC algorithm may have interpreted as forming only one note. This requires segmentation based on salience minima, which mark the limits of each note. In fact, the salience value depends on the evidence of pitch for that particular frequency, which is lower at the onsets and offsets. 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. Thus, notes can be segmented by detecting clear minima in the pitch salience curve.

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. Low-salience notes, too-short notes and harmonically-related notes are discarded. Hence, the perceptual rules of sound organization designated as "harmonicity" and "common fate" are exploited (Bregman 1990 pp. 245-292).

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 the present approach, the problem of source separation is not attacked. Instead, the strategy is rooted in two assumptions, designated as the "salience principle" and the "melodic smoothness principle". 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. 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. In fact, small frequency intervals favour melody coherence, since smaller steps in pitch result in melodies more likely to be perceived as single 'streams' (Bregman 1990 pp. 462). Thus, melody extraction is improved by taking advantage of the melodic smoothness principle, where notes corresponding to abrupt transitions are substituted by salient notes in the allowed range.

3.2. Tappert and Batke

This transcription system is originally a part of a query by humming (QBH) system (Batke et al. 2004). It is implemented using mainly parts of the system PreFEst described in (Goto 2000).

The audio signal is fed into a multirate filterbank containing five branches, and the signal is down sampled stepwise from Fs/2 to Fs/16 in the last branch, where Fs is the sample rate (see also (Fernández-Cid and Casajús-Quirós 1998) for using such a filterbank). A short-time Fourier transform (STFT) is used with a constant window length N in each branch to obtain a better time frequency resolution for lower frequencies. In our system, we used Fs = 16 kHz and N = 4096.

Quantization of frequency values following the equal tempered scale leads to a sparse spectrum with clear harmonic lines. The band pass simply selects the range of frequencies that is examined for the melody and the bass lines.

The expectation-maximization (EM) algorithm (Moon 1996) uses the simple tone model described above to maximize the weight for the predominant pitch in the examined signal. This is done iteratively leading to a maximum a posteriori estimate, see (Goto 2000). A set of F0 candidates is passed to the tracking agents that try to find the most dominant and stable candidates. The tracking agents are implemented similarly to those in Goto's work, but in a modified manner. The agents contain four time frames of F0 probability vectors - two of the past, the actual and the upcoming frame. These agents are filled with the

local maxima of the F0 probability vectors. To find the path of the predominant frequency, all maxima values over four frames within an agent are added, and to punish discontinuities this sum is divided by the number of gaps in the agent, e.g. the probability is zero. Finally, the agent with the highest score determines the fundamental frequency found.

3.3. Poliner and Ellis

In this system, the melody transcription problem was approached as a classification task. The system proposed by Poliner and Ellis uses a Support Vector Machine (SVM) trained on audio synthesized from MIDI data to perform N-way melodic note discrimination. Labeled training examples were generated by using the MIDI score as the ground truth for the synthesized audio features. Note transcription then consists simply of mapping the input acoustic vectors to one of the discrete note class outputs.

Although a vast amount of digital audio data exists, the machine learning approach to transcription is limited by the availability of labeled training examples. The analysis of the audio signals synthesized from MIDI compositions provides the data required to train a melody classification system. Extensive collections of MIDI files exist consisting of numerous renditions from eclectic genres. The training data used in this system is composed of 32 frequently downloaded pop songs from www.findmidis.com.

The training files were converted from the standard MIDI file format to mono audio files (.WAV) with a sampling rate of 8 kHz using the MIDI synthesizer in Apple's iTunes. A Short Time Fourier Transform (STFT) was calculated using 1024-point Discrete Fourier Transforms (128 ms), a 1024-point Hanning window, and a hop size of 512 points. The audio features were normalized within each time frame to achieve, over a local frequency window, zero mean and unit variance in the spectrum, in an effort to improve generalization across different instrument timbres and contexts. The input audio feature vector for each frame consisted of the 256 normalized energy bins below 2 kHz.

The MIDI files were parsed into data structures containing the relevant audio information (i.e. tracks, channels numbers, note events, etc). The melody was isolated and extracted by exploiting MIDI conventions for representing the lead voice. This is made easy because very often the lead voice in pop

MIDI files is represented by a monophonic track on an isolated channel. In the case of multiple simultaneous notes in the lead track, the melody was assumed to be the highest note present. Target labels were determined by sampling the MIDI transcript at the precise times corresponding to each STFT frame.

The WEKA implementation of Platt's Sequential Minimal Optimization (SMO) SVM algorithm was used to map the frequency domain audio features to the MIDI note-number classes (Witten and Frank 1999). The default learning parameter values (C=1, γ=0.01, ε= 10^{-12}, tolerance parameter = 10^{-3}) were used to train the classifier. Each audio frame was represented by a 256-feature input vector, and there were 60 potential output classes spanning the five-octave range from G2 to F#7. Approximately 128 minutes of audio data corresponding to 120,000 training points was used to train the classifier.

In order to predict the melody of the evaluation set, the test audio files were resampled to 8 kHz and converted to STFT features as above. The SVM classifier assigned a MIDI note number class to each frame. The output prediction file was created by interpolating the time axis to account for sampling rate differences and converting the MIDI note number to frequency using the formula $f = 440 \cdot 2^{\frac{m-69}{12}}$ Hz, where m is the MIDI note number (which is 69 for A440).

3.4. Bello

The approach here presented is previously unpublished. It is based on the limiting assumption that melody is a sequence of single harmonic tones, spectrally located at mid/high frequency values, carrying energy well above that of the background, and presenting only smooth changes in its frequency content.

The implemented process, illustrated in Figure 3, can be summarized as follows: First, the signal is pre-processed; then potential melodic fragments are built by following peaks from a sequence of autocorrelation functions (ACF); using a rule-based system, the fragments are evaluated against each other and finally selected to construct the melodic path that maximizes the energy while minimizing steep changes in the tonal sequence.

