OVERLAPPED EVENT-NOTE SEPARATION BASED ON PARTIALS AMPLITUDE AND PHASE ESTIMATION FOR POLYPHONIC MUSIC TRANSCRIPTION

J.J. Carabias-Orti, P. Vera-Candeas, N. Ruiz-Reyes, F.J. Cañadas-Quesada and R. Mata-Campos

Telecommunication Engineering Department, University of Jaén
C/ Alfonso X el Sabio, 23700, Linares, Spain
phone: +34 953648593, fax: +34 953648508 , email: {carabias, pvera, nicolas, fcanadas, raul}@ujaen.es

ABSTRACT

We propose a discriminative model for polyphonic music transcription that deals with the well-known overlapped partial problem by taking into account the instrument envelope pattern for each note. The process to obtain the music scene-adaptive envelope patterns for each note is detailed. Firstly, spectral features are obtained individually for each note. Then, support vector machines (SVM) are trained on the notes energy. We apply a scheme of one-versus-all (OVA) SVM classifiers to make an approximation of the active frame-level note instances. Finally, amplitudes and phases are estimated by considering the envelope patterns for different notes, distributing the energy according to the note estimated envelope pattern adjustment. Also, temporal information is added by introducing Hidden Markov Models. Our approach has been tested with synthesized and real music recordings, obtaining promising results.

1. INTRODUCTION

The concept of music automatic transcription can be defined as the process of detecting symbolic information (*note-event*) from an audio signal. Such information is related to the high-level musical structures that might be read on a score by a musician [1]. Some interesting applications derived from music automatic transcription are structured audio coding, content-based music retrieval, musicological analysis, WAV-MIDI conversion or audio remixing.

Although the current transcription systems perform well for some signals composed of simultaneous note-events. These systems tends to be inaccurate when these note-events present a rational frequency relation (overlapped partials problem). Also, the notes with harmonic relation between their partials will be referred in this work as overlapped notes.

The employment of instrument-specific information has been widely used in music transcription. Frequency-domain prior knowledge about played instruments benefits the reliability of $f0$ estimation as demonstrated in [2]. The estimation of the typical envelope for different instruments is the common way to introduce frequency-domain information in music transcription systems. Klapuri [3] employ an *ad hoc* envelope for piano transcription. Yin [4] *et al.* introduced an instrument model depending on the MIDI note for the instrument envelope. Also temporal information can be included in the prior information about instruments. Bello *et al.* [1] utilized both frequency and temporal information for automatic piano transcription. A physical model of the piano was also used in [5] to generate spectral patterns that can be compared to the incoming spectral data. This information is learnt for each instrument and pitch from isolated notes on different music database.

2. OVERVIEW

The proposed transcription model is specifically designed to overcome the above mentioned overlapped partial problem by employing instrument-specific information. The main idea is to distribute the energy along the overlapped notes by taking into account the envelope pattern for different notes.

The information about the instrument is directly obtained from the music file we are going to analyze, independently of the music scene in which the instrument is played. For achieving this goal, we use some clues to determine the envelopes that belong to isolated notes. These clues are obtained from perceptual significance, spectral smoothness, stability and distance between pattern envelopes in frequency at different frames of the same note. With this method, we aim to obtain adaptive envelope patterns for monotimbral polyphonic music files.

In figure 1 a block diagram of the system performance is shown.

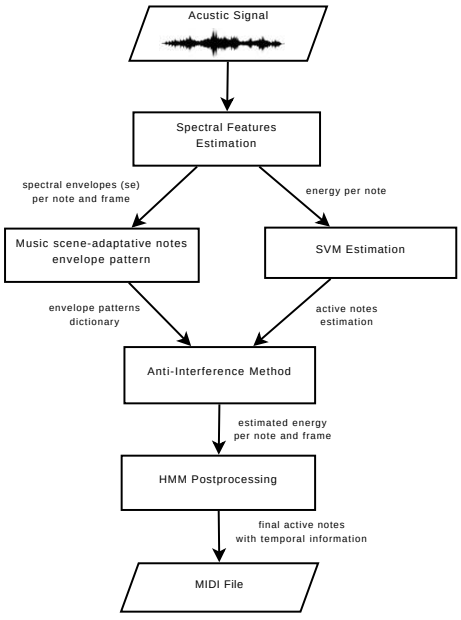

Figure 1: *Block diagram of the proposed approach to separate overlapped notes in a polyphonic music transcription scheme.*

