AUTOMATIC POLYPHONIC PIANO NOTE EXTRACTION USING FUZZY LOGIC IN A BLACKBOARD SYSTEM

Giuliano Monti, Mark Sandler

Department of Signal Processing
Queen Mary University of London
giuliano.monti@elec.qmul.ac.uk

ABSTRACT

This paper presents a piano transcription system that transforms audio into MIDI format. Human knowledge and psychoacoustic models are implemented in a blackboard architecture, which allows the adding of knowledge with a top-down approach. The analysis is adapted to the information acquired. This technique is referred to as a prediction-driven approach, and it attempts to simulate the adaptation and prediction process taking place in human auditory perception. In this paper we describe the implementation of Polyphonic Note Recognition using a Fuzzy Inference System (FIS) as part of the Knowledge sources in a Blackboard system. The performance of the transcription system shows how polyphonic music transcription is still an unsolved problem, with a success of 45% according to the Dixon formula. However if we consider only the transcribed notes the success increases to 74%. Moreover, the results obtained in the paper presented in [1], show how the transcription can be used with success in a retrieval system, encouraging the authors to develop this technique for more accurate transcription results.

1. BACKGROUND

Systems for automatic transcription have been studied since 1975. In the early years Moorer [2] implemented a system for the polyphonic transcription of music played by two monophonic instruments. Limitations on the note range and the overlapping of the two instruments were required in order to perform the task successfully. These limitations were tackled in future systems, adding further knowledge in the transcription. Bregman's publication opened up to researchers a completely new field to investigate, and suitable architectures to implement psychoacoustic rules have been employed effectively. Psychoacoustics and Artificial Intelligence (AI) were merged to achieve a better understanding and solution to music transcription and in general of Auditory Scene Analysis, which became Computational Auditory Scene Analysis (CASA). Blackboard systems [3] and Integrated Processing and Understanding of Signals(IPUS) [4] along with Multi-Agent architectures [5] are currently widely employed in the solution of musical problems.

2. FFT FRONT-END AND PSYCHOACOUSTIC MASKING

The front-end extracts from audio the basic features that will be interpreted by the system to write the final score. Our front-end uses a psychoacoustic model to select the 'important' peaks in the frequency domain representation given by the Fourier Transform. The output of the front-end is a set of Spectal Peaks' Amplitude and Frequency parameters.

2.1. Power Spectrum

The audio samples are normalised at the beginning of the analysis to have 1 as the maximum absolute amplitude value. This signal, $x(n)$, is re-sampled after anti-aliasing filtering at 11025 Hz and an FFT of 2048 points (185 msec) is calculated leading to a frequency resolution of circa 5.4 Hz in the spectrum. The hop size between two consecutive frames is of 256 samples (23.2 msec) to improve the time resolution, compromised by the choice of large windows. The coefficients are multiplied by a normalisation coefficient equal to $2/N$, where N is the FFT length in samples. The power spectra distrubution P(k) is obtained using the following formula:

$$P(k) \;=\; PN + 10log_{10}\left|\frac{2}{N}\sum_{n=0}^{N-1} w(n)x(n)e^{-j\,\frac{2\pi kn}{N}}\right| \quad (1)$$
$$0 <= k <= \frac{N}{2}$$

where the power normalisation term, PN, is fixed at 90 dB and the Hann window, $w(n)$, is defined as:

$$w(n) = \frac{1}{2}\left[1 - \cos(\frac{2\pi n}{N})\right] \quad (2)$$

The choice of PN leads to a Sound Pressure Level SPL of 84 dB, when analysing a full-scale sinusoid. The SPL is low-bounded at -15 dB for very low amplitude input tones.