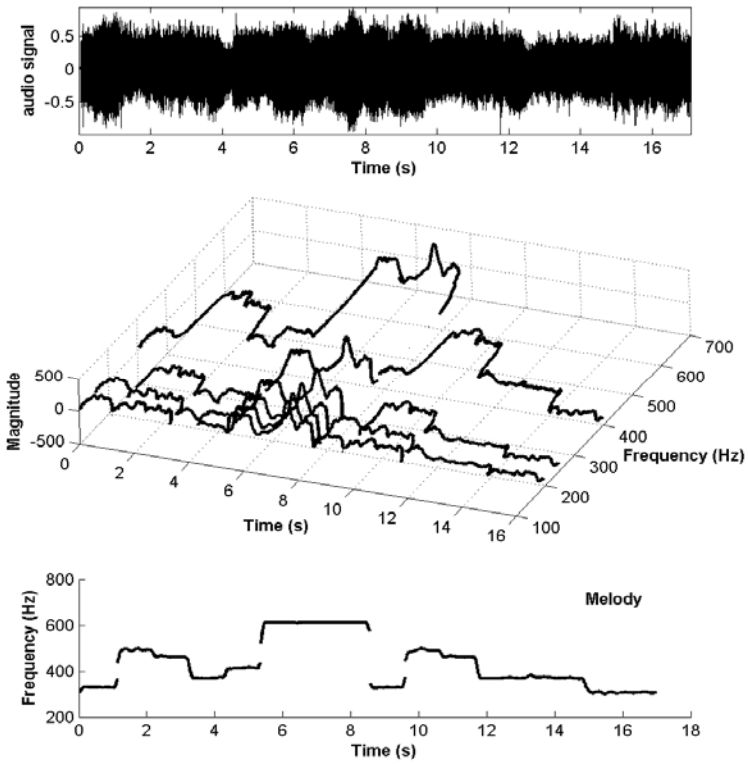

Figure 3. Overview of Bello's approach to melody extraction: polyphonic audio signal (top); F0 trajectories built from following peaks from the ACF (middle); melody estimated from selecting the path of trajectories that maximizes energy and minimizes frequency changes between trajectories (bottom).

The first stage of the analysis is to limit the frequency region where the melody is more likely to occur. To this end, a high-pass zero-shift elliptic filter is implemented, with cut-off frequency of 130 Hz. The aim of the filtering is to avoid any bias towards the bass-line contents of the signal, which is also expected to be very salient. The process is similar to the one used in (Goto 2000).

After pre-processing, the system attempts to detect strong and stable fundamental frequency (F0) trajectories in the audio signal. To do so, it explicitly assumes the melody to be composed of high-energy tones, strongly pitched and harmonic, and immersed in non-tonal background noise. This overly-simplistic model allows us to see the signal as monophonic, and the process of identifying melodic fragments in the mixture as segregating between voiced and unvoiced segments. This is simply not true

for most music signals, but results in later sections show that the approach is nevertheless able to estimate melodic lines from more complex backgrounds than the model seems to suggest.

An autocorrelation function-based algorithm (ACF) is chosen for the estimation of salient tones. ACF algorithms are well suited for the above model, given their known robustness to noise. They have been extensively used for pitch estimation in speech and monophonic music signals with harmonic sounds (Talkin 1995, Brown and Zhang 1991). The short-time autocorrelation r(n) of a K-length segment of x(k), the pre-processed signal, can be calculated as the inverse Fourier transform of its squared magnitude spectrum X(k), after zero-padding it to twice its original length (Klapuri 2000). For real signals, this can be expressed as:

$$r(n) = \frac{1}{K}\sum_{k=0}^{K-1}\left|X(k)\right|^2 \cos\left(\frac{2\pi nk}{K}\right)$$

The predominant-F0 period usually corresponds to one of the maxima in the autocorrelation function, in most cases the global maximum (excluding the zero-lag peak). The prominence of these maxima can be used to separate melodic fragment from other F0 trajectories: the ratio between the amplitude of the zero-lag peak and the amplitude of period-peaks (Yost 1996, Wiegrebe et al. 1998), and the ratio between peaks and background (Kaernbach and Demany 1998) have been used for measuring pitch strength, or for segregating voiced and unvoiced segments in the signal. Experiments have also shown that wider period peaks and the multiplicity of non-periodic peaks are signs of pitch weakness.

To create F0 trajectories, we use a peak continuation algorithm similar to the scheme used in (McAulay and Quatieri 1986) in the context of sinusoidal modelling. The algorithm tracks autocorrelation peaks across time. While constructing these trajectories, the algorithm also stores useful information about the relative magnitudes of the period peaks incorporated into these tracks. It also cleans the data by eliminating short trajectories that are likely to belong to noisy and transient components in the signal.

Once all F0 trajectories are created, the system uses a rule-based framework to evaluate the competing trajectories and determine the path that is most likely to form the melody. In a first stage of the competition, the system selects strong (i.e. with high energy content) and voiced (i.e. with high period-

peak to zero-lag peak ratio and high period-peak to background ratio) trajectories. For octave-related simultaneous trajectories, selection is biased towards higher-frequency trajectories, as ACF usually presents maxima in the position of the integer multiples of the F0 period (twice too-low pitches (Klapuri 2000)). Surviving trajectories are organized into time-aligned melodic paths. As mentioned before, the algorithm works on the assumption that melody is more distinctly heard than anything else in the mixture and that it will only present smooth frequency changes. Therefore, it selects the path of F0 trajectories that minimizes frequency and amplitude changes between them and maximizes the total energy of the melodic line. Future implementations of the system will explore the use of standard clustering algorithms to group trajectories according to the mentioned features (Marolt 2004).

4. Methodology for evaluation

There are a number of issues that make a fair comparison of existing approaches to melody extraction a hard task. First there is a lack of standard databases for training and evaluation. Researchers choose the style, instrumentation and acoustic characteristics of the test music as a function of their particular application. To this fact, we need to add the difficulties of producing reliable ground-truth data, which is usually a deterrent towards the creation of large evaluation databases. Furthermore, there are no standard rules regarding annotations, so different ground-truths are not compatible. Finally, different studies use different evaluation methods rendering numeric comparisons useless.

