A Connectionist Approach to Automatic Transcription of Polyphonic Piano Music

Matija Marolt, *Member, IEEE*

Abstract— **In this paper, we present a connectionist approach to automatic transcription of polyphonic piano music. We first compare the performance of several neural network models on the task of recognizing tones from time-frequency representation of a musical signal. We then propose a new partial tracking technique, based on a combination of an auditory model and adaptive oscillator networks. We show how synchronization of adaptive oscillators can be exploited to track partials in a musical signal. We also present an extension of our technique for tracking individual partials to a method for tracking groups of partials by joining adaptive oscillators into networks. We show that oscillator networks improve the accuracy of transcription with neural networks. We also provide a short overview our entire transcription system and present its performance on transcriptions of several synthesized and real piano recordings. Results show that our approach represents a viable alternative to existing transcription systems.**

Index Terms—**adaptive oscillators, music transcription, neural networks**

I. Introduction

MUSIC transcription could be defined as an act of listening to a piece of music and writing down music notation for the piece. If we look at the traditional way of making music, we can imagine a performer reading a score, playing an instrument and thus producing music. Transcription of polyphonic music (polyphonic pitch recognition) is the reverse process; an acoustical waveform is converted into a parametric representation, where notes, their pitches, starting times and durations are extracted from the signal. Transcription is a difficult cognitive task and is not inherent in human perception of music, although it can be learned. It is also a very difficult problem for current computer systems. Separating notes from a mixture of other sounds, which may include notes played by the same or different instruments or simply background noise, requires robust algorithms with performance that should degrade gracefully when noise increases.

Applications of a music transcription system are versatile. Transcription produces a compact and standardized parametric representation of music. Such representation is needed for content-based retrieval of music in most current musical databases. It is useful in music analysis systems for tasks such as melody extraction, music segmentation and rhythm tracking. Transcription aids musicologists in analyzing music that has never been written down, such as improvised or ethnical music. The conversion of an acoustical waveform into a parametric description is also useful in the process of making music, as well as in newer coding standards, such as MPEG-4, which may include such descriptions.

First attempts of transcribing polyphonic music have been made by Moorer [1]. His system was limited to two voices of different timbres and frequency ranges and had limits on allowable intervals. In recent years, several systems have been developed. Some of them are targeted to transcription of music played on specific instruments [2-4], while others are general transcription systems [5-6]. All of them share several common characteristics. In the beginning, they calculate a time-frequency representation of the musical signal. Authors use various representations ranging from Fourier analysis to bilinear distributions. In the next step, the time-frequency representation is refined by locating partials in the signal. To track partials, most systems use ad hoc algorithms such as peak picking and peak connecting. Partial tracks are then grouped into notes with different algorithms relying on cues such as common onset time and harmonicity. Some authors use templates of instrument tones in this process [3-6], as well as higher-level knowledge of music, such as probabilities of chord transitions [6].

Recognizing notes in a signal is a typical pattern recognition task and we were surprised to that few current systems use machine learning algorithms in the transcription process. Therefore, our motivation was to develop a transcription system based on connectionist algorithms, such as neural networks, which have proved to be useful in a variety of pattern recognition tasks. We tried to avoid using explicit symbolic algorithms, and employed connectionist approaches in different parts of our system instead.

Music transcription is a difficult task, and we therefore put one major constraint on our transcription system: it only transcribes piano music, so piano should be the only instrument in the analyzed musical signal. We didn't make any other assumptions about the signal, such as maximal polyphony, minimal note length, style of transcribed music or the type of piano used. The system takes an acoustical

Manuscript received October 23, 2001.

M. Marolt is with Faculty of Computer and Information Science, University of Ljubljana, Trzaska 25, 1000 Ljubljana, Slovenia (phone:+386 1 4768483; fax: +386 1 4264647; e-mail: matija.marolt@fri.uni-lj.si).

waveform of a piano recording (44.1 kHz sampling rate, 16 bit resolution) as its input. Stereo recordings are converted to mono. The output of the system is a MIDI file containing the transcription. Notes, their starting times, durations and loudness' are extracted from the signal.

The organization of this paper is as follows. In Section II we propose a new model for tracking partials in a polyphonic audio signal, based on networks of adaptive oscillators. Section III presents a comparison of several neural network models for recognizing piano notes in outputs of the partial tracking model. Section IV presents a quick overview of our complete transcription system and in section V we present performance statistics of the system on transcriptions of several synthesized and real recordings of piano music. We also provide a comparison of our results to results of some other authors. Section VI concludes this paper.

II. Partial Tracking with Networks of Adaptive Oscillators

Most current transcription systems (including ours) are composed of two main parts: a partial tracking module, which calculates a clear and compact time-frequency representation of the input audio signal, and a note recognition module, which groups the found partials into notes. In contrast to most other current transcription approaches, we use connectionist methods for solving both problems. In this section, we propose a new model for tracking groups of partials in an audio signal with networks of adaptive oscillators. We describe how neural networks can be used for note recognition in section III, where we also provide a comparison of several neural network models for this task.

Tones of melodic music instruments can be roughly described as a sum of frequency components (sinusoids) with time-varying amplitudes and almost constant frequencies. These frequency components are called partials and can be recognized as prominent horizontal structures in the time-frequency representation of a musical signal. By finding partials, one can obtain a clearer and more compact representation of the signal, and partial tracking is therefore used in all current transcription systems. Although partial tracking algorithms play an important role in transcription systems, because they provide data to the note recognition module, little attention has been paid to the development of these algorithms. Most systems use a procedure similar to that of a tracking phase vocoder [13]. After the calculation of a time-frequency representation, peaks are computed in each frequency image. Only peaks with amplitude that is larger than a chosen (possibly adaptive) threshold are kept as candidate partials. Detected peaks are then linked over time according to intuitive criteria such as proximity in frequency and amplitude, and partial tracks are formed in the process. Such approach is quite susceptible to errors in the peak peaking procedure, where missed or spurious peaks can lead to fragmented or spurious partial tracks. Some systems therefore use additional heuristics for merging fragmented partial tracks. The second main shortcoming of the "peak picking-peak connecting" approach is detection of frequency modulated partials. Here, the peak connecting algorithm can fail if it is not designed to tolerate frequency modulation. An innovative approach to partial tracking has been proposed by Sterian [3], who still uses a peak picking procedure in the first phase of his system, but later uses Kalman filters, trained on examples of instrument tones, to link peaks into partial tracks. His system still suffers due to errors in the peak picking stage, but its main drawback is that partials have to be at least 150 ms long to be discovered. For our system, this is a very serious limitation, because tones in piano music are frequently shorter than 100 ms.