3. MUSIC SCENE-ADAPTIVE NOTES ENVELOPE PATTERN

In this section, we describe the successive steps to estimate notes spectral envelopes in a music scene-adaptive approach. The constraints here are the following:

- Music signals are monotimbral (they contain only one instrument).
- The instruments can be harmonic (i.e. flute, oboe, clarinet, etc.) or slightly inharmonic (i.e. piano, cello, etc.). Nonharmonic instruments (i.e. drums) are not allowed.
- The proposed approach involves an unsupervised method to build the music scene-adaptive dictionary.

When changing the instrument or even the music scene (i.e. the same instrument played in another hall or by a different musician), amplitudes of envelope patterns are adapted to the new scenario in an unsupervised way.

The aim here is to compute a spectral envelope for each note (from 24 to 108 in MIDI scale). For such goal, instrument (or music scene-adaptive) spectral envelopes are computed by taking into account several criteria: perceptual significance, stability, smoothness and spectral similarity.

The process to compute music scene-adaptive spectral envelopes for all notes is fully detailed in [6], also a brief review of the method performance is described bellow.

1. *Spectral analysis and peak tracking*. The windowed DFT is computed using a Hanning window of N=1024 samples (N/f_s=128ms at a sampling rate of f_s=8000 Hz). overlapped with a hop size of 80 samples (10ms) is performed. From the windowed DFTs, a phase vocoder is implemented for spectral peak tracking. As a result, those spectral peaks with phase-continuity are regarded as "tonal trajectories".
2. *Note spectral envelope*. Once the music signal has been analyzed and tonal trajectories identified, we aim to compute the spectral data vectors for all notes as:

$$d_n^t = [d_{n,1}^t, d_{n,2}^t, ..., d_{n,k}^t, ..., d_{n,K}^t];\ n = 24, ..., 108 \quad (1)$$

When a tonal trajectory is identified for note n at the t-th frame, $d_{n,k}^t$ is the complex value with the amplitude and phase for the k-th partial. Otherwise, $d_{n,k}^t = 0;\ \forall k$. Here, we use $K = 10$.
Then, we compute the spectral pattern vectors for each note as follows:

$$p_n^t = \left(\left|d_n^t\right|\right)^2;\ n = 24, ..., 108 \quad (2)$$

From spectral patterns p_n^t, *spectral envelopes* are defined by normalization:

$$se_n^t = \bar{p}_n^t;\ n = 24, ..., 108 \quad (3)$$

fulfilling the constraint $||se_n^t|| = 1$.
3. *Non-audible spectral envelopes removal*. Once spectral envelopes $\{se_n^t;\ n = 24, ..., 108\}$ have been computed, we check whether they are audible or not. Non-audible spectral envelopes are removed. In this work, the masking model described in [7] has been implemented.
4. *Stability checking*. Non-stable spectral envelopes will be discarded. A spectral envelope of a given note is considered to be stable if it fulfils the following conditions:
   (a) The energy of the spectral pattern is above a given threshold. This condition avoids audible spectral envelopes with low energy.
   (b) The spectral envelope does not change above a given threshold in relation to the 10 previous spectral envelopes. The second condition avoids audible spectral envelopes with high time-variation.
5. *Representative spectral envelopes for each note*. After removing non-stable audible spectral envelopes, the following step deals with the overlapped partial problem. If this problem arises, the method proposed in [8] is applied to determine the optimum candidate. This method is based on computing the energy and smoothness of each overlapped spectral envelope. At this stage, a set of representative spectral envelopes for each note is accomplished. The high variability of spectral envelopes for each note involves using some kind of associative algorithm that allows grouping the different types of envelopes for each single note. Unfortunately, it is impossible to know in advance the number of groups. Therefore, we need a density-based clustering method to estimate the different envelope models for each note without any prior knowledge.
6. *Spectral envelope clustering*. In this work, we have used a statistical clustering method based on a Gaussian Mixture Model (GMM) of different probability distributions, estimated by the Expectation-Maximization (EM) algorithm [9]. Here, we have used ten-fold cross-validation to determine the optimum number of clusters. Thus, for each note, one or more clusters (groups of envelopes) will be provided by the algorithm, the centroid of each cluster being a possible envelope pattern for the note.
7. *Final envelope pattern for each note*. After obtaining the different groups of envelopes for each note, the last step consists on providing the representative envelope pattern for each of them. The last step involves three stages:
   (a) *Cluster size filtering*. For each note, the clusters with low number instances are discarded.
   (b) *Selecting only one envelope pattern (cluster) per note*. The second stage is performed by analyzing the similarity of all the envelope pattern clusters for the note with the neighbour notes envelope patterns, when they only have one cluster, choosing the minimum distance envelope pattern is chosen. In other case, we select the cluster with higher number of instances.
   (c) *Solving problems by similarity*. The second stage provides only one envelope pattern per note. However, at this stage, two problems can arise:
      - The envelope pattern of the current note does not seem to that of the nearest neighbours (downward and upward). Here, the envelope pattern is substituted by that of the neighbour note with highest similarity.
      - There can be notes without envelope pattern. Here, the envelope pattern for each "empty" note is taken from the nearest note (downward or upward) having envelope pattern.

