BLACKBOARD SYSTEM AND TOP-DOWN PROCESSING FOR THE TRANSCRIPTION OF SIMPLE POLYPHONIC MUSIC

Juan Pablo Bello and Mark Sandler

Department of Electronic Engineering, King's College London,
Strand, London WC2R 2LS, UK
juan.bello_correa@kcl.ac.uk

ABSTRACT

A system is proposed to perform the automatic music transcription of simple polyphonic tracks using top-down processing. It is composed of a blackboard system of three hierarchical levels, receiving its input from a segmentation routine in the form of an averaged STFT matrix. The blackboard contains a hypotheses database, a scheduler and knowledge sources, one of which is a neural network chord recogniser with the ability to reconfigure the operation of the system, allowing it to output more than one note hypothesis at a time. The basic implementation is explained, and some examples are provided to illustrate the performance of the system. The weaknesses of the current implementation are shown and next steps for further development of the system are defined.

1. INTRODUCTION

Musical transcription of audio data is the process of taking a sequence of digital data corresponding to the sound waveform and extracting from it the symbolic information related to the high-level musical structures that might be seen on a score [1]. The score and the orchestra are the parts that can be defined in a musical track [2] and in an academic music representation, just the former can be described. The purpose of the present work is to automatically extract score "features" from monophonic and simple polyphonic music tracks (monotimbric music with chords), using a computational reasoning model called blackboard system [3][4] and combining top-down (prediction-driven) processing with the bottom-up (data-driven) techniques already implemented in [5].

1.1. Blackboard system

The blackboard system is a relatively complex problem-solving model prescribing the organisation of knowledge and data, and the problem-solving behaviour within the overall organisation [4]. It receives its name from the metaphor of a group of experts trying to solve a problem plotted on a blackboard, each expert just act when her/his specific area of expertise is required in the developing of the solution.

In contrast to the usual paradigm of signal processing algorithms, where algorithms are described by data flowcharts showing the progress of information along chains of modules [6], the architecture of the blackboard system is opportunistic, choosing the specific module needed for the development of the solution at each time step. Due to its open architecture different knowledge can be easily integrated into the system, allowing the utilisation of various areas of expertise. The basic structure of a blackboard system is depicted in figure 1 and consists of three fundamental parts: *the blackboard*: global database where the hypotheses are proposed and developed, which interact with all the modules present in the system; *the scheduler or opportunistic control system:* determines how the hypotheses are developed and by who; and *the knowledge sources or "experts" of the system:* modules that execute the actions intended to develop the hypotheses present in the blackboard.

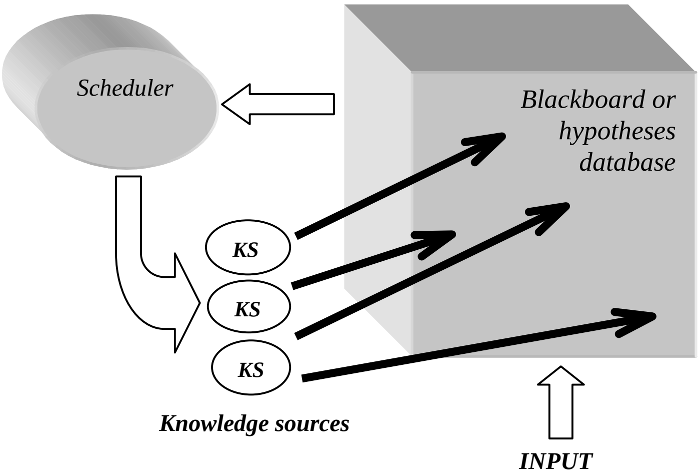

Figure 1: The basic structure of a blackboard system

The system operates in time steps, executing one action at a time. The scheduler prioritises within the existing list of knowledge sources, determining the order in which these actions are executed. Each knowledge source consists of a sort of "if/then" (precondition/action) pair. When the precondition of a certain knowledge source is satisfied, the action described in its programming body is executed, placing its output in the blackboard. These knowledge sources can perform different kinds of activities, such as detecting and removing unsupported hypothesis from the blackboard or stimulating the search for harmonics of a given note hypothesis.