In the following, we propose solutions to the above issues, as first steps towards the generation of an overall methodology for the evaluation of melody extraction algorithms.

4.1. Evaluation material

Music exists in many styles and facets, and the optimal method for melody extraction for one particular type of music might be different from the one for another type. This implies that the material used for the evaluation needs to be selected from a variety of styles. The goal is to identify the style dependencies of proposed algorithms, and to determine which algorithm works best as a general-purpose melody extractor. An attempt was made to compile a set of musical excerpts that would present the algorithms

with different types of difficulties. A total of 20 polyphonic musical excerpts were chosen, each of around 20 seconds in duration. These segments can be categorized as shown in Table 2.

Category	Real/ Synthetic	Instrument carrying the melody	Style	Quantity of excerpts	Item number
Midi	Synthetic	MIDI instruments	Folksong (2), pop(2)	4	5,6
Jazz	Real	Saxophone	Jazz	4	3,4
Daisy	Synthetic	Voice	Pop	4	1,2
Pop	Real	Voice (male)	Pop	4	9,10
Opera	Real	Voice: male(2), female(2)	Classical (opera)	4	7,8

Table 2: Evaluation Material.

This limited set was collected as a representative sample, as it was not possible to cover all existing types of music in our evaluation set. The main limiting factors were the problems of access to copyright free material and the need for ground truth annotations.

A subset of the evaluation material (including the ground truth data) was made available to the participants in advance, thus allowing for training of the methods prior to the evaluation. For reasons of consistency, we combined half of the items from each category in this tuning set. Also, a software tool (Matlab script) was provided along with the training set. These scripts allowed the contestants to use the same evaluation method as in the final evaluation (see 4.3). The tuning set and the evaluation documents can be found at (ISMIR Contest Definition Page 2004).

4.2. Annotations

One of the most complex issues when generating an evaluation test set for melody extraction is that of obtaining ground truth annotations. Besides the well-known problems of access to licensed material, there are a number of practical issues that make the process of annotation very difficult indeed, as is already explained in (Lesaffre et al. 2004).

For recordings of polyphonic music presenting a multiplicity of instruments plus voice, we usually do not have a temporally synchronised annotation of the melodic notes. A way of overcoming this problem is using MIDI-controlled instruments that by definition produce a sonic output for which we have a

symbolic representation of its components. This is the case of the Midi and Daisy categories in our test set. However, synthesized music does not present the same acoustic complexity as music from real sounds. This makes for a test set which is not representative of the issues faced in real recordings, thus results in the evaluation will not be a good indication of the usefulness of extraction approaches in real applications.

Alternatively, melody transcriptions could be made from real data using trained individuals (such as musicians and musicologists). However, manual transcription is a very difficult and time-consuming process, and this solution has proven to be highly unpractical and usually costly.

We propose an alternative solution, where multi-track recordings of real music are used to create synchronised signals for both: the whole mixture and the melody only. The melody-only signal is processed through a monophonic pitch estimator (Cano 1998) and finally corrected manually to eliminate estimation errors. As a convention, a value of 0 Hz was assigned to non-melodic frames. Although this solution is limited by the availability of copyright-free multi-track recordings, it represents the best compromise we could find to deal with the problem of generating ground-truth data.

Furthermore, we propose a standard for the annotation of both, frame-by-frame and segmented melody information (ISMIR Contest Definition Page 2004).

4.3. Evaluation metrics

Three different evaluation metrics are proposed, as they were considered of interest to both, participants and potential developers. The code for computing the evaluations can also be found on the internet (ISMIR Contest Definition Page 2004).

4.3.1 Fundamental frequency

Metric 1 consists on a frame-based comparison of the estimated F0 and the annotated F0 on a logarithmic scale. The extracted pitch estimations are therefore converted to the cent:

$$f_{cent} = 1200 \cdot \left[\log_2 \left(\frac{f_{Hz}}{13.75Hz} \right) - 0.25 \right]$$

The concordance is measured by averaging the absolute difference between the reference pitch values and the extracted ones. The error is bounded by 1 semitone (i.e. 100 cents):

$$err_i = \begin{cases} 100 & for \ \left| f_{cent}^{est}(i) - f_{cent}^{ref}(i) \right| \geq 100 \\ \left| f_{cent}^{est}(i) - f_{cent}^{ref}(i) \right| & else \end{cases}$$

We obtain the final accuracy score in percent by subtracting the bounded mean absolute difference from 100:

$$score = 100 - \tfrac{1}{N} \sum_{i=1}^{N} err_i$$

Non-melodic frames are coded as zeros, so the bounded absolute error will be 100 in case of a mismatch. Each frame contributes to the final result with the same weight.

4.3.2 Fundamental frequency disregarding octave errors

In Metric 2, the values of F0 are mapped into the range of one octave before computing the absolute difference. Octave errors, which are a very common problem in F0 estimation, are disregarded this way.

It should be stated that these two metrics operate in a domain which is still close to the signal and not yet as abstract as a transcribed melody. The objective of the first two evaluation metrics is to measure the reliability of the fundamental frequency estimation.

4.3.3 Melodic similarity

For this metric, melodies are considered as sequences of segmented notes. Metric 3 is the edit distance between the estimated melody and the annotated melody according to the previous definition. The annotated melody was obtained by manual score alignment. Compared to the other two metrics, the abstraction level here is clearly higher, because note segmentation is required. Especially for sung melodies this is non-trivial, because of vibratos and strongly varying pitch.

The edit distance metric calculates the cost for transformation of one melody to another one. Different weights can be assigned to different transformation operations (insertion, deletion, replacement, etc.). It is very difficult to find the right configuration in order to model a human listener's impression of melodic

similarity or dissimilarity in a general way. We did not optimize or adapt the configuration in any way, because the main concern was the consistency of the evaluation. The disadvantage of this approach is that it is difficult to interpret the resulting numbers beyond a nature as an ordinal scale. In order to give the opportunity of a subjective comparison, we provide the extracted melodies as synthesized MIDI tracks on the contest webpage (ISMIR Contest Definition Page 2004). More information on the edit distance and its implementation is found in (Gratchen et al. 2002).