2.2. Global Masking Threshold

As part of every front-end there is a peak picking algorithm, which selects the most important information to process in one frame of the signal. Simple thresholding has been widely used in common peak-picking. Although, this approach is not optimal because of the human non linear perception of loudness at different frequencies. For this porpose, psychoacoustic experiments led to the study of models for a psychoacoustic weighting of the signals. In this way, the signal components in the frequency domain assume importance according to their psychoacoustic relevance. In our work, the ISO MPEG-1 [6] psychoacoustic model of masking has been extended to work with large time windows. Some simplifications of the algorithm were also necessary for computational efficiency. The absolute threshold of hearing is characterised by the amount of energy needed in a pure tone such that it can be detected by a listener in a noiseless environment. The frequency dependence of this threshold was quantified in 1940, when Fletcher reported test results for a range of listeners which were generated in a study of typical American hearing acuity. The quiet threshold is well approximated by the non-linear function of equation 3 which is representative of a young listener with acute hearing.

$$\begin{aligned} T_q(f) \;=\; & 3.64(f/1000)^{-0.8} - \\ & - 6.5e^{-0.6(f/1000-3.3)^2} + \\ & + 10^{-3}(f/1000)^4 \quad \text{(dB SPL)} \end{aligned} \qquad (3)$$

When one sound is rendered inaudible because of the presence of another sound, masking occurs. Simultaneous masking refers to a frequency-domain phenomenon which has been observed within critical bands. In this domain we distinguish two types of simultaneous masking, namely tone-masking-noise and noise-masking-tone. Masking also occurs in the time-domain. Sharp signal transients create pre- and post- masking regions in time during which a listener will not perceive signals beneath the elevated audibility thresholds produced by a masker. We decided to calculate the masking threshold considering all the maskers as tones-masking-noise, and we didn't take into account time-masking. This is due to the fact that our model is principally oriented to the signal's tonal analysis in its stationarity.

Since masking refers to a psychoacoustic phenomenon, the masking threshold will be calculated in the Bark domain. The Bark scale, in fact, refers to the critical bands of hearing. The conversion from frequency to bark is given by equation 4 and its function can be approximated by a logarithmic function.

$$Bark(f) = 13\arctan(0.00076f) + 3.5\arctan((f/7500)^2) \qquad (4)$$

From the PSD of equation 2 we detect all the local maxima, then we replace any two maxima in a 0.5 Bark sliding window by the stronger of the two. Once the Maskers are calculated, a decimation process takes place before calculating the global masking threshold according to the following scheme:

$$i = \begin{cases} 4 + k - (kmod8) & 1 <= k <= nFFT/8 \\ 4 + k - (kmod32) & k > nFFT/8 \end{cases} \qquad (5)$$

where k is the FFT index and i the decimation index. Decimation increases with frequency, because the critical bandwidth increases and the accuracy requirement is less demanding. The effect of the decimation scheme above is to reduce the number of bins for the calculation of the global masking threshold, without loss of maskers. In fact, using long FFT frames, i.e. 2048 at 11025Hz the computational load becomes heavy and data decimation halves the processing time. The maskers are then relocated according to the decimation scheme.

Having obtained a decimated set of maskers, each threshold represents a masking contribution at frequency bin i due to the masker. The Tonal Masker Thresholds, according to the ISO MPEG-1, $T_{TM}(i,j)$, are given by

$$T_{TM}(i,j) = P_{TM}(j) - 0.275z(j) + SF(i,j) - 6.025 \quad \text{(dB SPL)} \qquad (6)$$

where $P_{TM}(j)$ denotes the SPL of the tonal masker in frequency bin j, $z(j)$ denotes the Bark frequency of bin j, and the spread of masking from masker bin j to maskee bin i, $SF(i,j)$, is modeled by equation 7. $SF(i,j)$ is a piecewise linear function of masker, $P_{TM}(j)$, and Bark maskee-masker separation, $\Delta_z = z(i) - z(j)$. $SF(i,j)$ approximates the basilar spreading of the masking threshold given a certain excitation.

$$SF(i,j) = \qquad (7)$$

$$\begin{cases} 17\Delta_z - 0.4P_{TM}(j) + 11 & -3 \le \Delta_z \le -1 \\ -17\Delta_z & -1 \le \Delta_z \le 0 \\ (0.4P_{TM}(j) + 6)\Delta_z & 0 \le \Delta_z \le 1 \\ (0.15P_{TM}(j) - 17)\Delta_z - 0.15P_{TM}(j) & 1 \le \Delta_z \le 9 \end{cases}$$