There are several implementations of blackboard systems in automatic music transcription [3][7][8], however part of the knowledge a human being use to transcribe music is based on his/her experience hearing music files and the inherent structures present in these, and in those systems this knowledge is ignored.

As [6] specifies, the structure of the blackboard makes little distinction between explanatory and predictive operations; hypotheses generated for modules of inference can reconfigure the operation of the system and bias the search within the solution space.

1.2. Top-Down and Bottom-up Processing

In bottom-up processing, the information flows from the low-level stage, that of the analysis of the raw signal, to the highest level representation in the system, in our case that of the note hypotheses. In this technique, the system does not know anything about the object of the analysis previous to the operation, and the result depends on the evolution of the data in its unidirectional flow through the hierarchy of the processor. This approach is also called data-driven processing. In contrast, the approach when the different levels of the system are determined by predictive models of the analysed object or by previous knowledge of the nature of the data is known as top-down or prediction-driven processing [9].

Despite the fact that top-down processing is believed to take place in human perception, most of the systems implemented until now are based on bottom-up processing, and just in the last years the implementation of predictive processing to recreate these perceptual tasks had become a common choice between researchers of this field [1][6][9][10]. On tasks such as automatic music transcription, the "inflexibility" of bottom-up systems made them unable to achieve results in a general context, outlining the need for the prediction-driven approach to be used.

In this work, the top-down processing is achieved through the implementation of a connectionist system. This kind of system consists of many primitive cells (units), which are working in parallel and are connected via directed links. Through these links, activation patterns are distributed imitating the basic mechanism of the human brain, which is why these models are also called neural networks [11]. Knowledge is usually distributed throughout the net and stored in the structure of the topology and the weights of the links; the networks are organized by automatic training methods, which help the development of specific applications. If adequately trained, these networks can acquire the experience to make decisions in very specific problems. Here, the problem is to identify the presence of a chord in a given segment of a music signal. As extensive documentation of neural networks is available, no further explanation of this topic will be developed here, just the basics of the implemented system are explained in section 2.3.

2. IMPLEMENTATION

2.1. Segmentation

Just a brief explanation of the system's front end is described here. The onset detection aims to evaluate the time instant when a new note is played in a sound file. Analysing the running spectrum of a sound it is possible to notice that when a new event occurs the high frequency content is increased [12][13]. The measure of the high frequency content (HFC) is given by:

$$HFC = \sum_{k=2}^{(N/2)+1} \left(\left| X(k) \right|^2 \bullet k \right) \qquad \text{(Eq.1)}$$

Where "X(k)" is the FFT array of the audio signal and "N" is its length, while "k" is used as a linear factor to emphasize the high frequencies in the frame. The HFC and the Energy (E) calculated on each frame are used to build the detection function:

$$DF_r = \frac{HFC_r}{HFC_{r-1}} * \frac{HFC_r}{E_r} \qquad \text{(Eq.2)}$$

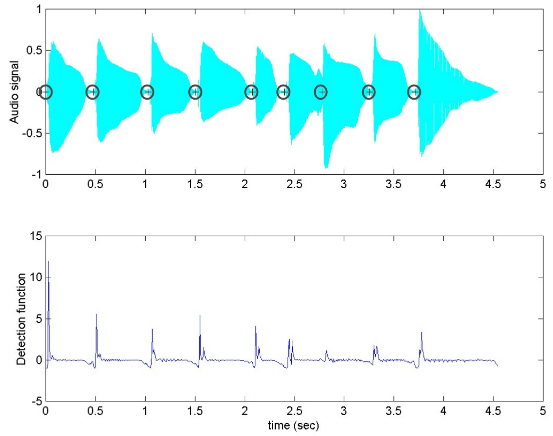

Figure 2: The original signal (a tenor sax riff), the detection function and the estimated onsets and offsets (circled) over the signal.