5. Results and discussion

5.1. About the experiments

The experiments were performed on the musical items described in Section 4.1. The sound file format was specified as 16-bit PCM wave files of monaural recordings sampled at 44.1 kHz. No interpolations were performed between the extracted and the annotated melodies, so all participants were requested to use an analysis window size of 2048 samples (46.44 ms), and a hop size of 256 samples (5.8 ms).

Algorithms were submitted by email. Deleting or updating a submission was possible until the final deadline. Participants were allowed to make their submissions either in binary format or as MATLAB/Octave script files. The only restrictions being the input/output definitions, which corresponded to the filenames of the input sound file and the output text file respectively. The output text file needed to follow the same formatting as provided for the annotations of the tuning material (ISMIR Contest Definition Page 2004). To avoid conflicts with disclosure agreements or the like, it was not mandatory for the participants to make their complete algorithms public.

5.2. Overall results

Figure 4 presents a summary of average results per approach, using metrics 1 and 2. This metrics evaluate the accuracy in pitch estimation for both melodic and non-melodic frames (thus implicitly evaluating melodic/non-melodic discrimination). A monophonic pitch tracker, developed in the context of the SMSTools (Cano 1998), is used to establish a baseline performance for the evaluation.

Results clearly show that the method proposed by Paiva (ID 1) outperforms all other approaches for the two depicted evaluation metrics. When octave-errors are discarded, the performance of the other three approaches is almost the same. In all cases, proposed approaches outperform the baseline system by a considerable margin. The best average result (69.7% for system 1 using metric 2) indicates that there is plenty of room for improvement on the development of an automatic melody extraction algorithm.

Participant ID	**Paiva (1)**	**Tappert and Batke (2)**	**Poliner and Ellis (3)**	**Bello (4)**
Metric 1: Average pitch accuracy rate	75.25	39.73	50.95	48.99
Maximum	92.21	80.15	86.87	82.20
Minimum	35.99	7.64	12.47	0.00
Metric 2: Average pitch accuracy rate	75.83	56.11	52.12	56.89
Maximum	92.21	80.71	87.18	82.22
Minimum	37.78	32.62	16.30	20.38

Table 3: Performance statistics for the individual approaches (all values in %).

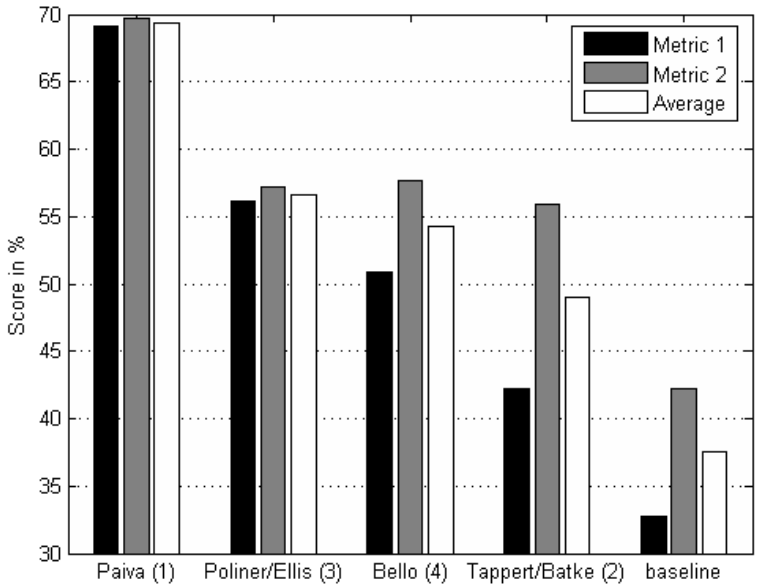

Figure 4: Summary of results for metrics 1 and 2.

5.3. About pitch estimation

Table 3 shows pitch estimation results in melodic frames only. It supports the observations made from Figure 4, on that approach 1 (Paiva) performs best at estimating pitch. Its relative performance increase between metrics is of 0.77%, compared to 2.30% for approach 3 (Poliner and Ellis), 16.13% for approach 4 (Bello) and 41.23% for approach 2 (Tappert and Batke). This shows clearly that approaches 1 and 3 are less prone to octave-related errors than the other approaches, as could be deduced from Figure 4 already. The susceptibility to octave errors appears as a weakness of signal-driven methods when compared to the auditory-driven approach implemented by Paiva, and to the machine learning approach of Poliner and Ellis. The latter approach is particularly interesting, as it demonstrates that a pure classification approach to melody extraction, that makes no assumptions about spectral structure, performs relatively well in comparison to traditional approaches that assume particular frequency structures for musical notes. However, for machine learning approaches the question of generalization arises. Figure 5 shows that the algorithm performed considerably better on the training set (64.9% on average) than on the previously unknown test set (50.35% on average), a relative decrease of 27.6%. This effect is not observed for the other methods.