The shortcomings of most current partial tracking approaches have led us to the development of a new partial tracking model. In this section, we propose a partial tracking model based on a connectionist paradigm. It is composed of two parts: an auditory model, which emulates the functionality of human ear, and adaptive oscillators that extract partials from outputs of the auditory model. We also present an extension of the model for tracking individual partials to a model for tracking groups of harmonically related partials by joining adaptive oscillators into networks.

A. *Auditory Model*

The first stage of our partial tracking algorithm transforms the acoustical waveform into time-frequency space with an auditory model, which emulates the functionality of human ear. The auditory model consists of two parts. A filterbank is first used to split the signal into several frequency channels, modeling the movement of basilar membrane in the inner ear. The filterbank consists of an array of bandpass IIR filters, called gammatone filters. The implementation we use is described in [14-16]. We are using 200 filters with center frequencies logarithmically spaced between 70 and 6000 Hz.

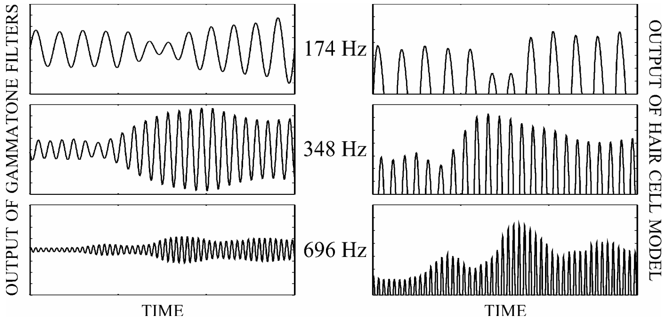

Fig. 1. Analysis of three partials of piano tone F3 with the auditory model.

Subsequently, the output of each gammatone filter is processed by the Meddis' model of hair cell transduction [17]. The hair cell model converts each gammatone filter output into a probabilistic representation of firing activity in the auditory nerve. Its operations are based on a biological model of the hair cell and it simulates several of the cell's characteristics, most notably half-wave rectification, saturation and adaptation. Saturation and adaptation are very important to our model, as they reduce the dynamic range of the signal, and in turn enable our partial tracking system to track partials with low amplitude. These characteristics can be observed in Fig. 1, displaying outputs of three gammatone filters and the hair cell model on the 1., 2., and 4. partial of piano tone F3 (pitch 174 Hz).

B. *Partial Tracking with Adaptive Oscillators*

The auditory model outputs a set of frequency channels containing quasi-periodic firing activities of inner hair cells (see Fig. 1). Temporal models of pitch perception are based on the assumption that periodicity detection in these channels forms the basis of human pitch perception. Periodicity is usually calculated with autocorrelation. This produces a three-dimensional time-frequency representation of the signal (autocorrelogram), with time, channel center frequency and autocorrelation lag represented on orthogonal axes. A summary autocorrelogram (summed across frequency channels) can be computed to give a total estimate of periodicity of the signal at a given time. Meddis and Hewitt [18] have demonstrated that the summary autocorrelogram explains the perception of pitch in a wide variety of stimuli.

We decided to use a different approach for calculating periodicity in frequency channels. It is based on adaptive oscillators that try to synchronize to signals in output frequency channels of the auditory model. A synchronized oscillator indicates that the signal in a channel is periodic, which in turn indicates that a partial with frequency similar to that of the oscillator is present in the analyzed signal.

An oscillator is a system with periodic behavior. It oscillates in time according to its two internal parameters: phase and frequency. An adaptive oscillator adapts its phase and frequency in response to its input (driving) signal. When a periodic signal is presented to an adaptive oscillator, it synchronizes to the signal by adjusting its phase and frequency to match that of the driving signal. By observing the frequency of a synchronized oscillator, we can make an accurate estimate of the frequency its driving signal.

Various models of adaptive oscillators have been proposed, some have also found use in computer music researches for modeling rhythm perception [19-20] and for simulation of various psychoacoustic phenomena [21]. After reviewing several models, we decided to use a modified version of the Large-Kolen adaptive oscillator [19] in our system.

The Large-Kolen oscillator oscillates in time according to its period (frequency) and phase. The input of the oscillator consists of a series of discrete impulses, representing events. After each oscillation cycle, the oscillator adjusts its phase and period, trying to match its oscillations to events in the input signal. If input events occur in regular intervals (are periodic), the final effect of synchronization is alignment of oscillations with input events. Phase and period of the Large-Kolen oscillator are updated according to the modified gradient descent rule, minimizing an error function that describes the difference between input events and beginnings of oscillation cycles. The speed of synchronization can be controlled by two oscillator parameters.

Our partial tracking model uses adaptive oscillators to detect periodicity in output channels of the auditory model. Each output channel is routed to the input of one adaptive oscillator. The initial frequency of the oscillator is equal to the center frequency of its input channel. When an oscillator synchronizes to its input, this indicates that the input signal is periodic and consequently that a partial with frequency similar to that of the oscillator is present in the input signal. A synchronized oscillator therefore represents (tracks) a partial in the input signal.

To improve partial tracking, we made a few minor changes to the Large-Kolen oscillator model. Most notably, we added a new measure of successfulness of synchronization that is used as the oscillator's output value. The measure is related to the amount of phase corrections made in the synchronization process; less phase corrections signify better synchronization. Oscillator's output therefore indicates how successfully the oscillator managed to synchronize to its input signal.

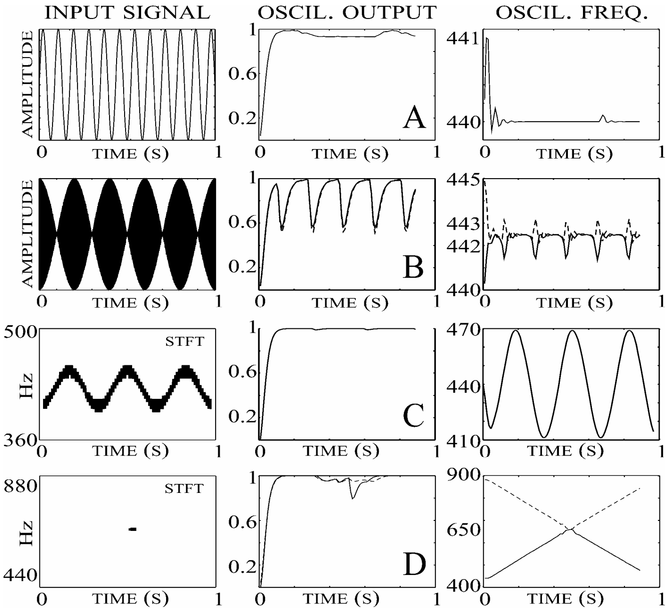

Fig. 2. Partial tracking with adaptive oscillators.