As can be seen in figure 2, this function shows sharp peaks in the instant when the transient occurs. A criteria based on the slope of these peaks was used to determine the onset's time. After this process, the segmentation is performed averaging the signal's STFT between onsets. This is used as the input of the blackboard system.

2.2. Blackboard Implementation

The Blackboard system's architecture is based on that of Martin's implementation [3] and is shown in figure 3.

At the lower level, the system receives the averaged STFT of the signal and identifies the peaks of the spectrum. Of this group just the peaks higher than an amplitude threshold are considered to build a *Tracks* matrix, containing the magnitude and frequency of each. This information is fed to the database and exposed to the evaluation of the knowledge sources (KS) to produce new hypotheses.

There are three different levels of information present on the database: tracks, partials and notes. The *tracks* information is automatically provided at the beginning of the system operation, however the *notes* and *partials* information are the product of the knowledge sources interaction with the database. It is the main task of the Scheduler to determine the need for a specific kind of information and to activate the corresponding knowledge source. In the present system a table of preconditions is evaluated at each time step and a rating is given to each knowledge source determining the order in which these will operate.

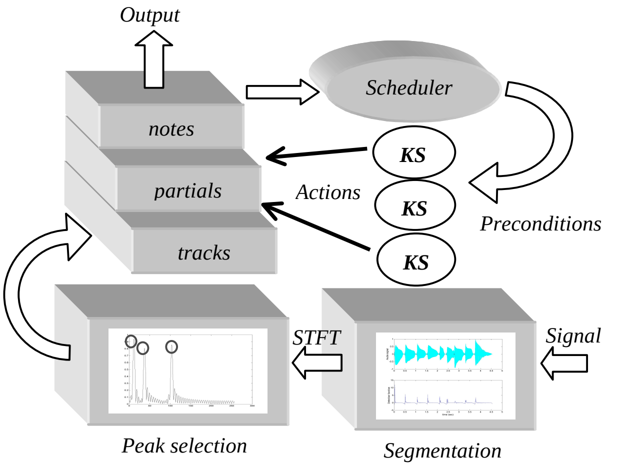

Figure 3: The blackboard architecture of this implementation.

At the *tracks* level, all the remaining peaks of the STFT have an equal chance of becoming notes, but as the operation of the system goes forward and new hypotheses are produced and evaluated by the KS, ratings are given to narrow the search for musical notes in the spectrum.

In the case of the *partials*, the rating is based on the magnitude of the nearest peak (within a specific range) to the ideal frequency of the hypothesis. For *notes*, rating is based on the presence and magnitude of peaks corresponding to the ideal partials this note should have [14]. All this information is stored in a matrix called “Hypotheses”.

2.3. Neural Network Implementation

In the neural network implemented, the information flows in one way from input to output. There is no feedback, which means that the output of any layer does not affect that same layer. This type of network is known as feed-forward.

The structure of this implementation consists of three layers: an input, an output and a hidden layer. The activation function implemented for all the neurons is the sigmoid transfer function. The learning is supervised. Training a feed-forward neural network with supervised learning consists of the following procedure [11]:

1. An input pattern is presented to the network. The input is then propagated forward in the net until activation reaches the output layer. This is called the forward propagation phase.
2. The output of the output layer is then compared with the teaching input. The error, i.e. the difference δ_j between the output o_j and the teaching input t_j of a target output unit j, is then used together with the output o_i of the source unit i to compute the necessary changes of the link w_{ij}. To compute the deltas of inner units for which no input is available, (units of hidden layers) the deltas of the following layer, which are already computed, are used in a formula given below. In this way the errors (deltas) are propagated backward, so this phase is called backward propagation.
3. In this implementation offline learning is used, which means that the weights changes $\Delta\omega_{ij}$ are cumulated for all patterns in the training file and the sum of all changes is applied after one full cycle (epoch) through the training pattern file. This is also known as batch learning.