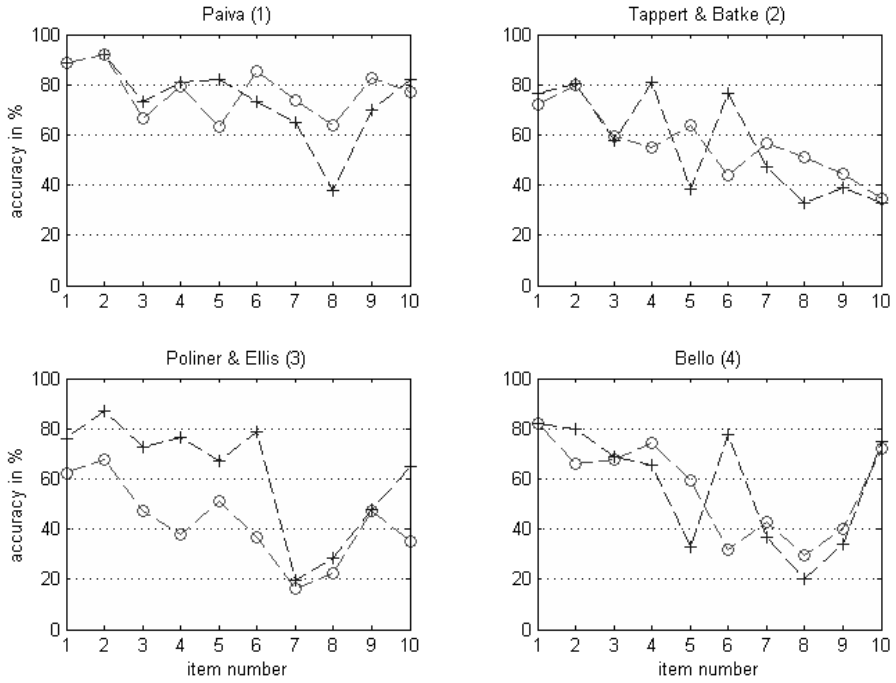

Figure 5: Accuracy in pitch estimation (metric 2, only melodic frames) for the training set (-+-) and the test set (-o-). Item numbers are listed in Table 2.

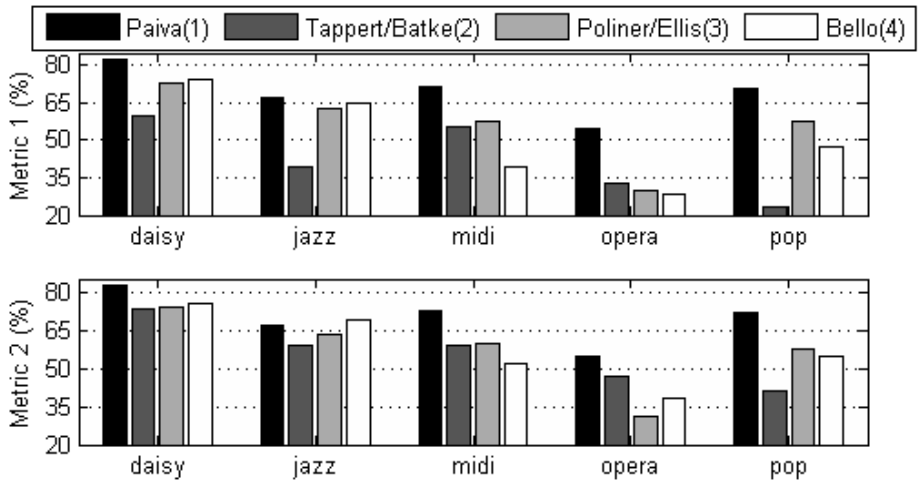

Figure 6: Metric 1 and 2 results averaged for each category of the evaluation material.

Another effect that can be observed here is that the pitch accuracy of approaches 3 (Poliner/Ellis) and 4 (Bello) has a large variation across the set. They seem better suited for certain musical styles than for other ones. However, the very limited size of the data set must be considered before drawing such

conclusions. Approach 2 (Tappert/Batke) shows a relatively greater consistency but it also stretches from 80% down to slightly above 30%. With the exception of only one track (opera_male2) approach 1 (Paiva) always yields a pitch accuracy above 60%, thus appearing to be the least dependent on the musical style in the quartet.

Figure 6 shows the evaluation results sorted by musical style of the evaluation material. We can observe that approach 1 (Paiva) is the one that performs the best in all categories. Only for jazz approach 4 and 3 are at the same level. Bello (4) even gets marginally better results in this category, when octave errors are not considered. We can also see that opera is the most difficult material, showing the worst results for all approaches except for approach 2 (Tappert/Batke) which performed slightly worse on the pop items. An explanation for the difficulty of the opera items is the complexity of the pitch envelop, which includes vibratos and other expressive characteristics (cp. Figure 7). From Figure 7 it can also be noticed that Paiva's algorithm can cope best with fast pitch variations. The precision in tracking the extreme vibrato (across 3 semitones) in this example is quite impressive. Bello's algorithm seems to be too strongly restricted by rules for smooth continuation in order to follow the fast pitch changes. Approach 2 (Tappert/Batke) shows a tendency to quantizing the output to stable frequency values when the pitch is varying heavily. For an annotation task this is not a disadvantage, but it might be problematic when the pitch needs to be tracked more precisely for example in order to obtain information about expressivity. Since approach 3 (Poliner/Ellis) only deals with pitch at semitone resolution it is no surprise that strong vibratos cause problems. However, the algorithm manages to roughly track the turning points.

On the other side, the Daisy category seems to be the easiest one to deal with. All approaches show the best results here. This can be due to the fact that it is a synthetic voice with simpler pitch envelopes (cp. Figure 8). Also the intensity relation between the voice and the background is higher than for the other items. For the first frames of the depicted excerpt it can be seen how approaches 1 and 2 are focussing on a voice of the accompaniment, while approaches 3 and 4 manage to detect the absence of the melodic voice.

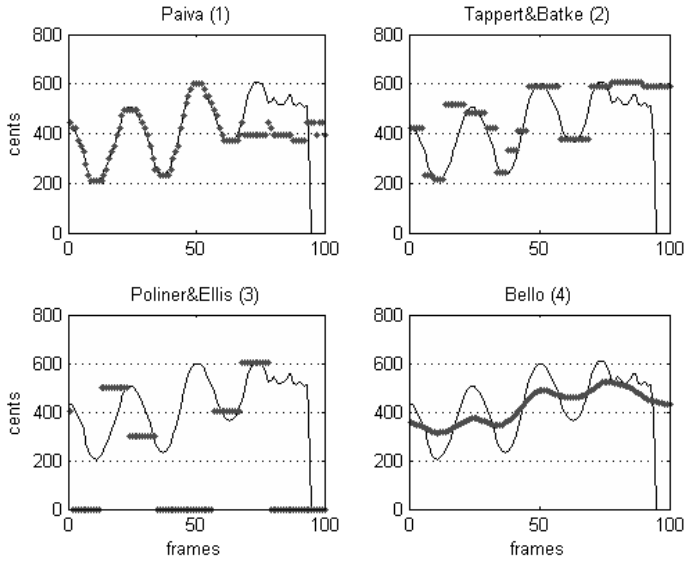

Figure 7: Short excerpt from female opera singing mapped into one octave (solid = reference pitch, dots = extracted pitch). Non-melodic frames are coded as 0 cents.