The modified Large-Kolen oscillator can successfully track partials with diverse characteristics. Four examples are given in Fig. 2. Example A presents a simple case of tracking a 440 Hz sinusoid. The oscillator (initial frequency 440 Hz) synchronizes successfully, as can be seen from its output, and after an initial 1 Hz rise, its frequency settles at 440 Hz. Example B shows how two oscillators with initial frequencies set to 440 and 445 Hz synchronize to a sum of 440 and 445 Hz sinusoids (5 Hz beating). Both oscillators synchronize successfully at 442.5 Hz, as can be seen from their outputs and frequencies. The behavior is consistent to human perception of the signal. Example C is shows the tracking of a frequency modulated 440 Hz sinusoid. The oscillator synchronizes successfully, its frequency follows that of the sinusoid. The last example (D) shows how two oscillators track two frequency components that rise/fall from 440 to 880 Hz.

Tracking is successful; each oscillator tracks the component closest to its input frequency channel.

C. Tracking Groups of Partials with Networks of Adaptive Oscillators

In the previous section we demonstrated how adaptive oscillators can be used to track partials in a musical signal. We extended the model of tracking individual partials to a model of tracking groups of harmonically related partials by joining adaptive oscillators into networks.

Networks consist of up to ten interconnected oscillators. Their initial frequencies are set to integer multiples of the frequency of the first oscillator (see Fig. 3). As each oscillator in the network tracks a single partial close to its initial frequency, a network of oscillators tracks a group of up to ten harmonically related partials, which may belong to one tone with pitch equal to the frequency of the first oscillator. Output of the network is related to the number of partials found by its oscillators and therefore represents the strength of a group of partials that may belong to tone with pitch f (Fig. 3).

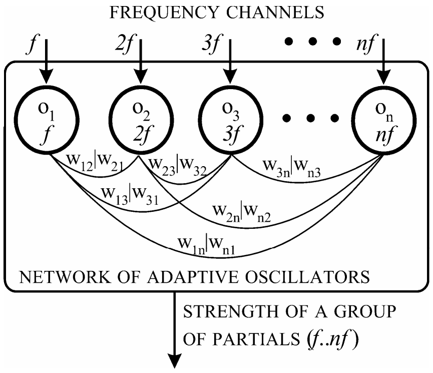

Fig. 3. A network of adaptive oscillators.

Our system uses 88 oscillator networks to track partial groups corresponding to all 88 piano tones (A0-C8). The initial frequency of the first oscillator in each network is set to the pitch of one of 88 piano tones. Initial frequencies of other oscillators are integer multiples of the first oscillator's frequency (see Fig. 3). Networks consist of up to ten oscillators. This number decreases as the frequency of the first oscillator in the network increases, because the highest tracked partial lies at 6000 Hz; i.e. network corresponding to tone A6 only has three oscillators with initial frequencies set to 1760 Hz, 3520 Hz and 5280 Hz.

Within a network, each oscillator is connected to all other oscillators with excitatory connections. These connections are used to adjust the frequencies and outputs of non-synchronized oscillators in the network with the goal of speeding up their synchronization. Only a synchronized oscillator can change frequencies and outputs of other oscillators in the network. Adjustments are made according to the following rules:

$$
\begin{aligned}
r_p &= \exp(-1000(dp_d/(sp_s)-1)^2) & \qquad r_c &= \exp(-2.3c_d^2/c_s^2) \\
p_d &= p_d + r_p r_c w_{sd}(sp_s - dp_d)/d & \qquad c_d &= c_d + c_d r_p r_c w_{sd}
\end{aligned}
\tag{1}
$$

d is the number of the destination (non-synchronized) oscillator in the network (1 to 10), while *s* represents the number of the source (synchronized) oscillator. The period of the destination oscillator p_d and its output value c_d change according to two factors: r_p and r_c (Fig. 4). These are two gaussians, representing the ratio of periods of the two oscillators (p_d - period of the destination oscillator, p_s - period of the source oscillator) and the ratio of outputs of the two oscillators (c_d - output of the destination oscillator, c_s output of the source oscillator). Factor r_p is a gaussian with maximum value, when periods of both oscillators are in a perfect harmonic relationship ($dp_d/sp_s = 1$). The value falls as periods drift away from this perfect ratio and approaches zero, when the ratio is larger than a semitone. r_c has the largest value, when a synchronized oscillator influences the behavior of a non-synchronized oscillator (c_s is large, c_d is small) and falls as c_d increases. Connection weights w_{sd} are calculated according to amplitudes of partials in piano tones; the first few partials are considered to be more important and consequently the influence of lower-numbered oscillators in the network is stronger than the influence of higher-numbered oscillators (w_{1n}>w_{n1}).

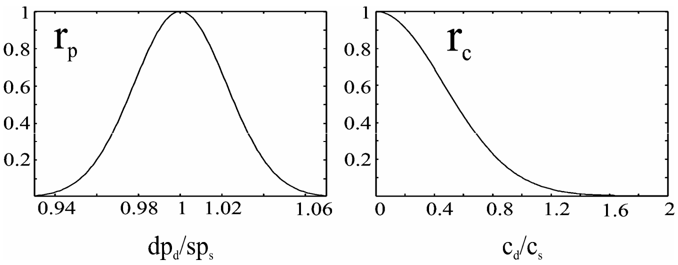

Fig. 4. Plot of factors used for updating periods and output values of oscillators in a network.

Adjustments push the period (frequency) of a non-synchronized oscillator closer to frequency of the partial it should track and also increase its output value. This results in faster synchronization of all oscillators in the network and consequently in faster discovery of a group of partials. The output of a network is calculated as a weighted sum of outputs of individual oscillators in the network and represents the strength of a group of partials tracked by oscillators in the network. Outputs of individual oscillators are weighted according to their importance and deviation of their frequency (f_i) from ideal frequency if_0; an oscillator with large deviation has little influence on output of the network, as it probably tracks a partial that does not belong to the network's group of partials. Larger deviations are tolerated for higher-numbered oscillators to account for frequency stretching. Because the network's output only depends on outputs of its oscillators, it is virtually independent of the amplitude of the tracked partials.