(dB SPL)

Once the individual masker function are obtained, the global threshold is calculated by combining them for each frequency bin. The model assumes that masking effects are additive. The global masking threshold, $T_g(i)$ is therefore obtained by computing the sum

$$T_g(i) = 10\log_{10}(10^{0.1T_q(i)} + \sum_{j=1}^{L} 10^{0.1T_{TM}(i,j)})(dBSPL) \qquad (8)$$

where $T_q(i)$ is the absolute threshold of hearing for frequency bin i, and $T_{TM}(i,j)$ are the individual masking thresholds modelled as tonal, with L the number of maskers. In other words, the global threshold for each frequency bin represents a signal-dependent, power additive modification of the absolute threshold due to the basilar spread of all maskers in the signal power spectrum.

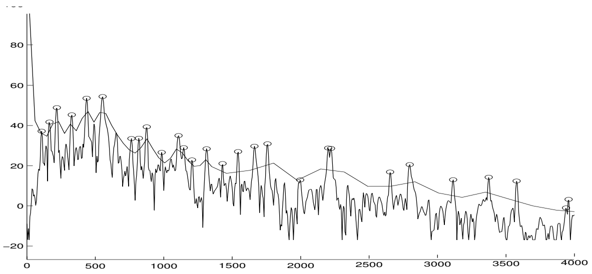

Figure 1: *Masking level and peak picking in the spectrum. x axis:frequency(Hz), y axis: SPL (dB)*

2.3. Peaks, Instantaneous Frequencies and Amplitudes

The frequency resolution of the FFT can be calculated by:

$$resolution = \frac{Fs}{N} \qquad (9)$$

where Fs is the sampling frequency and N the FFT frame size. In polyphonic music we need to isolate the spectral components of different signals. If we consider two notes, whose fundamental frequencies are 50 Hz apart, we need a resolution that will allow us to have at least one point between the two frequencies, yielding to a resolution better than 25 Hz choosing $N = 2048$. We decided to sacrifice time resolution for good note separation in the frequency domain. After this premise, every peak in the spectrum will be an approximated value of the real frequency. The incertainity is equivalent to the resolution. Sometimes, for low notes, this resolution is not enough to determine the exact pitch with the fundamental frequency. Using the phase unwrapping method from the phase vocoder technique we calculate the istantaneous spectral frequency for each spectral peak.

2.3.1. *Phase unwraping*

In order to find the istantaneous frequency, we interpolate the phases of the peak bins in two consecutive frames. The FFT outputs frequencies are quantised to the centre frequency of the filterbank channels. If we consider the FFT bin k of one peak selected in the spectrum, we calculate the phase difference:

$$\Delta\phi = \phi_i(t_{n+1}) - \phi_i(t_n) + 2m\pi = \Delta t\omega_i(t_n) + 2m\pi \qquad (10)$$

where $\omega_{i(t_n)}$ is the instantaneous frequency remained constant over the duration $\Delta t = (t_{n+1} - t_n)$, which is equal to the hop size. The term $2m\pi$ comes from the fact that only the principal determination of the phase is known. Parameter m is calculated by solving the following inequity:

$$|\Delta\phi - \Omega_k\Delta t - 2m\pi| < \pi \qquad (11)$$

and there is only one integer m that satisfies inequity of equation 11. Once m is determined by adding or subtracting multiples of 2π until the preceding inequality is satisfied, this is the process of phase unwrapping [7], the instantaneous frequency can be obtained as follow:

$$\omega_i(t_n) = \Omega_k + \frac{1}{\Delta t}(\Delta\phi - \Omega_k\Delta t - 2m\pi) \qquad (12)$$

Since we calculate the FFT using Han window, once the instantaneous frequency is calculated, the instantaneous amplitude is also calculated.