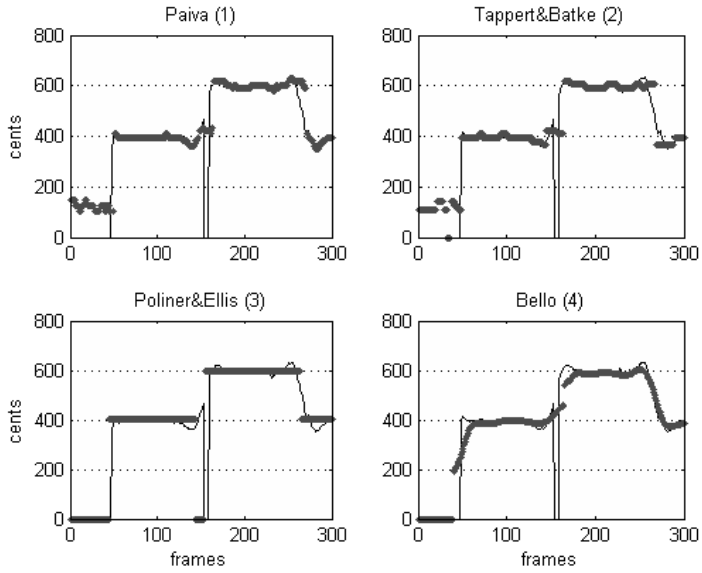

Figure 8: Short excerpt from synthetic singing (daisy) mapped into one octave (solid = reference pitch, dots = extracted pitch). Non-melodic frames are coded as 0 cents.

5.4. About melodic/non-melodic discrimination

An important aspect of the melody extraction task is the segregation of melodic information from the mixture. In these experiments, non-melodic frames account for a little subset of the test data, and

therefore they have limited impact on the results of the evaluation. However, a good discrimination can make the case for an approach to be useful in real-world applications, where the ratio between melodic and non-melodic frames may differ from the numbers in the test set.

Participant ID	Paiva (1)	Tappert and Batke (2)	Poliner and Ellis (3)	Bello (4)
Median of recall	32.05	55.59	88.25	63.67
Median of precision	48.59	62.73	25.37	40.73

Table 4: Individual recall and precision values for non-melodic frame detection (all values in %).

Table 4 shows the statistics for non-melodic frame detection, where recall gives the percentage of annotated non-melodic frames recognized as non-melodic, and precision gives the percentage of frames recognized as non-melodic being annotated as non-melodic. The median values were chosen to compensate for outliers due to the small number of non-melodic frames in some of the tracks. It shows that the machine learning approach of Poliner and Ellis has a strong tendency to classify dubious frames as non-melodic. This leads to by far the best recall rate among the methods under test. However, due to many false positives, the achieved precision is clearly the poorest. Bello's approach works surprisingly well for discrimination, considering that it uses basic autocorrelation-based measures for melodic/non-melodic discrimination deriving from monophonic signal processing. The auditory driven approach from Paiva (1) reveals a weakness here, which comes from the fact that the proposed algorithm does not tackle the melodic/non-melodic discrimination issue. In fact, the most salient notes in the defined allowed range are output, no matter if they belong to the melody or not. The approach by Tappert and Batke is showing the best precision while still obtaining a moderately high recall. This might be an indication for the advantage of using explicit tone models. Still, the results are far from an optimal solution.

5.5. About note segmentation

As mentioned before, for a number of applications it is useful to see the melody as a MIDI-like sequence of quantised notes. Segmenting the melody into notes also allows for the measuring of note attributes that can then be used for higher-level musical analysis of the signal. This is one of the less explored areas of

this research field, although it holds the potential to provide more musically-meaningful observations than the low-level feature-based analysis of the signal.

In our evaluation, metric 3 was intended to evaluate the ability of the proposed approaches to estimate note segments. Only two of the algorithms provided the specified output for the calculation of the edit distance. Results in Figure 9 show that Paiva's approach scored better than Bello's with an average of 8.63 over 14.12. Comparing the results for metric 1 and metric 3, we can see that in several cases (e.g. daisy1, daisy2, pop4) approach 4 outperforms approach 1 in terms of predominant frequency estimation, while still yielding a worse edit distance score. One reason for this is that Bello's algorithm tends to segment long notes into tremolo-like repetitions if parameters are not stable enough. In the edit distance metric these repetitions are treated (and punished) as wrong insertions, increasing the distance to the original. As mentioned in section 4.3 the interpretation of these distance values is difficult, since they are don't correspond one-to-one to human perception of melodic similarity. The interested reader might visit the web page (ISMIR Contest Definition Page 2004) and listen to the generated MIDI files based on the extracted melodies. This might help to get a subjective impression of the quality.

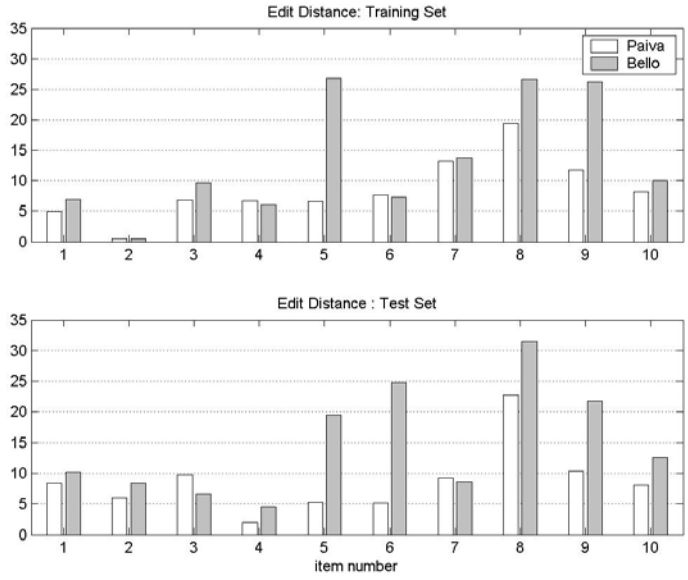

Figure 9: Metric 3 for the training set (top) and the test set (bottom) for Paiva's and Bello's approaches. Item numbers are listed in Table 2.