Connecting oscillators into networks has several advantages for our transcription system. Output of a network represents the strength of a group of harmonically related partials tracked by oscillators in the network, which may belong to one tone. Such output provides a better indication of presence of the

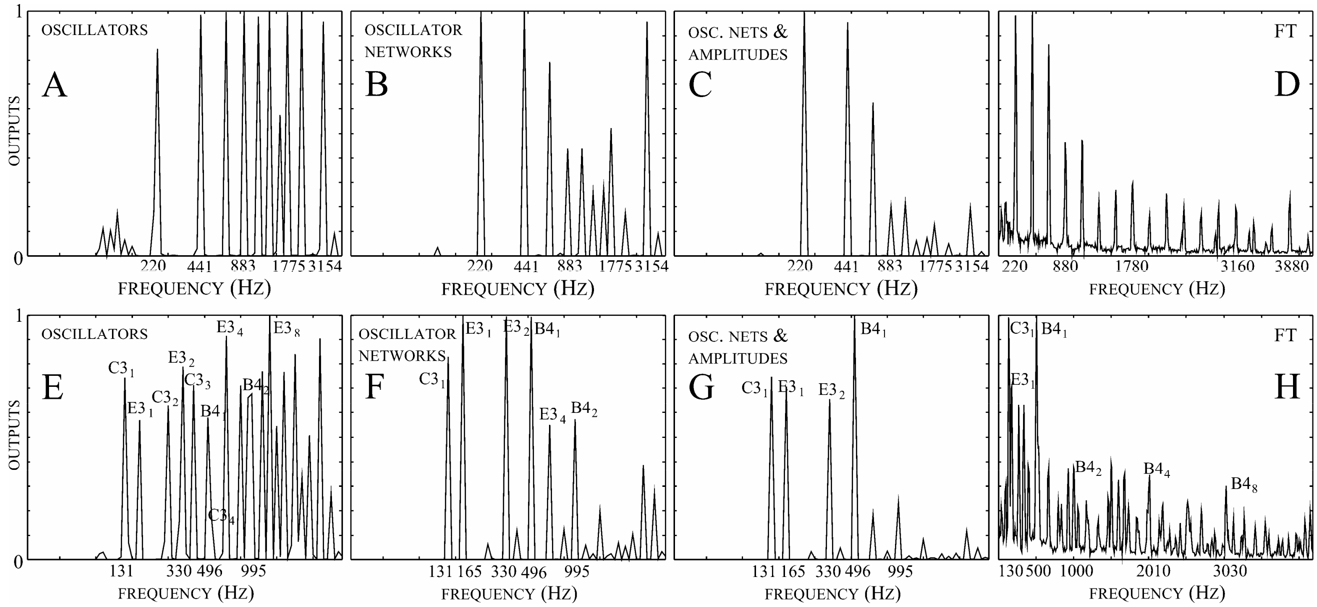

Fig. 5. Representations of piano tone A3 and chord C3E3B4.

tone in the input signal than do outputs of individual oscillators. Noise doesn't usually appear in the form of harmonically related frequency components, so networks of oscillators are more resistant to noise and provide a clearer time-frequency representation of the signal. Within the network, each oscillator is connected to all other oscillators with excitatory connections. Connections are used by synchronized oscillators to speed up synchronization of non-synchronized oscillators, leading to a faster network response and faster discovery of a group of partials.

Fig. 5 displays slices taken from three time-frequency representations of piano tone A3 (pitch 220 Hz –A-D) and piano chord C3E3B4 (E-H), calculated 100 ms after the onset: representation with uncoupled oscillators, representation with networks of adaptive oscillators and short-time Fourier transform. The representation with uncoupled oscillators was calculated with 88 oscillators tuned to fundamental frequencies of piano tones A0-C8. For tone A3, oscillator outputs (independent of partial amplitudes) are presented in Fig. 5A. Fig. 5B shows outputs of 88 oscillator networks, the combination of these outputs with amplitudes of partials is shown in Fig. 5C. Fig. 5D displays 440 frequency bins of the Fourier transform calculated with a 100 ms Hamming window.

Individual oscillators have no difficulty in finding the first eight partials of tone A3 (A). Not all of the higher partials are found, because they are spaced too close together (we use only one oscillator per semitone). Noisy partials found below 220 Hz are the consequence of noise caused by the hammer hitting the strings. Oscillator networks (B) produce a clearer representation of the signal; most notably the noisy partials below 220 Hz are almost completely eliminated. Networks coinciding with tones A3 and A4 produce the highest outputs, because all partials in the networks are found. The output of the network at 3154 Hz, representing the 14th partial, is also very high, because it only has one oscillator that synchronizes with the partial. The combination of outputs of networks with partial amplitudes (C) produces the clearest representation, with the first three A3 partials standing out.

For piano chord C3E3B4 (Figs. 5E-5H), oscillator networks also produce the clearest representation. When amplitudes are combined with networks' outputs (Fig. 5G), only four partials stand out: first partials of all three tones (C3, E3, B4) and the second partial of tone E3 (C_2).

Both examples show that oscillator networks produce a compact and clear representation of partial groups in a musical signal. The main problem of this representation lies in occasional slow synchronization of oscillators in networks, which can lead to delayed discovery of partial groups. This is especially true at lower frequencies, where delays of 40-50 ms are quite common, because synchronization only occurs once per cycle; an oscillator at 100 Hz synchronizes with the signal every 10 ms, so several 10s of milliseconds are needed for synchronization. Closely spaced partials may also slow down synchronization, although it is quite rare for a group of partials not to be found.

III. Neural Networks for Note Recognition in Polyphonic Piano Music

A note recognition module is the central part of every transcription system. Its input usually consists of a set of partials found by the partial tracking module and its task is to associate the found partials with notes. Statistical methods are frequently used to group partials into notes [3,5,6]; in our transcription system the task is performed by neural networks.

We use a set of 76 neural networks to perform note recognition. Inputs of each network are taken from outputs of the partial tracking module presented in the previous section. They contain one or more time frames (sampled at every 10 ms) of output values of oscillator networks, amplitude envelopes of signals in frequency channels of the auditory model (calculated by half-wave rectification and smoothing) and a combination of amplitude envelopes and oscillator networks' outputs.

Each network is trained to recognize one piano note in its input; i.e. one network is trained to recognize note A4, another network recognizes note G4... Altogether 76 networks are used to recognize notes from A1 to C8. This represents the entire range of piano notes, except for the lowest octave from A0 to Ab1. We decided to ignore the lowest octave, because of poor recognition results. These notes are quite rare in piano pieces, so their exclusion doesn't have a large impact on overall performance of the system. Because each neural network recognizes only one note (we call it the target note) in its input, it only has one output neuron; a high output value indicates the presence of the target note in the input signal, a low value indicates that the note is not present.