2.3.2. *Amplitude Value Correction*

In order to have precise values for both frequencies and amplitudes, we unwrapped the phases of each frequency bin of the FFT to find the istantaneous frequency. Once we have found the istantaneous frequency, we can calculate the correction for the amplitude value corresponding to a certain bin. The FFT spectrum is the convolution of the signal spectrum with the Hann window spectrum. The plot of the Hann main lobe is portrayed in figure 2. Knowing the shape of the Hann main lobe and the istantaneous frequency, it is possible to correct the amplitude value given by the FFT. We calculate the frequency $deviation$ as a fraction of a bin with the following equation:

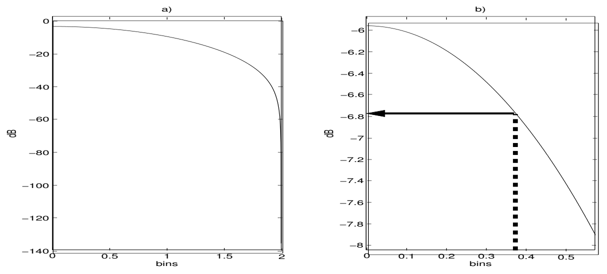

Figure 2: *a) Hann main lobe b) Zoom of Hann main lobe: the value of the amplitude's correction is found from the frequency deviation in bins*

$$deviation = \left|\frac{f_{ist} - f_{bin}}{df}\right| \qquad (13)$$

where $f_{ist} = \frac{\omega_{i(t_n)}}{2\pi}$ is the instantaneous frequency, f_{bin} the frequency corresponding to the FFT bin, and df is the frequency resolution given by equation 9. The frequency deviation value is in the range [0,0.5]. The instantaneous Amplitude value is given by calculating:

$$A_{ist} = A_{bin} + Hann(0) - Hann(deviation) \qquad (14)$$

where A_{ist} and A_{bin} are the istantaneous and bin amplitude value; $Hann(0)$ and $Hann(deviation)$ are the Hann window spectral amplitude values at the centre of the bin and corresponding to the frequency deviation.
At this point the set of spectral peaks instantaneous amplitudes and frequencies are passed to the blackboard system for the calculation of the score.

3. BLACKBOARD MODEL

This metaphor has been used to describe the work of experts trying to solve a problem in front of a physical blackboard. The original data is written on the blackboard. Then each expert contributes according to his knowledge and hypotheses are produced for the possible solution. All the information and hypotheses are written on the blackboard. The experts are represented by the Knowledge Sources (KSs), which operate to modify the blackboard data until a signal explanation is found. The hypotheses, which have good support in all KSs, are then confirmed as the final interpretation. The blackboard architecture also needs a Scheduler, which is the control unit. The scheduler decides which of the KSs must operate relying on the information written on the blackboard. Figure 3 illustrates the Blackboard system implemented in this paper.

3.1. Blackboard Data Abstraction

The Blackboard workspace is arranged in a hierarchy of data abstraction levels. The first level is represented by the Spectrogram Buffer, which stores two seconds of the most recent spectra. The Spectral Peaks represent the basic information to build the Blackboard objects. Harmonicaly related Spectral Peaks are grouped into Note Candidates. If a Note Candidate receives a good rating, it is transformed in a Note Hypothesis. Then the hypotheses that last for a minimum activation time become Active Notes. The Blackboard Data space is available to any active KS and represents the state of the system.

3.2. Scheduler and KSs

The Scheduler decides which KS has to be activated, depending on the state of he Blackboard. We grouped the front-end with the other KSs to fit the model. The front-end is called by the Scheduler at the beginning of each new frame and provides the FFTs coefficients, stored in the circular Spectrogram Buffer, and the Spectral Peaks. Figure 4 illustrates the processing stages imposed by the Scheduler, that we are going to describe in the next sections.