5.6. About computation time

We also measured roughly the computation time for each of the algorithms. This gives an idea of the computational complexity and the degree of code optimization for each approach. These statistics have only informative character and are not relevant for identifying a best-performance approach. Algorithms were computed in two machines: windows PC Pentium 1.2 GHz, 1 Gb RAM and Linux PC Pentium 2 GHz, 500 Mb RAM. Results, presented in table 5, indicate that the signal-driven algorithms, those of Tappert and Batke and Bello, were the fastest. Conversely, the approach by Poliner and Ellis is about 7 times slower than those, and Paiva's algorithm takes around 50 times as much time. This indicates the importance of an efficient front-end implementation, especially for auditory driven front-ends (a good example is (Klapuri and Astola 2002)).

Participant ID	**Paiva (1)**	**Tappert and Batke (2)**	**Poliner and Ellis (3)**	**Bello (4)**
Operating system	Windows (MATLAB)	Linux (MATLAB)	Linux	Linux (MATLAB)
Average Time Per Item t	3346,67	60,00	470,00	82,50

Table 5: Averaged computation time (in s) per item for the individual approaches.

Since this field of research is still struggling to reach results reliable enough to compare to a trained human, computation time is not yet a major issue. However, the tremendous discrepancy between the auditory driven approach and the other ones raises the question, whether future optimization will be able to overcome this disadvantage, thus making it attractive to use also with large music collections.

5.7. About the evaluation methodology

Manual annotations are a very imprecise and time-consuming process. Although the provided material is already quite valuable for evaluation, there is the need of increasing its quantity and quality. For instance, annotations should be validated by different people, something that requires a concerted effort from researchers in this field. The semi-automatic approach here proposed could offer a solution to the problem of annotating large quantities of music; however standard tools and methodologies for semi-automatic annotation need development, something that is proving difficult in the short term. Some relevant considerations on manual annotation are presented in (Lesaffre et al. 2004).

At the current level for the state-of-the-art in melody extraction (where there is still a lot of room for improvement), it seems beneficial to use different metrics in order to compare different aspects of the proposed approaches. The underlying concept of this comparison was a contest scenario where a winner was supposed to be identified. In order not to make it complex, the variety of explicitly tested properties was reduced to the most relevant ones. The three proposed metrics are biased towards a bottom-up perspective rather than a top-down way of thinking.

A way to improve on that is to develop better benchmarks for evaluations. It is undoubtedly difficult to find a representative selection of manageable size, given the copyright-free requirements of the test collection and their availability in multi-track form (if our semi-automatic annotation is to be used). An additional strategy is to compile a selection of special problem cases presenting gradual levels of difficulty.

6. Conclusions and further work

In this paper we have proposed a methodology for the evaluation of automatic melody extraction methods that operate on polyphonic and multi-instrumental music signals. Results in this study are obtained by using this methodology on a broad and interesting cross-section of melody extraction algorithms. Proposed approaches pose different solutions to the issues of predominant pitch estimation in polyphonic mixtures and the discrimination of melodic segments from the mixture.

Results indicate that, for the three used evaluation metrics, the approach proposed by Paiva outperformed the other proposed approaches. It shows better performance at estimating predominant pitches, less octave-related errors and better segmentation of the melody into notes. The system revealed a weakness in discriminating between melodic and non-melodic frames. The machine learning approach by Poliner and Ellis performed relatively well, despite not including sophisticated pre- or post-processing and the use of a quantized output (in time and frequency). However, it showed a marked difference in performance between the tuning and testing set, raising concerns about its generality.

Bello's and Tappert/Batke's approaches were the fastest, highlighting the advantages of using efficient signal analysis techniques for their operation, while Paiva's approach demonstrated to be extremely slow in its current implementation, raising concerns about the feasibility of its operation on large music collections. Approach 2 (Tappert/Batke) showed the best capability in discriminating melodic and non-melodic frames, although the results are still far from optimal. Its pitch estimation performance was less dependent on the music style than it was observed for approach 3 (Poliner/Ellis) and 4 (Bello), but it was also yielding slightly worse results and showed the highest tendency to octave-related errors. Bello's approach was capable of a quite good discrimination between melodic and non-melodic frames as well. The limitations of its autocorrelation-based front-end make it also susceptibility to octave-related errors in pitch detection, but not to the same extend as in case of approach 2. In fact, when octave errors are disregarded, the performance of systems 2 (Tappert/Batke), 3 (Poliner/Ellis), and 4 (Bello) is almost equal.

A fair comparison of different approaches to the same problem is essential to sustained research progress. Although there might be a lot of headroom for improvement in the procedure, this first attempt gives already useful information and important feedback. The publication of the entire testing material also enables researchers who did not participate to estimate their level of achievement in melody extraction, and provides a valuable evaluation material for our community.

Further steps towards adequate benchmarking have to be taken. One of them is the organization of another melody extraction evaluation exercise under the context of the ISMIR 2005 conference. Previous efforts have already secured a larger dataset for the evaluation, while interest has been gathered from a larger number of participants.

7. Acknowledgments

The authors would like to thank Maarten Grachten for his contribution on the algorithm for computing melodic similarity, and people from the MTG for their contributions to the evaluation material. This work was partially funded by the European Commission through the SIMAC project IST-FP6-507142.

8. References

ISMIR Contest Definition Page, Melody Extraction Category 2004.

http://ismir2004.ismir.net/melody_contest/results.html

Batke, J. M., G. Eisenberg, P. Weishaupt and T. Sikora. 2004. A query by humming system using MPEG-7 descriptors. In Proceedings of the 116th AES Convention, Berlin, May.