A. *Comparison of Neural Network Models for Note Recognition*

As we found no previous references to works that use neural networks for transcription of polyphonic music, we made a comparison of several neural network models for note recognition. We tested multilayer perceptrons (MLPs) [8], radial basis function (RBF) networks [11], time-delay neural networks (TDNN) [10], Elman's partially recurrent networks [9] and fuzzy-ARTMAP networks [12]. Supervised learning was used to train all of the tested network models. Because no standard database of music pieces that could be used to train or test transcription systems currently exists, we first constructed a database for training and testing our neural networks. Supervised learning requires that pairs of input-output patterns be presented to the network during training. We therefore constructed the database from a set of synthesized piano pieces and piano chords, which enabled us to collect pairs of input-output patterns for training. The database includes patterns taken from a set of 120 MIDI piano pieces, rendered with 16 different piano samples obtained from commercially available piano sample CD-ROMs (using a sampler with digital I/O). The set contains pieces of various styles, including classical from several periods, ragtime, jazz, blues and pop. To diversify the distribution of notes in the training set and to provide more training patterns for networks that recognize low and high notes (these were not very frequent in the chosen pieces), we complemented the song set with a set of synthesized chords with polyphony from one to six. Notes in each chord were chosen randomly. Altogether, the database consists of around 300,000 pairs of input-output patterns.

The database was used to train a set of neural networks for each of the tested neural network models. Each network in a set recognizes one piano note (its target note) in its input. The training set for each network included approx. 30000 patterns with 1/3 of them containing the target note. Networks were tested on a different database, constructed from 40 new MIDI piano pieces and piano chords (not used in the training database), rendered with over 20 piano samples. The database contains approx. 60000 input-output patterns; each network was tested on 6000 patterns. Average performance statistics on the test database of the entire set of networks for each neural network model are given in Table I.

TABLE I
PERFORMANCE STATISTICS OF NEURAL NETWORK MODELS FOR NOTE RECOGNITION

neural network model	correct	spurious
time-delay NNs	96.8%	13.1%
Elman's NNs	95.2%	13.5%
multilayer perceptrons	96.4%	16.0%
RBF NNs	88.2%	14.6%
fuzzy-ARTMAP	84.1%	18.9%

The table lists average percentages of correctly found and spurious notes (notes found, but not present in the input pattern) for each network model. Time-delay neural networks showed the best performance on the test set. Networks had a single hidden layer with 18 neurons and two time delays. Inputs of the network consisted of three consecutive time frames (time step 10 ms) of outputs of the partial tracking model. We used a modified backpropagation algorithm [9] for training. The performance of TDNNs was superior in comparison to other network models in the number of correctly found notes, as well as in the number of spurious notes found (most of them were octave errors). The largest increase in performance was observed in networks recognizing notes in the C4-A5 interval (261-880 Hz), where time delays contributed to more accurate resolution of octave errors that frequently occur in this interval, mostly because of a high number of partials produced by the lower-pitched notes (A2-C4).

B. *Impact of Partial Tracking on the Accuracy of Note Recognition with Time-Delay Neural Networks*

To assess the impact that the proposed partial tracking module has on the accuracy of note recognition (transcription) with TDNNs, we compared the performance of TDNNs trained on patterns that consisted of outputs of the partial tracking module (as described previously) to the performance of TDNNs trained on patterns that consisted of outputs of a multiresolution time-frequency transform, similar to constant-Q transform [7] with window sizes from 90 ms to 5 ms at frequencies from 60 Hz to 9 kHz.

TABLE II
AVERAGE PERFORMANCE STATISTICS OF SYSTEMS WITH AND WITHOUT PARTIAL TRACKING

	correct	spurious	oct. err.
No PT	92.8	27.9	39.5
With PT	94.4	11.1	77.9

We tested the performance of both sets of TDNNs on transcriptions of several synthesized piano pieces. Table II lists average performance statistics of both sets of networks on seven synthesized piano pieces of different complexities and styles, containing over 20000 notes. Percentages of correctly found notes, spurious notes and octave errors are given for both sets of networks. The percentage of correctly found notes is similar in both systems; partial tracking improves accuracy by approximately 1.5%. Partial tracking significantly reduces the number of spurious notes, as it more than halves. Just as important is the change in the structure of errors. Almost 80% of all errors in the system with partial tracking are octave errors that occur when the system misses or finds a spurious note, because of a note an octave, octave and a half or two octaves apart. Octave errors are very hard to remove, but because the missed or spurious notes are consonant with other notes in the transcribed piece, they aren't very apparent if we listen to the resynthesized transcription. Octave errors are therefore not as critical as some other types of errors (i.e. halftone errors), which make listening to the resynthesized transcription unpleasant. We therefore consider the higher percentage of octave errors in the system with partial tracking to be a significant improvement. Overall, we can conclude that the partial tracking model proposed in section II significantly improves transcription accuracy with TDNNs.

IV. System for Transcription of Piano Music

The presented partial tracking model and time-delay neural networks were incorporated into a system for transcription of piano music, called SONIC. The system also includes an onset detector, a module for detecting repeated notes and simple algorithms for length and loudness estimation (see Fig. 6), all of these parts are briefly presented in this section.

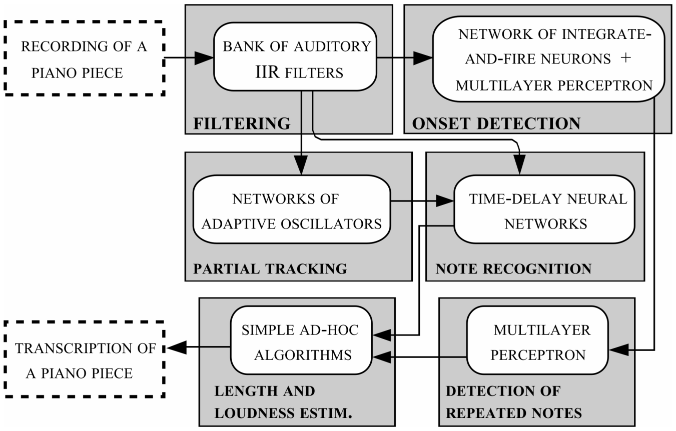
RECORDING OF A PIANO PIECE
BANK OF AUDITORY IIR FILTERS
FILTERING
NETWORK OF INTEGRATE-AND-FIRE NEURONS + MULTILAYER PERCEPTRON
ONSET DETECTION
NETWORKS OF ADAPTIVE OSCILLATORS
PARTIAL TRACKING
TIME-DELAY NEURAL NETWORKS
NOTE RECOGNITION
TRANSCRIPTION OF A PIANO PIECE
SIMPLE AD-HOC ALGORITHMS
LENGTH AND LOUDNESS ESTIM.
MULTILAYER PERCEPTRON
DETECTION OF REPEATED NOTES

Fig. 6. Structure of SONIC.