Here, the input pattern consists of a 256 points spectrogram of a piano signal’s segment (either a note or a chord), part of the batch of samples covering five octaves of the instrument. The target output is just represented for the absence “0” or presence “1” of a chord in the sample. The weight changes were calculated using the backpropagation weight update rule, also called generalized delta-rule, which reads as follows [11]:

$$\Delta\omega_{ij} = \eta\delta_j o_i$$

(Eq.3)
where,

$$\delta_j = f_j'(net_j)(t_j - o_j)$$

(Eq.4) if unit j is an output unit or

$$\delta_j = f_j'(net_j)\sum_k \delta_k \omega_{jk}$$

(Eq.5) if unit j is a hidden unit, where:

η learning factor *eta* (a constant)
δ_j error (difference between the real output and the teaching input) of unit j
t_j teaching input of unit j
o_i output of the preceding unit i
i index of a predecessor to the current unit j with link w_{ij} from i to j
j index of the current unit
k index of a successor to the current unit j with link w_{jk} from j to k

2.4. Neural Network Interaction with the Blackboard

The network is trained offline to obtain a set of parameters adequate to the task required, in this case the recognition of the presence of a chord in a spectrogram. When the overall system is running, the network receives as an input the same STFT data the blackboard system analyses. In the original blackboard’s process, just the note hypotheses with rating bigger than a cut-off threshold remained as valid hypotheses [5], in this version of the system, the output of the neural network changes the performance of the system allowing more than one note hypothesis to survive if necessary. As illustrated in figure 4, this process reshapes the *Hypotheses* matrix, changing its structure and adding a new level of information in the system: *chords*. Due to this, the knowledge sources that interact with the mentioned matrix are structurally modified and urged to link strong note hypotheses present in the blackboard to produce hypothetic chords. As the scheduler detects the need for this new kind of information, the priority list of KS operation is reconfigured to favour the modified knowledge sources. However it is possible that even with the neural network proposing the presence of a chord in the segment, just a single note is output by the system due to the lack of multiple strong rated note hypotheses in the system. In this first approach, just chords of two or three notes can be identified by the system.

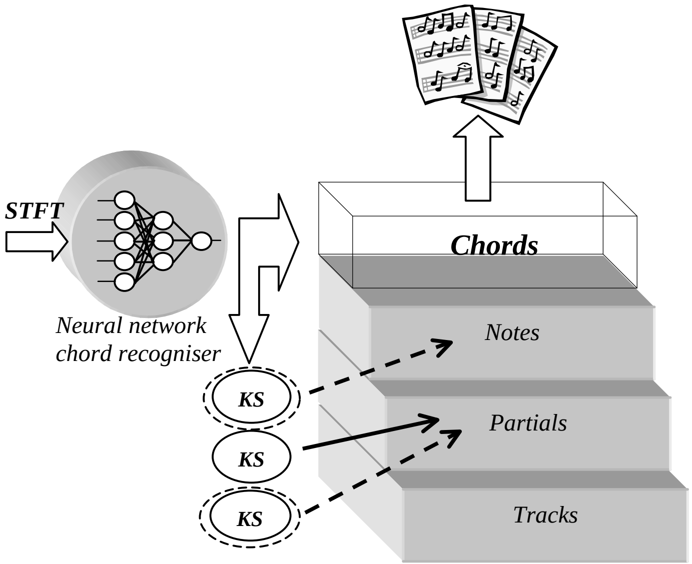

Figure 4: The Interaction between the blackboard and the neural network.