Bello, J.P. 2003. Towards the automated analysis of simple polyphonic music: A knowledge-based approach. Ph.D. Thesis. University of London, London, UK.

Bregman, A. S. 1990. Auditory Scene Analysis: The Perceptual Organization of Sound, MIT Press.

Brown, J.C. and B. Zhang. 1991. Musical Frequency Tracking using the Methods of Conventional and 'Narrowed' Autocorrelation. Journal of the Acoustic Society of America, 89, pp. 2346-2354.

Cano, P. 1998. Fundamental Frequency Estimation in the SMS analysis. Proceedings of COST G6 Conference on Digital Audio Effects, Barcelona.

Dixon, S. 2000. On the Computer Recognition of Solo Piano Music. MikroPolyphonie 6: Special Issue on ACMC 2000. Sept. 2000 – March 2001.

Eggink, J. and Brown, G.J. 2004. Extracting melody lines from complex audio. Proc. International Conference on Music Information Retrieval, ISMIR'04 pp. 84-91.

Fernández-Cid, P. and F. J. Casajús-Quirós. 1998. Multi-pitch estimation for polyphonic musical signals. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3565¬-3568.

Gómez, E., A. Klapuri and B. Meudic. 2003. Melody Description and Extraction in the Context of Music Content Processing. Journal of New Music Research, 32(1).

Goto, M. 2000. A robust predominant-f0 estimation method for real-time detection of melody and bass lines in CD recordings. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.

Grachten, M., J. Ll. Arcos and R. López de Mántaras. 2002. A Comparison of Different Approaches to Melodic Similarity. In Proceedings of the International Conference on Music and Artifical Intelligence.

Kaernbach, C. and L. Demany. 1998. Psychophysical evidence against the autocorrelation theory of pitch perception. Journal of the Acoustic Society ofAmerica, 104, pp. 2298-2306.

Klapuri, A. 2000. Qualitative and quantitative aspects in the design of periodicity estimation algorithms. In Proceedings of the European Signal Processing Conference.

Klapuri, A. and J. T. Astola. 2002. Efficient calculation of a physiologically-motivated representation for sound. In Proceedings of IEEE International Conference on Digital Signal Processing, Santorini, Greece.

Klapuri, A. 2004. Signal Processing Methods for the Automatic Transcription of Music. PhD Thesis, Tampere University of Technology.

Kashino, K., T. Kinoshita and H. Tanaka. 1995. Organization of hierarchical perceptual sounds: music scene analysis with autonomous processing modules and a quantitative information integration mechanism. In Proceedings of the International Joint Conference On Artificial Intelligence.

Lesaffre, M., M. Leman, B. De Baets and J. P. Martens. 2004. Methodological Considerations Concerning Manual Annotation Of Musical Audio In Function Of Algorithm Development. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), Barcelona, Spain.

Marolt, M. 2004. On Finding Melodic Lines in Audio Recordings. In Proceedings of DAFx, Naples, Italy.

Mc Adams, S. 1994. Audition: physiologie, perception et cognition. In M. Richelle, J. Requin & M. Robert (eds.), Traité de psychologie expérimentale, Presses Universitaires de France, Paris, pp. 283-344.

McAulay, R. J. and T.F. Quatieri. 1986. Speech Analysis/Synthesis based on a Sinusoidal Representation. IEEE Transactions on Acoustics, Speech and Signal Processing, 34(4), pp. 744-754.

Moon, T. K. 1996. The expectation-maximization algorithm. IEEE Signal Processing Magazine, pp. 47--60.

Nettheim, N. 1992. On the spectral analysis of melody. Journal of New Music Research, 21, pp. 135-148.

Paiva, R. P., T. Mendes and A. Cardoso. 2004. A Methodology for Detection of Melody in Polyphonic Musical Signals, In Proceedings of the 116th AES Convention.

Paiva, R. P., T. Mendes, and A. Cardoso. 2004. Exploiting Melodic Smoothness for Melody Detection in Polyphonic Audio, Technical Report.

Paiva, R. P., T. Mendes, and A. Cardoso. 2005. An Auditory Model Based Approach for Melody Detection in Polyphonic Musical Recordings. U. K. Wiil (ed.) Computer Music Modelling and Retrieval – CMMR 2004, Lecture Notes in Computer Science, vol. 3310.

Scheirer, E. D. 2000. Music listening systems. PhD Thesis, MIT.

Selfridge-Field, E. 1998. Conceptual and Representational Issues in Melodic Comparison. In Melodic Similarity: Concepts, Procedures, and Applications. MIT Press, Cambridge, Massachusetts.

Serra, X. 1997. Musical Sound Modeling with Sinusoids plus Noise. In Roads, Pope, Picialli and De Poli (eds.), Musical Signal Processing.

Slaney, M. and R. F. Lyon. 1993. On the Importance of Time – A Temporal Representation of Sound. In: Cooke, Beet and Crawford (eds.), Visual Representations of Speech Signals.

Talkin, D. 1995. Robust algorithm for pitch tracking. In Speech Coding and Synthesis, Kleijn, W. B and Paliwal, K. K (eds.), Elsevier Science B. V.

Walmsley, P. J., S. J. Godsill and P. J. Rayner. 1999. Bayesian graphical models for polyphonic pitch tracking. In Proceedings of the Diderot Forum, Vienna.

Walmsley, P. J., S. J. Godsill and P. J. Rayner. 1999. Polyphonic pitch tracking using joint bayesian estimation of multiple frame parameters. In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.

Wiegrebe, L., R. D. Patterson, L. Demany and R. P. Carlyon. 1998. Temporal dynamics of pitch strength in regular interval noises. Journal of the Acoustic Society of America, 104, pp. 2307-2313.

Witten, I.H. and E. Frank. 1999. Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann.

Yost, W. A. 1996. Pitch strength of iterated rippled noise. Journal of the Acoustic Society of America, 100, pp. 3329-3335.