A. Onset Detection

We added an onset detector to SONIC to improve the accuracy of onset times of notes found by the system. We based our onset detection algorithm on a model proposed by Smith [22] for segmentation of speech signals. The algorithm first splits the signal into several frequency bands with a bank of gammatone filters. We are using the same set of filters as in our partial tracking system. The signal is split into 22 overlapping frequency bands, each covering half an octave. Channels are full-wave rectified and then processed with the following filter:

$$O(t) = \int_0^t (\exp(-\frac{t-x}{f_s t_s} - \exp(-\frac{t-x}{f_s t_l}))s(x)dx \qquad (2)$$

s(x) represents the signal in each frequency channel, f_s the sample rate, t_s and t_l are two time constants. The filter calculates the difference between two amplitude envelopes; one calculated with a smoothing filter with short time constant t_s (6-20 ms), and the other with a smoothing filter with a longer time constant (20-40 ms). The output of the filter has positive values when the signal rises and negative otherwise. Outputs of all 22 filters are fed into a fully connected network of integrate-and-fire neurons. Each neuron in the network is connected to the output of one filter. It accumulates its input over time and if its internal activation exceeds a certain threshold, the neuron fires (emits an output impulse). Firing of a neuron provides indication of amplitude growth in its input frequency channel. After firing, activity of the neuron is reset and the neuron is not allowed to respond to its input for a period of time (50 ms in our model). Neurons are connected to all other neurons in the network with excitatory connections. The firing of a neuron raises activations of all other neurons in the network and accelerates their firing, if imminent. Such mechanism clusters neuron firings, which may otherwise be dispersed in time and improves the discovery of weak onsets.

A network of integrate-and-fire neurons outputs a series of impulses indicating the presence of onsets in the signal. Not all impulses represent onsets, because various noises and beating can also cause amplitude oscillations in the signal. We use a MLP neural network to decide which impulses represent onsets. We trained the MLP on a set of piano pieces, the same as we used for training note recognition networks.

We tested the algorithm on a mixture of synthesized and real piano recordings. It correctly found over 98.5% of all onsets and produced around 2% of spurious onsets. Most of the missed notes were notes played in very fast passages, or notes in ornamentations such as thrills; spurious notes mostly occurred because of beating or other noises in the signal.

B. Repeated Note

Detecting repeated notes in a musical signal can be a difficult problem, even if the played instrument has pronounced onsets (such as piano). An illustration of the problem is given in Fig. 7. The upper part of Fig. 7 shows outputs of the onset detection network and five note recognition networks on an unknown piece of music. Four onsets and five notes were found; note C4 lasts through the entire duration of the piece, while other notes appear for shorter periods of time. Four transcription examples show four possible interpretations of these outputs. Interpretations differ in the way note C4 is handled; we could transcribe it as one whole note, four quarter notes... Altogether eight combinations are possible, and all of them are consistent with networks' outputs.

It becomes apparent that the system needs an algorithm for

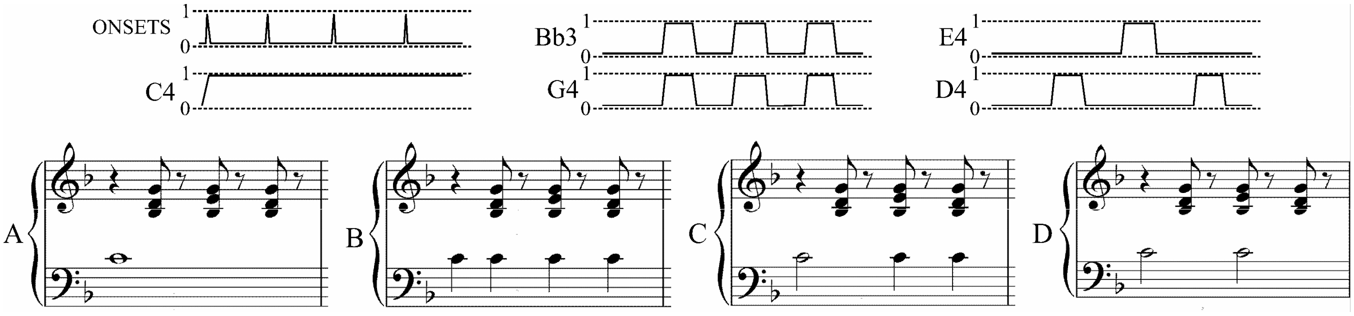

Fig. 7. Different interpretations of networks' outputs.

detecting repeated notes. At first, we used the most obvious solution, which is to track the amplitude of the first harmonic of a possible repeated note and produce a repetition if the amplitude rises enough. Because of shared partials between notes, this approach fails when a note that shares partials with the repeated note occurs in the signal. We therefore decided to entrust the decision on repeated notes to a MLP neural network, trained on a set of piano pieces. Inputs of the MLP consist of amplitude changes, as well as several other parameters. This solution improves transcription accuracy for approximately 2.5% over the "first harmonic" approach.

C. *Tuning, Note Length and Loudness Estimation*

Before transcription actually starts, a simple tuning procedure is used to calculate tuning of the entire piano and initialize frequencies of adaptive oscillators accordingly. The procedure uses adaptive oscillators to find partials in the piano piece and then compares partial frequencies to frequencies of an ideally tuned piano. The tuning of the piano is calculated as a weighted average of deviations of partial frequencies from ideal tuning. Stretching of piano tuning is also taken into consideration in the process. The tuning procedure guarantees unchangeable transcription accuracy, when the piano is tuned differently then the standard A4=440 Hz. The procedure only calculates the tuning of the entire piano, not of individual piano tones.

SONIC also calculates the length and loudness of each note. Both are needed to produce the final MIDI file containing the transcription. The length of a note is calculated by observing activations of the note recognition network; note is terminated when the network's activation falls below the training threshold. Loudness is calculated from the amplitude envelope of the note’s first harmonic.

V. Performance Analysis

A. *Synthesized and Real Recordings*

In this section, we present the performance of our system on transcriptions of three synthesized and three real recordings of piano music. Originals and transcriptions of all presented pieces (and more) can be heard on http://lgm.fri.uni-lj.si/SONIC. Table III lists percentages of correctly found and spurious notes in transcriptions, as well as the distribution of errors into octave, repeated note and other errors. Separate error distributions are given for missed and spurious notes. An error can fall into several categories, so the sum of error percentages may be greater than 100. The total number of notes, as well as maximal and average polyphony of each piece are also shown.

The transcribed synthesized recordings are: (1) J.S. Bach, Partita no. 4, BWV828, Fazioli piano; (2) A. Dvořak, Humoresque no. 7, op. 101, Steinway D piano; (3) S. Joplin, The Entertainer, Bösendorfer piano. Real recordings are: (4) J.S. Bach, English suite no. 5, BWV810, 1st movement, performer Murray Perahia, Sony Classical SK 60277; (5) F. Chopin, Nocturne no. 2, Op. 9/2, performer Artur Rubinstein, RCA 60822; (6) S. Joplin, The Entertainer, performer unknown, MCA 11836.