Finally, only one spectral envelope se_n for each note is obtained. Then, vector of partial amplitudes a_n is defined from spectral envelope se_n as follows:

$$a_n = \sqrt{se_n};\ n = 24, ..., 108 \quad (4)$$

Vector a_n defines amplitudes for music scene-adaptive note envelope pattern.

4. FRAME-LEVEL NOTE CLASSIFICATION METHOD

4.1 Spectral Features

The information needed for the transcription system is the following:

- The spectral data vectors d_n^t for each note are calculated as in eq. 1.
- The spectral patterns p_n^t for each note are calculated as in eq. 2.
- The energy for each note is computed as:

$$e_n^t = \sum_{k=1}^{K} p_{n,k}^t \quad (5)$$

- The perceptual significance for each note, which is computed as:

$$ps_n^t = 10 \cdot \log_{10} \prod_{k=1}^{K} \frac{p_{n,k}^t}{v_{n,k}^t} \quad (6)$$

where $v_{n,k}^t$ is the Van de Par masking [7] for the current note partial and frame.

Then, the first step would be an approximation to the real transcription by using support vector machines, the explanation will be seen at the next section.

4.2 SVM Estimation

The support vector machine (SVM) [9] is a supervised classification system that uses a hypothesis space of linear functions in a high-dimensional feature space in order to learn separating hyperplanes that are maximally distant form all training patterns.

Separated one-versus-all (OVA) SVM classifiers are trained on the energy for each of the 85 notes (from 24 to 108 in MIDI scale) using the scheme proposed in [10], which gave them good results.

Although the SVM is able to detect most of the active notes, too much noise notes are generated, most of them due to the overlapped partial problem. For this reason, we pursue a method to distribute the energy between the rational frequency relation notes shared partials. This method is explained in the next section.

4.3 Anti-Interference Method

This method is the main contribution of this work. Basically, the method relies on analyzing the adjustment of the calculated note envelope to the corresponding spectral pattern, considering also the possible interferences with other notes in order to distribute the energy between them.

Thus, for each frame, we will consider the notes actives by the SVM classifier as possible candidates. These notes are sorted according to their perceptual significance, from the highest to the lowest one. Then, for each candidate, we will estimate its partials amplitude and phase by analyzing the similarity of the note spectral input data with the corresponding envelope pattern dictionary, composed by the envelope patterns of the candidates with rational frequency relation (overlapped partials) with the current candidate.

The main problem derived from interfering notes is to distribute correctly amplitudes of overlapped partials. These partials are interfered in both amplitude and phase. This complex sums can provoke constructive or destructive interferences in function of the relation between partial phases. In case of constructive interference, the resulting partial has more energy than the sum of energies of isolated partials. But in the destructive case, the resulting energy can become zero. We here propose a novel algorithm that takes partial phases into account in order to distribute overlapped partial energy along the notes with rational frequency relation.