After the selection of hypotheses is made, each of the frequencies obtained is rounded towards the nearest ‘musical’ frequency (ideal frequency corresponding to a musical note) and introduced into a score file. This is a text file written in CSOUND™ language [15], which can be compiled and rendered with an orchestra file (a sine wave sound for these experiments), obtaining an audible representation of the original signal.

The MIDI number corresponding to each frequency is calculated as well [16][17], and the output is represented in the form of a “piano roll”, proportioning a graphical representation of the proposed musical events detected by the system.

3. EXAMPLES

In the first example, illustrated in figure 5, a piano riff is plotted, consisting on a succession of four notes (C_5 D_5 E_5 F_5) followed by a C major chord (C_5 E_5 G_5). The notes and the chords are recognised successfully by the system. This example is intended just to show the main capabilities of the current system. Notice that the notes and silences are well differentiated and the network identified the presence of a chord related with the last onset, causing the blackboard to output the three higher rated hypotheses of the segment.

The second example shown in figure 6 represents a four bar section of a piano song, including four chords. Several mistakes are made in the transcription of the notes of three chords (the first three of the figure), where correct note hypotheses were discarded by the system in favour of their lower octave equivalents. This octave error was detected as well in the note before the last chord, where the note C_6 was selected over the correct C_5. Another error in the transcription is related to the non-detection of an onset in the seventh second of the song causing a wrong segmentation of the piece. The spectrogram of this segment was identified as a chord by the neural network, probably due to the presence of two strong fundamental pitches in the time window averaged. As can be seen in the figure 6 a nonexistent chord was plotted between the times of 6.7164 and 7.3839 seconds, containing both the original notes played in that segment. The other twelve notes of the piece and the last chord were correctly identified by the system.

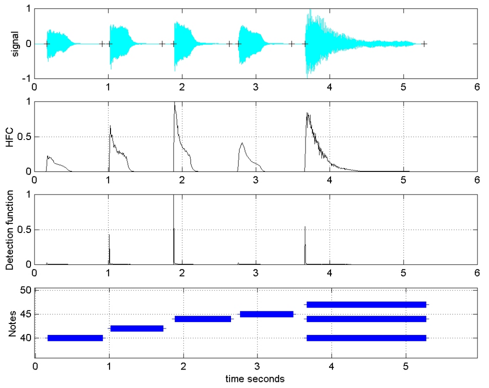

Figure 5: Example of automatic transcription of a piano riff.

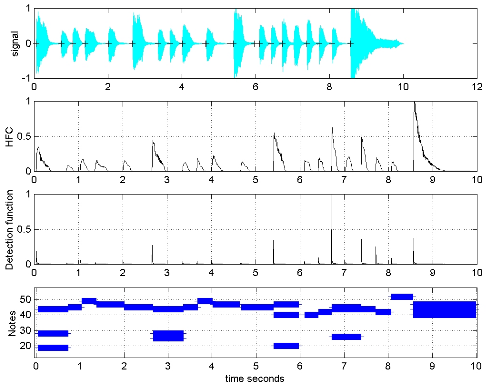

Figure 6: Example of automatic transcription of a simple polyphonic piano song of four measures.

4. CONCLUSIONS AND NEXT STEPS

More tests needs to be performed to fully evaluate the current system. However, from the preliminary tests some strong limitations can be detected in the system than need to be overcome in future implementations. The incorrect selection of notes in different octaves than played, is a recurrent mistake in different tests performed. This is due to the fact that same notes in different octaves have the same harmonic content and the knowledge sources of the blackboard base their selection of notes on the partials information. This is fully dependent on the rating function implemented, specifically of the weight used for the fundamental, even and odd partials. This works as a timbre model and modifying it can make the system more robust in identifying octaves for specific instruments (piano in this case) but will affect the generality of the system. More knowledge sources could be implemented that use musical knowledge to discern this type of structures.