The average number of correctly found notes in synthesized recordings is around 90%. The average number of spurious notes is 9%. Most of the missed notes are either octave errors or misjudged repeated notes. Notes are also missed in very fast passages, such as arpeggios or thrills (most missed notes in Partita), when they are masked by louder notes (many notes in Humoresque) or due to other factors such as missed onsets and high polyphony. A majority of spurious notes are octave errors, often combined with misjudged repeated notes. These are especially common in pedaled music (Humoresque) or in loud chords (The Entertainer). Other reasons for spurious notes include missed and spurious onsets and errors due to high polyphony.

Some common errors can be seen in a transcription example taken from Humoresque and shown in Fig. 8A (table III/2). Missed notes are marked with a - sign, spurious notes are marked with a + sign. All three spurious notes are octave errors. Out of the two missed notes, A5 was missed, because it is masked by the louder E3C4 chord, while note E3 is a missed repeated note.

Results on real recordings are not as good as those on synthesized recordings. Poorer transcription accuracy is a consequence of several factors. Recordings contain reverberation and more noise, while the sound of real pianos includes beating and sympathetic resonance. Furthermore, performances of piano pieces are much more expressive, they contain increased dynamics, more arpeggios and pedaling. All of these factors make transcription more difficult.

TABLE III
PERFORMANCE STATISTICS OF TRANSCRIPTIONS OF 3 SYNTHESIZED AND 3 REAL PIANO RECORDINGS

	corr. notes	spur. notes	missed notes			spurious notes			num. notes	avg. poly	max. poly
			octave	repeat.	other	octave	repeat.	other			
1	98.1	7	31.4	23.6	56.4	84.4	22.3	7.9	6680	2.7	6
2	92.3	10.6	53.2	39.2	29.4	95.3	29.9	0	1008	4.1	12
3	86	9.5	80.8	25.6	9	96	8.2	5.1	1564	3.4	9
4	88.5	15.5	35.1	18.2	52.2	80.5	17.6	13.9	1351	2.6	6
5	68.3	13.6	30.3	2.1	75.3	79	6.4	20.7	457	4.4	11
6	85.9	15.2	70.3	10.8	27	87.4	7.1	12.3	1564	3.4	9

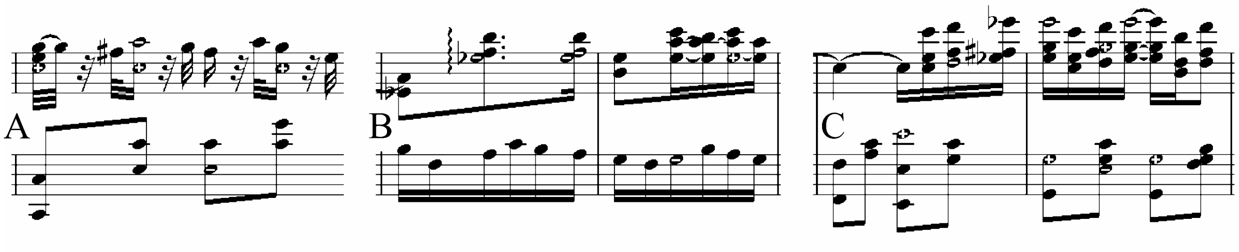

Fig. 8. Transcription examples: A: Humoresque (III/2), B: BWV810 (III/4), C: The Entertainer (III/6).

The analysis of SONIC's performance on the real recording of Bach's English Suite (table III/4, Fig. 8B) showed that besides octave and repeated note errors, most of the missed notes are either quiet low pitched notes (E3 in measure 2, Fig. 8B) or notes in arpeggios and thrills. Chopin's Nocturne (table III/5) proved to be the greatest challenge for our system. The recording is a good example of very expressive playing, where a distinctive melody is accompanied by quiet, sometimes barely audible left hand chords. The system misses over 30% of all notes, but even so the resynthesized transcription sounds very similar to the original (listen to the example on the aforementioned URL address). We compared transcriptions of the real and synthesized version of The Entertainer (table III/3 and III/6, Fig. 8C) and both turned out to be very similar. Transcription of the real recording contains more spurious notes, mostly occurring because of pedaling, which was not used in the synthesized version. The number of correctly found notes is almost the same in both pieces. Octave errors are the main cause of both types of errors.

B. Comparison to Other Approaches

The lack of a standard set of test examples makes comparison of different transcription systems a difficult task, at best. The task is further complicated by the fact that systems put very different constraints on the type or style of music they transcribe. In this section, we present the performance of our system on examples other authors used to evaluate their systems. Note however, that even though we used the same examples as others, comparisons are to be taken with some restraint, as the transcribed pieces were recorded or synthesized under different conditions.

Klapuri [5] developed a system for transcription of polyphonic music. He tested his system on three short passages taken from two piano pieces: J.S. Bach, Inventio 8 and L.V. Beethoven, Fur Elise. Both pieces were recorded on a real piano in a controlled studio environment. Tones of the same piano were previously analyzed and their spectral templates were used in the transcription process. We compared Klapuri's results to the performance of our system on synthesized passages of the same pieces. Results were similar; our system correctly found approximately 2% more notes, but also produced approximately 4% more spurious notes. Most spurious notes were octave errors, which Klapuri managed to reduce by using spectral templates of piano tones in the transcription process. Unfortunately, no results of transcriptions of real piano recordings were published, which would make the comparison of more valid. His system has lately been improved [23], but as to our knowledge it has not yet been evaluated on transcriptions of piano pieces.

Rossi [4] developed a system for transcription of polyphonic piano music. Like Klapuri, Rossi first analyzed the tones of a piano, and then used spectral templates of these tones for transcribing music played on the same piano. She tested her system on three 17th century chorales. SONIC's transcriptions of these pieces contain more spurious notes, all of them octave errors, and a similar number of correctly found notes. Octave errors were removed effectively in Rossi's system by using spectral templates of piano tones. No evaluations of transcriptions of real piano recordings were published to make the comparison more valid.

Sterian [3] developed a system for transcription of music played on brass and woodwind instruments. He published performance statistics of transcriptions of parts of a synthesized and real recording of Bach's Contrapunctus I from The Art of Fugue. Sterian used Kashino's recognition factor R [6] to evaluate the performance of his system; $R=100*(0.5*(\mathit{correct}\text{-}\mathit{spurious})/\mathit{all\_notes}+0.5)$. The accuracy of his system ranged from R=1 to R=0.8 on one to four-voice parts of the synthesized version of Contrapunctus I and from 0.8 to 0.5 on the same parts of the real recording. SONIC's accuracy is better; R ranges from 1 to 0.95 on the synthesized recording and from 0.9 to 0.8 on the real recording of Contrapunctus I (performer V. Feltsman, MusicMasters 67173).