Let us define d_n^t (equation 1) as the spectral input data for note n at frame t. SVM output informs us about the possible active notes at frame t. Using the envelopes estimated in section 3, we have designed a model to distribute spectral energy between active notes. For note n at frame t, its partial amplitudes and phases are generated by the following model,

$$d_n^t = (D \circ e^{j\Phi})\alpha_n^t \quad (7)$$

where $\circ$ is the Hadamard product, D is the dictionary defined from envelope patterns. Also matrix Φ and vector α_n^t are the phases and amplitudes for active notes at frame t. It is supposed that each active note has the envelope pattern estimated in section 3. In this way, only the amplitude that multiply to the normalized envelope pattern must be estimated for obtaining all partial amplitudes. Vector d_n^t has K-length (see eq. 1) where K is the considered number of partials. Matrixes D and Φ have K rows and L columns, L being the number of active notes with harmonic relation with note n. Finally, vector α_n^t has L-length.

In order to clarify equation (7), an illustrated example is introduced in Figure 2. Lets suppose that the active notes according to SVM output at a frame t are 36 and 48 (in MIDI scale), being note 36 the most significant from a perceptual point of view. Consequently, anti-interference method begins with note 36. The spectral input data d_{36}^t for partial amplitudes belonging to note 36 is shown in Figure 2 top plot (circles). Then, the next step consist in analyzing whether other actives notes have any rational frequency relation with the current one. In this way, notes 36 and 48 are octaves, so it exists a 2:1 relation between their partials, and the overlapped partial problem appears. Thus, when anti-interference method is dealing with note 36 and there is one overlapped note ($L = 2$), the matrix D is organized as:

$$D = \begin{pmatrix} a_{36,1} & 0 \\ a_{36,2} & a_{48,1} \\ a_{36,3} & 0 \\ a_{36,4} & a_{48,2} \\ a_{36,5} & 0 \\ a_{36,6} & a_{48,3} \\ a_{36,7} & 0 \\ a_{36,8} & a_{48,4} \\ a_{36,9} & 0 \end{pmatrix} \quad (8)$$

where $a_{n,k}$ contains the partial amplitudes from the estimated envelope patterns of note n (see equation (4)). As can be seen, each row in matrix D represents the overlapped partials structure for the current note in relation with the over-

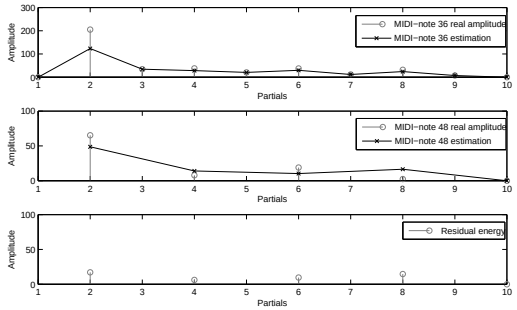

Figure 2: *Anti-interference method performance when the MIDI-notes 36 and 48 are analysed. On the top plot, the MIDI-note 36 original amplitude partials and the envelope adjustment estimation could be seen. On the middle plot, the MIDI-note 48 shared partials amplitude after estimating the MIDI-note 36 and the MIDI-note 48 amplitude estimation is shown. Finally, on bottom plot the residual shared partials amplitude could be seen*

lapped active notes at this frame. The unknown values of our model are phase matrix Φ and amplitude vector α_n^t. Then, a minimization method over equation 7 is applied to obtain the unknown values. However, these values must be initialized in order to avoid too much iterations of the minimization problem. Phase matrix has exactly the same organization as the dictionary D. In our implementation, each row of the phase matrix is initialized to the same value, that is, the phase of the corresponding partial for the spectral data d_n^t. The amplitude vector, α_n^t, is initialized to the correlation between spectral data and the dictionary as we can see in eq. 9.

$$\alpha_n^t = \left(d_n^t\right)^* \cdot D \tag{9}$$

The minimization is performed by the Nelder-Mead simplex (direct search) method. Once the method has estimated phase matrix and amplitude vector, the complex values for note n can be obtained as

$$cv_{n,k}^t = \alpha_{n,1}^t (a_{n,k} \circ e^{j\Phi_{1,k}}) \tag{10}$$

where $cv_{n,k}^t$ is the complex value for the k partial at note n and frame t, $\alpha_{n,1}^t$ and $e^{j\Phi_{1,k}}$ are the estimated envelope amplitude and phases, and $a_{n,k}$ is the dictionary note spectral envelope amplitudes. In Figure 2 top plot (in crosses), the amplitudes of the complex values for note 36 are depicted. Finally, once we have estimated the amplitudes and phases for the current MIDI note, we subtract the estimated complex amplitudes from the spectral input. The resulting partials are shown in Figure 2 middle plot (in circles).