3.2.1. *Active Notes Streaming*

After the Spectral Peaks are produced by the front-end, the scheduler looks in the Blackboard for Active Notes. The Active Notes receive particular attention by the system, because they come from Hypotheses that were confirmed a number of times and therefore ‘to be trusted’. Often in polyphony the Notes are played, whilst

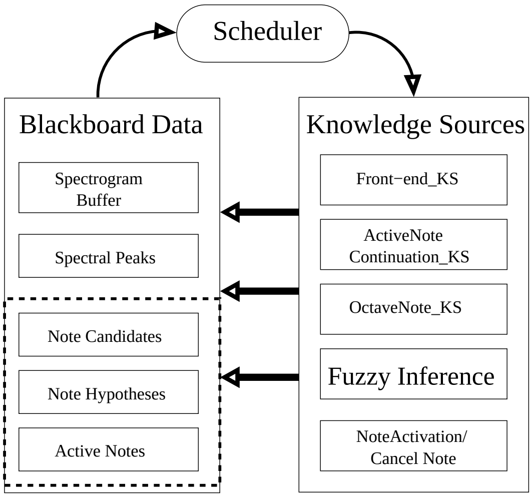

Figure 3: *Blackboard system*

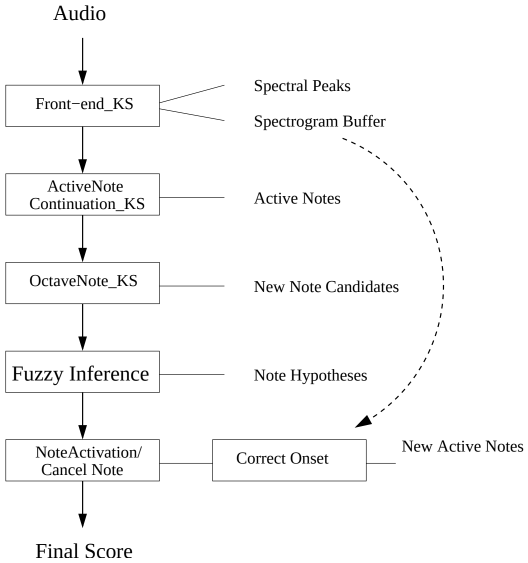

Figure 4: *Processing flow implemented in the Scheduler*

other keep ringing. A frame by frame multi-pitch tracker will give different results depending on the relative energy of the Notes. We want to exploit the beginning part, where the note intensity and pitch are detectable in a mixture of sounds. Then, we follow the note ringing, by assuming its continuation. This concept of 'streaming' is implemented in the ActiveNoteContinuation_KS. This KS checks if the Active Note fundamental frequency is among the Spectral Peaks. In that case extract from the Spectral Peaks all the frequency and amplitude belonging to the harmonic support of the Active Note and prevents them to generate new Note Candidates. After all the Active Notes have been processed for continuation, the scheduler calls the OctaveNote_KS.

3.2.2. Octave Note Detection

By Octave Notes we mean all the Notes, whose fundamental frequency corresponds to a multiple of another Note's fundamental, even if the interval is not strictly of an octave. This means that the two spectra are ideally (not considering inharmonicity) completely overlapped. If $f_2 = nf_1$ then each partial of f_2 overlaps every n^{th} partial of f_1. In our system we don't detect any Octave Notes that are played simultaneously. This assumption is made due to the fact that this kind of octave detection must rely on instrument tone modeling. Rather, we preferred to detect Octave Notes played with a different onset time, but still overlapping in frequency, which are also easier to distinguish for humans than the synchronous case. The OctaveNote_KS analyses the Active Notes partial's amplitudes. When the partial's amplitude exceeds a given threshold a new Note Candidate is written on the blackboard. From the analysis of the Active Notes, many partials may increase in amplitude, caused by the playing of another note. In this case, we eliminate all the Candidates whose frequency is multiple of other Candidates. The partial's onset threshold is increasing with frequency, because at the high frequency the signal is very noisy and there are good chances to make errors. This KS also is responsable for the detection of repeated Notes. An adaptive thresholding is implemented, in this case. The maximum amplitude of the Active Note's fundamental is stored in the Blackboard and a proportional threshold about (80/90%) is set. When the fundamental's amplitude crosses that threshold again a new Note Candidate is written on the blackboard.