Dixon published preliminary results of his system for transcription of piano music [2]. He made an extensive evaluation of his system on 13 piano sonatas composed by W.A. Mozart. Pieces were played by a real performer, but the recordings were synthesized with different piano samples. When the system was not specifically tuned for the piano sample used, it correctly found 90% of all notes and produced 30% of spurious notes. We were unable to obtain all 13 Mozart sonatas used by Dixon, but the average score of SONIC on seven synthesized Mozart sonatas was significantly better; 92% of notes were correctly found, together with 8% of spurious notes.

VI. Conclusion

In this paper, we presented a connectionist approach to transcription of polyphonic piano music. We first proposed a new model for tracking partials in polyphonic musical signals, based on an auditory model for time-frequency representation and adaptive oscillators for discovery and tracking of partials. By using a connectionist approach, we avoided some of the problems of classical partial tracking approaches, such as missed or spurious peaks, which lead to fragmented or spurious partial tracks, and also showed that our model successfully tracks partials in the case of beating and frequency modulation. An additional advantage of our partial tracking model is that it can be extended to a model for tracking groups of harmonically related partials by joining oscillators into networks. Oscillator networks provide a clearer time-frequency representation of a signal and are especially suitable for transcription purposes. We showed partial tracking with networks of adaptive oscillators significantly improves the accuracy of transcription with time-delay neural networks. We then presented a comparison of several neural network models for note recognition; the best performance was obtained by time-delay neural networks. We presented an overview of our transcription system called SONIC and presented performance statistics of transcriptions of several synthesized and real piano recordings. We also provided a rough comparison of the performance of our system to several others, and showed that it achieves similar or better results. Overall, results show that neural networks present a good alternative in building transcription systems and should be further studied. Further researches will include addition of feedback mechanisms to the currently strictly feedforward approach, with the intention of reducing some common types of errors. Additionally, an extension of the system to transcription of other instruments may be considered.

Acknowledgment

The author would like to thank the reviewers for their suggestions and comments.

References

[1] J.A. Moorer, "On transcription of musical sound by computer," Computer Music Journal vol. 1 no. 4, pp. 32-38, 1977.

[2] S. Dixon, "On the computer recognition of solo piano music," in Proceedings of Australasian Computer Music Conference, Brisbane, Australia, 2000.

[3] A.D. Sterian, Model-based Segmentation of Time-Frequency Images for Musical Transcription. Ph.D. Thesis, Univesity of Michigan, 1999.

[4] L. Rossi, Identification de Sons Polyphoniques de Piano. Ph.D. Thesis, L'Universite de Corse, France, 1998.

[5] A. Klapuri, Automatic Transcription of Music. M.Sc. Thesis, Tampere University of Technology, Finland, 1997.

[6] K. Kashino, K. Nakadai, T. Kinoshita, H. Tanaka, "Application of Bayesian probability network to music scene analysis," in Proceedings of International Joint Conference on AI, Workshop on Computational Auditory Scene Analysis, Monteal, Canada, 1995.

[7] J.C. Brown, "Calculation of a constant Q spectral transform," Journal of Acoustical Society of America, vol. 89 no. 1, pp. 425-434, 1992.

[8] S. Haykin, Neural networks: a comprehensive foundation. New Jersey: Prentice-Hall, 1999.

[9] A. Zell et al, SNNS - Stuttgart Neural Network Simulator v4.2, User Manual. University of Stuttgart, 1997.

[10] A.T. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K.J. Lang, "Phoneme recognition using time-delay neural networks," IEEE Trans. on Acoustics, Speech and Signal Processing vol. 37, no. 3, pp. 328-339, 1989.

[11] I. Gabrijel, A. Dobnikar, "Adaptive RBF Neural Network," D. W. Pearson (Ed.), Proceedings of the Second International ICSC Symposium on Soft Computing, pp. 164-170. Millet, Alberta, Canada, 1997.

[12] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen, "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps," IEEE Transactions on Neural Networks, vol. 3 no. 5, pp. 698-713, 1992.

[13] C. Roads, The Computer Music Tutorial. Cambridge, MA: MIT Press, 1996.

[14] R. D. Patterson, J. Hodsworth, "A functional model of neural activity patterns and auditory images," in Advances in speech, hearing and auditory images 3, W.A. Ainsworth (ed.), London: JAI Press, 1990.

[15] M.Slaney, "An efficient implementation of the Patterson-Holdsworth auditory filterbank," Apple Computer Technical Report #35, 1993.

[16] B.C.J. Moore, B.R. Glasberg, "Suggested formulae for calculating auditory-filter bandwidths and excitation patterns," Journal of the Acoustical Society of America, vol. 74, no. 3, pp. 750-753, 1983.

[17] R. Meddis, "Simulations of mechanical to neural transduction in the auditory receptor," J.Acoust.Soc.Amer., vol. 79, no. 3, pp. 702-711, 1986.

[18] R. Meddis, M.J. Hewitt, "Virtual pitch and phase sensitivity of a computer model of the auditory periphery I: pitch identification," Journal of Acoustical Society of America, vol. 89, no. 6, pp. 2866-2882, 1991.

[19] E.W. Large, J.F. Kolen, "Resonance and the perception of musical meter," Connection Science, no. 6, vol. 2, pp. 177-208, 1994.

[20] J.D. McAuley, Perception of time as phase: toward an adaptive-oscillator model of rhytmic pattern processing. Ph.D. Thesis, Indiana University, 1995.

[21] D. Wang, "Primitive Auditory Segregation Based on Oscillatory Correlation," Cognitive Science, no. 20, pp. 409-456, 1996.

[22] L.S. Smith, "Onset-based Sound Segmentation," in Advances in Neural Information Processing Systems 8, pp. 729-735, Touretzky, Mozer and Haselmo (eds.), Cambridge, MA: MIT Press, 1996.

[23] A. Klapuri, "Multipitch estimation and sound separation by the spectral smoothness principle," Proceedings of International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, USA, 2001.

Matija Marolt (M’96) received the B.S. and Ph.D. degrees, both in computer science, from University of Ljubljana, Slovenia in 1995 and 2002 respectively.

From 1995, he has been working at Laboratory of Computer Graphics and Multimedia, at Faculty of Computer and Information Science, University of Ljubljana, where he is currently Assistant. His research interests include music information retrieval, audio transcription and recognition and audio-visual interaction.