This process is repeated over all active notes (according to SVM output). It must be noted that at each iteration of our model a estimation of the energy for the considered note n at frame t can be obtained. Estimated amplitudes for note 48 are depicted in Figure 2 middle plot (in crosses). The residual amplitudes (the error of the model) are shown in Figure 2 middle plot (in circles). Finally, the estimated energy for each active note at frame t is obtained. We need also to take temporal information into account. For this reason, we considered a Hidden Markov Model that will be explained in the next section.

Algorithm	Acc	E_{tot}	E_{subs}	E_{miss}	E_{fa}
SVM Estimation	37.28%	131.39%	15.84%	1.78%	113.87%
SVM+AIM	46.30%	61.40%	14.61%	21.02%	25.77%
SVM+AIM+HMM	49.62%	53.42%	13.32%	21.51%	18.59%
Poliner and Ellis	67.7%	34.2%	5.3%	12.1%	16.8%
Ryynänen and Klapuri	46.6%	52.3%	15.0%	26.2%	11.1%
Marolt	36.9%	65.7%	19.3%	30.9%	15.4%

Table 1: *Frame-level transcription results on recorded piano test set. Comparison between our system and more recent state of the art transcription systems is performed with the results presented in [10].*

4.4 HMM Processing Stage

In order to include the inherent temporal information associated to the music, a two states (on/off) HMM model is used for each note. The idea is not to deal with polyphony but also to introduce an inertia factor to avoid undesirable activations of notes without continuity. Therefore, if the model state at time t is given by q_t , and the classifier output label is c_t , then we also need to find the most likely Viterbi state sequence which maximize,

$$\prod_t p(c_t|q_t)p(q_{t-1}) \tag{11}$$

where $p(q_t|q_{t-1})$ is the transition matrix, which as same as the state priors, is estimated from training data by analyzing the ground-truth transcriptions.

To estimate $p(c_t|q_t)$ we will apply the same method that in [10]. That is, we will assume the probability gave by the final energy estimation to be similar to the real state dynamics. Hence, if the acoustic data at each time is x_t , we may regard our active notes energy as giving us estimates of:

$$p(q_t|x_t) \propto p(x_t|q_t)p(q_t) \tag{12}$$

that is, the state dynamics of each HMM.

Finally, by dividing each estimated state dynamics by the prior of the corresponding note, we get scaled likelihoods that can be employed directly in the Viterbi search for the solution of equation 11.

5. RESULTS

Our database is composed of several monoaural live piano recording WAV files with a sampling rate of 8000 kHz and their corresponding MIDI files, that have been provided by [10] at http://labrosa.ee.columbia.edu/projects/piano. Within the database, we have 19 MIDI excerpts for training, and 10 for test, with 1 minute duration each one. This database is introduced here because comparison between more recent state of the art transcription systems is performed with this one in [10].

We have used the same metrics of accuracy and error score described in [10] to evaluate the performance of our note-detection approach.

As displayed in table 1, our results are better than the systems proposed by Ryyänen [11] and Marolt [12], but still far from the best performance system [10]. Also, we can see the improvement made by the anti-interference method and with the addition of temporal information.

The most important characteristic of our system is that it is not adjusted to a single instrument in a concrete scene, due to the adaptive envelope estimation. Thus, in order to check the robustness of the proposed method, the test database in

[10] has been synthesized with other instrument (using a ROLAND FANTOM-XR synthesizer). The instrument we have chosen for resynthesizing is trumpet because it exhibits a completely different spectral envelope with respect to piano.

Transcription results for the resynthesized database are shown in Figure 3, as can be seen, the system achieves better accuracy and error measures regarding to piano test database. The good performance for trumpet instrument is due to several reasons: synthetic files instead real ones, higher spectral smoothness and the non-existent of reverberation in wind instruments. We can conclude that our unsupervised process for obtaining spectral patterns does not depend on the target instrument.