3.2.3. Note Activation/Offset Note

The Fuzzy Inference System [8], that will be described in section 3.3, creates and update the Hypothesis on the Blackboard. The Note Activation/Offset Note KS looks at the Hypothesis in the Blackboard and transform them into Active Notes. A Note Hypothesis becomes an Active Note, when it lasts for a minimum activation time (80 msec). When a Note is activated, the correct onset time is found looking in the Spectrogram Buffer for the most recent rise in the fundamental's amplitude. The Note Hypotheses that have similar onset time are checked for Octaves and eventually deleted from the Blackboard. This KS deletes also from the Blackboard the Active Note or Hypothesis that weren't confirmed in the last frames, after writing the Active Notes on the final score with their pitch, onset and duration.

3.3. Fuzzy Inference System KS

The Fuzzy Inference System (FIS) KS take the Spectral Peaks that weren't selected in the Note Continuation process and creates new Candidates. The algorithm chooses the lowest frequency and build a vector of the partials amplitudes and frequencies collected from the Spectral Peaks. The new Candidate is evaluated by the FIS to become a Note Hypothesis. If the Candidates fails the fundamental frequency is deleted from the Spectral Peaks, while the partials are returned for the choice of the next Candidate's fundamental. If the Candidates becomes a Note Hypothesis all the partials are excluded from being possible fundamentals, still they can contribute to other notes rating. With this statement we avoid penalising notes that share many partials(like a note with its fifth).
To rate a Note Candidate the inference system analyses a set of 3 features, x_1, x_2, x_3, extracted from the Note Candidate's vector.

- x_1 is the fundamental of the note.

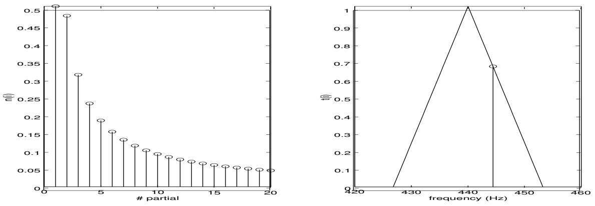

Figure 5: *a) n_i as a function of the partial number b) t_i calculated as the intersection point between the frequency detected and the triangole of one semitone centered on the ideal multiple of f*

- x_2 is the harmonic rate
- x_3 is the difference between the maximum peak in the spectrum and the Candidate fundamental's energy

3.3.1. Feature 1: Fundamental Frequency

This feature is the pitch of the Note under analysis and indicates if we are analysing a low, middle or high Note. Depending on this quantity we expect different characteristics in the note's spectrum. We chose the boundaries between low and middle and high notes at around 180 and 700 Hz. This can be visualised from the membership functions in the first column of figure 6.

3.3.2. Feature 2: Harmonic Rate

The Harmonic Rate is given by the equation15:

$$\Lambda(f) = \sum_{i=1}^{N} t_i n_i \qquad (15)$$

where n_i rates the presence of the ith partial. n_i decreases from $i = 1$, the fundamental, to the highest partial as shown in figure 5 a). The presence of peaks at or near multiples of f increases the harmonic rate $\Lambda(f)$ in a way which depends also on the peak's frequency position. t_i depends on how closely the ith peak is tuned to a multiple of f and is calculated by equation 16:

$$t_i = abs(f_i - p_i)/p_i * 0.03 \qquad (16)$$

where p_i is the ideal partial position, $p_i * 0.03$ is the approximation of a semitone interval and f_i the nearest peak to p_i inside the semitone window. The value of t_i is between 0 and 1 as can be seen from figure 5 b). n_i depends on whether the peak is closest to a low or high multiple of f. The coefficient n_i expresses the importance of the partial in determining the final likelihood and is given by the heuristic function [9]:

$$n_i = \begin{cases} 0.5 & i = 1 \\ \frac{0.9}{i-0.1} & i = 2..N \end{cases} \qquad (17)$$

It can be noticed that the value given at the fundamental position is very similar to the value given at the first harmonic. This heuristic rating has proved effective when the fundamental was missing. Typical values for $\Lambda(f)$ are around 2 for Low Notes, 1.5 for Middle Notes, and 0.7 for High Notes. This is taken into account when building the membership functions in coloumn two of figure 6. $\Lambda(f)$ increases with the quantity of partials found in the spectrum and doesn't depend on the partial's amplitude. We desired this assumption in order to have a parameter not depending on a particular timbre or instrument.