The onset detection is a limitation as well in the performance of the system. It does not work uniformly in all the frequency range, and with different kind of instruments. It relies on the energy content of the signal, which makes it especially weak on facing expressive features in music (legato, glissando, tremolo, vibrato, etc). A more successful approach would be to rely on pitch changes, but due to the computational load that the blackboard's operation imply this is not feasible with the current spectral analysis. More efficient signal processing methods will help to improve the relation between computational time and onset reliability. Analysis with wavelets, Multiresolution Fourier Transform (MFT) [18]or the log-lag correlogram [6][7] are currently studied for future systems.

The architecture of the blackboard needs to be modified, incorporating dynamic structures to handle different sized hypotheses, e.g. chords of more than three notes. Also, the training space of the network has to be expanded to all the octaves of the piano.

In general the system relies on heuristic parameters for its operation. This makes it less general, specifically in the handling of different timbres and different frequency ranges. Tests are being performed to determine these constraints and to develop more general rules to make the system less parameter dependant. The flexible architecture of the blackboard is its strong asset.

As a first approach, the results depicted here are very encouraging showing that further development of these ideas could be the way for more robust and general results. Currently we are working on these points mentioned to improve the system.

5. REFERENCES

[1] Eric Scheirer. "Extracting expressive performance information from recorded music". Master's thesis, MIT, 1995.

[2] Eric Scheirer. Editor, "MPEG-4 Structured Audio. FCD 14496-3 Subpart 5, May 1998".

[3] Keith Martin. "A Blackboard system for Automatic Transcription of Simple polyphonic Music". MIT Media Lab, Technical Report # 385, 1995.

[4] R.S Engelmore and A.J. Morgan. "Blackboard Systems". Addison-Wesley publishing, 1988.

[5] J.P. Bello, G. Monti and M. Sandler. "An Implementation of Automatic Transcription of Monophonic Music with a Blackboard System". Proceedings of the ISSC, June 2000.

[6] Daniel Ellis. "Prediction-driven computational auditory scene analysis". PhD Thesis, MIT, June 1996.

[7] Keith Martin. "Automatic Transcription of Simple polyphonic Music: Robust Front End Processing". MIT Media Lab, Technical Report # 399, December 1996.

[8] Daniel Ellis. "Mid-level Representation for computational auditory scene analysis". In Proc. Of the Computational Auditory Scene Analysis Workshop; 1995 International Joint Conference on Artificial intelligence, Montreal, Canada, August 1995.

[9] Anssi Klapuri. "Automatic Transcription of Music". MSc Thesis, Tampere University of Technology, 1998.

[10] Malcolm Slaney. "A critique of pure audition". In Proc. Of the Computational Auditory Scene Analysis Workshop, Montreal, Canada, August 1995.

[11] Stuttgart Neural Network Simulator. User Manual, version 4.1. University of Stuttgart, Institute for Parallel and Distributed High Performance Systems. Report No. 6/95.

[12] Tristan Jehan. "Music Signal Parameter Estimation". CNMAT Berkeley, USA. 1997.

[13] P. Masri and A. Bateman. "Improved Modelling of Attack Transient in Music Analysis-Resynthesis". University of Bristol. 1996.

[14] Randall Davis, Bruce Buchanan, and Edward Shortliffe. "Production Rules as a representation for a Knowledge-Based Consultation Program". Artificial Intelligence, 8:15-45, 1977.

[15] Barry Vercoe. "CSOUND A Manual for the Audio Processing System and Supporting Programs with Tutorials". Media Lab, M.I.T, Massachusetts, USA. 1992

[16] James H. McClellan, Ronald Schafer and Mark Yoder. "DSP First: A Multimedia Approach". Prentice Hall, USA. 1998

[17] MIDI Manufacturers Association. "The Complete MIDI 1.0 Detailed Specification", 1996.

[18] E.R.S Pearson. " The Multiresolution Fourier Transform and its Application to the Analysis of Polyphonic Music". PhD Thesis, Warwick University, 1991.