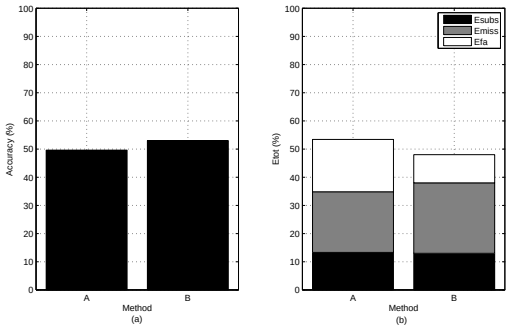

Figure 3: *Frame-level transcription results for the two instrument test databases. In the left panel (a), each stack corresponds to the accuracy rates using our proposed method both (A) Piano test database and (B) Trumpet test database. In the right panel (b), each stack corresponds to the total error rate which is broken down into substitution errors, miss errors and false alarm errors. Our proposal shows an accurate and robust behaviour independent from the music instrument.*

6. CONCLUSIONS AND FUTURE WORK

In this work, music scene-adaptive notes envelope patterns are applied to polyphonic music transcription. Thus, for each note, the final partial amplitudes and its phases are estimated by analyzing the calculated envelope of the current note and the envelopes of the notes with shared partials, that is, the notes with rational frequency relation with the current one. This transcription application obtains promising results in terms of accuracy and error measures when is compared with state-of-art transcription approaches. We have also demonstrated the robustness of our approach by changing the instrument (trumpet instead piano), resynthesizing the audio database and evaluating again transcription measures.

Although our method present a good approximation to resolve the overlapped partial problem by using instrument pattern adjust to distribute the energy along the overlapped partial notes, several issues have to be considered for next approaches.

- First of all, it would be necessary to change the scheme to analyze the all notes at the same time. That is, not to analyze from the note with highest perceptual significance to the lowest one. In this way, we are planning to test another method that compute all the possible solutions, choosing the configuration that minimizes the distributed energy along the notes.
- Another issue to take into account is to analyze the phase evolution, detecting the possible cases of negative or positive interference, in order to improve the energy distribution along the overlapped partials.

7. ACKNOWLEDGMENT

This work was supported by FEDER, the Spanish Ministry of Science and Innovation under Project TEC2006-13883-C04-03, and the Andalusian Business, Science and Innovation Council under project P07-TIC-02713.

REFERENCES

[1] Bello, J., Daudet, L. and Sandler, M. "Automatic piano transcription using frequency and time-domain information", *I*EEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 6, pp. 2242–2251, November, 2006.

[2] Goto, M. "A predominant-f0 estimation method for CD recordings: MAP estimation using EM algorithm for adaptive tone models", *Proc. ICASSP'01*, vol. 5, pp. 3365-3368, May, 2001.

[3] Klapuri, A. "Multiple fundamental frequency estimation by summing harmonic amplitudes", *Proc. ISMIR'06*, pp. 216-221, Victoria, Canada, 2006.

[4] Yin, J., Sim, T., Wang, Y. and Shenoy, A. "Music transcription using an instrument model", *Proc. ICASSP'05*, vol. 3, pp. 217-220, March, 2005.

[5] Ortiz-Berenguer, L.I., Casajus-Quiros, F.J. and Torres-Guijarro, S. "Multiple piano note identification using spectral matching method with derived patterns", *J*ournal of the Audio Engineering Society, vol. 53, no. 1/2, pp. 32-43, January/February, 2005.

[6] Carabias-Orti, J.J., Vera-Candeas, P., Ruiz-Reyes, N., Caadas-Quesada, F.J., and Cabaas-Molero, P. "Estimating Instrument Spectral Envelopes for Polyphonic Music Transcription in a Music Scene-Adaptive Approach", *A*ES 126th Convention, Munich, Germany, May, 2009.

[7] Van de Par, S., Kohlrausch, A., Charestan, G. and Heusdens, R. "A new psycho-acoustical masking model for audio coding applications", *IEEE ICASSP'02*, pp. 1805-1808, May, 2002.

[8] Pertusa, A. and Inesta, J.M. "Multiple fundamental frequency estimation using Gaussian smoothness". *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pp. 105-108, 2008.

[9] Witten, I.H. and Frank, E. "Data mining: practical machine learning tools and techniques", 2nd Edition, *Morgan Kaufmann*, San Francisco, 2005.

[10] Poliner, G. and Ellis, D. "A discriminative model for polyphonic piano transcription", *EURASIP Journal on Advances in Signal Processing*, 2007.

[11] Rynnänen, M and Klapuri M. "A connectionist approach to automatic transcription of polyphonic piano music", *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 439-449, 2004.

[12] Marolt, M. "A connectionist approach to automatic transcription of polyphonic piano music", *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 439-449, 2004.