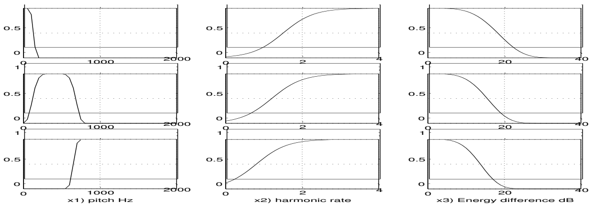

Figure 6: *Membership functions: columns refer to pitch, harmonic rate, relative energy (starting from left). row refer to Low, Middle, High Notes (starting from top)*

3.3.3. Feature 3: Fundamental's Relative Energy

This feature is calculated as the difference between Spectral Peaks maximum and the Fundamental's Energy. In the FIS the relative energy is determinant for High pitched notes, which require a high energy at the fundamental to receive a good rate and become Note Hypothesis. This feature becomes gradually less important as the pitch decreases, until considering the extreme case of the completely missing fundamental. In this case the Energy of the fundamental, which cannot be calculated from the spectral peaks, is chosen from the original spectrum. The membership function in column three of figure 6 shows the Energy rating for Low, Middle and High Notes.

3.3.4. Linguistic Decision Logic

The FIS rates each feature according to the membership functions shown in figure 6. From the top each row refers to the membership functions for Low, Middle and High Notes. Each feature is passed to the membership functions in the correspondent column to output a rate between 0 and 1. The total rate for each row is calculated choosing the minimum rate across that row. The 'min' operator, is a quantitative implementation of the logical 'and'. The output rate is not exploited in this first implementation, but only thresholded to give a yes/no response. The Linguistic rules implemented in the system decide if a group of partial is a good candidate. We distinguish tree kind of notes: for Low-Notes Candidates we expect high harmonic support(many partials) 'and' medium-high energy in the fundamental; for High-Note Candidates, we require a high Energy value of the fundamental with respect to the other spectral peaks 'and' we expect low harmonic rate; for Mid-Note Candidates we interpolate the two criteria. Therefore, we designed the sigmoid membership functions taking into account the average values for each feature in every category of Notes.

4. RESULTS

The polyphonic system has been tested with 14 piano pieces by several composers. The audio files were recorded at the LMA in Marseille, using a Disclavier Yamaha MIDI controller for a Conservatory C6 Yamaha Grand Piano. The original MIDI files were downloaded from various internet websites. Music was from composers such Beethoven, Debussy, Joplin, Mozart, Rachmaninoff,

Ravel and Scarlatti. We used a Macintosh G4 computer with 256 Mbyte of RAM and 40 Gbyte of Hard Disk, running the Pro Tools LE software on Mac OS 9.1. The audio and MIDI interface was a Digidesign DIGI-001 system, which provided the MIDI output for the Disclavier and the audio inputs for the two AKG-C1000S condenser microphones. The recording quality was set to stereo CD quality (16 bit at 44100 Hz). The Matlab code runs on mono recordings at 11025 Hz of sampling rate. For the conversion of the audio files we used the re-sample function with anti-aliasing filter in Sound Forge audio editor, running on a PC.
Figure7 shows the performance of the transcription algorithm. The parameters plotted are:

N Notes: number of original notes inside the window represents the polyphony

Nt Note transcribed: number of correctly transcribed notes

FP False Positive: number of transcribed notes that weren't played in the original MIDI

FN False Negative: number of not transcribed notes

OctP Octave Positive: Number of FP notes that are one octave above a note in the original MIDI

OctN Octave Negative: Number of FN notes which fundamental is a multiple of an other note in the transcribed score

Then, we evaluate the performance using the Dixon formula [10] in equation 18.

$$Perc1 = 100 * \frac{Nt}{Nt + FP + FN} \tag{18}$$

With this formula the detection success is 45%. Although the result might seem poor, which is not considering the polyphonic problem and the complexity of the original files, we can calculate the value of the ratio of correctly detected note against the total transcribed i.e.:

$$Perc2 = 100 * \frac{Nt}{Nt + FP} \tag{19}$$

In this case the detection rate rises at 74%. An application, emphasising the importance of the results in this paper, is the Music Information Retrieval system implemented in [1]. For the first time, a system that retrieved polyphonic scores from polyphonic audio queries has been realised successfully. The audio queries were converted in score format and searched in a 3000 music score database. The testing has been performed on the trascriptions of the 48 Bach's Fugues and Preludes. The system reported and average ranking of a little over 3 for the Bach's Prelude and a little over 2 for the Bach's Fugues, where random ranking would place the known item on average 1.500^{th}. The success of the system can be attributed to the high rate of good detection rate calculated in equation 19. The correctness of the transcribed notes has been proven to be sufficiently accurate for the harmonic modeling implemented in the that system.

5. ACKNOWLEDGMENTS

This work is supported by OMRAS (Online Music Retrieval And Search), jointly funded by JISC (UK) and NSF (USA). Thanks to Ali Taylan Cemgil for the Matlab MIDI Toolkit software [11], which has been used to write the output MIDI files.

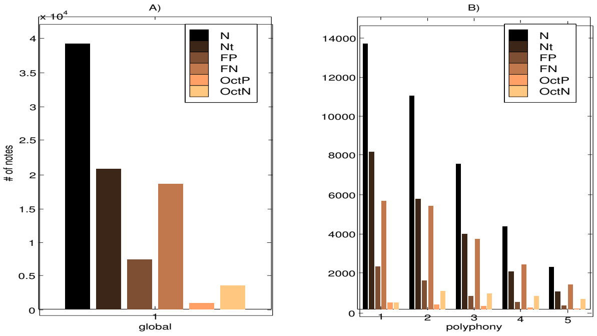

Figure 7: *Polyphonic results. A) global results, from left to right N, Nt, FP, FN, OctP, OctN B) partial results for polyphony 1 to 5*

6. REFERENCES

[1] J. Pickens J.P. Bello T. Crawford M. Dovey G. Monti M. Sandler, "Polyphonic Score Retrieval Using Polyphonic Audio Queries: A Harmonic Modeling Approach," *Proceedings of ISMIR 2002, Paris, France*, Oct. 2002.

[2] J. A. Moorer, "On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer," in *Available as Stanford University Department of Music Technical Report STAN-M-3*, Dept. of Computer Science, Stanford University, 1975.

[3] D.P.W. Ellis, *Prediction-driven computational auditory scene analysis*, Dept. of Electrical Engineering and Computer Science, MIT Media Laboratory, MA, USA, 1996.

[4] F. Klassner, V. Lesser, and H. Nawab, "The IPUS blackboard architecture as a framework for computational auditory scene analysis," in *Proc. of the Computational Auditory Scene Analysis Workshop, 1995 International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995.

[5] S.Hayamizu M. Goto, "A Real-Time Scene DescriptorSystem: Detecting Melody and Bass lines in Audio Signals," in *Proceedings of IJCAI Workshop on Computational Auditory Scene Analysis, August*, 1999, pp. 31–40.

[6] T. Painter, *A Review of Algorithms for Perceptual Coding of Digital Audio Signals*, available at http://www.mp3-tech.org/programmer/docs/, Dept. of Electrical Engineering, Telecommunications Research Center, Arizona State University, Tempe, Arizona.

[7] K. Brandenburg M. Kahrs, *Applications of signal processing to audio and acoustics*, Kluwer Academic Publishers.

[8] L.A. Zadeh, "fuzzy sets," *Information and Controls*, vol. 8, pp. 338–352, 1965.

[9] D. Cirotteau, D.Fober, S. Letz, Y. Orlarey, "Un pitch-tracker monophonique," in *Proceeding of the Journes d'informatique Musicale (JIM)*, Bourges, 2001, pp. 217–223.

[10] S. Dixon, "On the Computer Recognition of Solo Piano Music," in *Australasian Computer Music Conference, Brisbane, Australia*, 2000, pp. 31–37.

[11] A. T. Cemgil, Ed., *SNN - Department of Medical Physics and Biophysics, University of Nijmegen*, http://www.mbfys.kun.nl/ cemgil/software.html.