IMPROVEMENTS OF CONTINUOUS MODEL FOR MEMORY-BASED AUTOMATIC MUSIC TRANSCRIPTION

Štěpán Albrecht[1], *Václav Šmídl*[2]

[1]University of West Bohemia, Plzeň, Czech Republic, albrs@kiv.zcu.cz
[2] Institute of Information Theory and Automation, Prague, Czech Republic, smidl@utia.cas.cz

ABSTRACT

Automatic music transcription is a process recovering the most likely combination of sounds that produced the recorded audio signal. We are concerned with memory-based approach, where the observed signal is modeled as a superposition of sounds from a library. Moreover, we assume that only parts of the sounds can be played. The number of possible combinations is excessive and exact estimation is computationally prohibitive. We propose to transform the original discrete-event model into a less restricted parametrization and impose the constraints in a soft way via prior information. The resulting model is a non-linear state-space model with Gaussian disturbances. The posterior estimates are evaluated by the extended Kalman filter. Performance of the model is studied in simulation and it is shown that it outperforms previously published methods.

1. INTRODUCTION

Automatic music transcription (AMT) is a process of decomposing recorded music signal into a sequence of higher-level sound events. The entire AMT—i.e. resolving pitch, loudness, timing and instrument of all sound events in an input audio music signal [6]—is not theoretically possible in general [6], therefore practical AMT has to be restricted to a specific scenario. Commonly used scenarios are memory-based and data-based AMT. The former utilizes sound models corresponding to a certain musical instrument sound (allowing to identify the instruments), the latter utilizes only rules which hold in general. We are concerned with a special case of memory-based AMT. Kashino's transcription system [9] is another system that is considered as an entire memory-based AMT system in the sense of [6].

Intuitively, the problem can be understood as an 'inverse music sequencer', Fig. 1. Music sequencers have a pre-recorded library of sounds (sound components) which are combined together to create music signal. Input to the sequencer is a MIDI file which contains information about beginning of music events in time, their duration, IDs of sounds (in our case the pre-recorded sound components), their amplitude and modification type. Component modification(s)—e.g. component truncation or pitch shifting—were designed to reduce the size of the pre-recorded library. In this paper we consider only component truncation as a possible modification. Output of the sequencer is the audio signal. Input of our 'inverse music sequencer' is the recorded music signal and its output is the estimated (transcribed) MIDI-like representation of music events.

The sequencer composes the output from sounds stored in the library of K sounds. Each sound is composed of L_k frames, which are supposed to be played after each other.

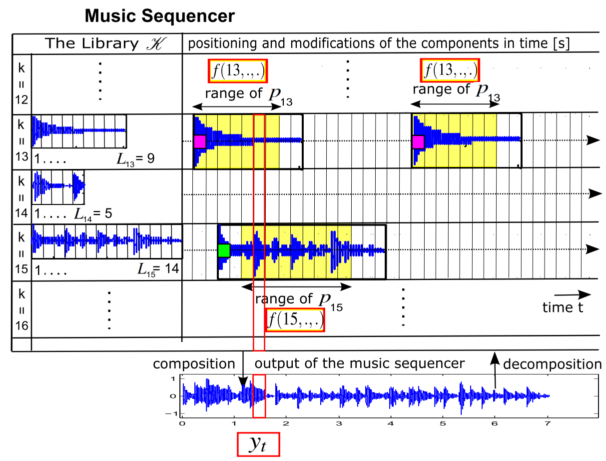

Figure 1: Principle of a music sequencer. The range of active frames $p_\kappa$is yellowed. Note that the amplitudes are the same for all events in a track k (represented by squares of the same color).

The input events are defined by: (i) index of the sound to play, $k=[1,\ldots,K]$, (ii) truncation of the sound, i.e. beginning of the range of frames from the kth sound to play, $\boldsymbol{p}$, and (iii) amplitude of the sound, $0 \leq g \leq 1$. We assume that each sound can be played only once at time t. The output sound is then:

$$\boldsymbol{y}_t = \sum_{\kappa \in \mathscr{K}_t} g_\kappa f(k_\kappa, \boldsymbol{p}_\kappa, t), \tag{1}$$

where $\boldsymbol{y}_t$ is the ϕ-dimensional vector of measurements at time t composed of either time- or frequency-representation of the input music signal segment (frame); κ denotes ID of the event from the set of events active at time t, $\mathscr{K}_t \subset [1,\ldots,K]$. Function $f(k_\kappa, \boldsymbol{p}_\kappa, t)$ looks up the frame from range $\boldsymbol{p}_\kappa$ of the k_κth sound that is active in time t, see illustration on Fig. 1.

Model (1) is a suitable representation of a sequencer, however, it is not suitable for the inverse operation since the number of possible configurations of the unknowns $\mathscr{K}_t$ and $\boldsymbol{p}_\kappa$ is enormous. Formally, (1) can be written as a sum over all frames

$$\boldsymbol{y}_t \quad = \quad \sum_{i=1}^{N} \alpha_{i,t} g_{i,t} \boldsymbol{f}_i, \tag{2}$$

where $N=\sum_{k=1}^{K} L_k$, $\alpha_{i,t} \in \{0,1\}$ is equal to 1 if ith frame, $\boldsymbol{f}_i$, is used in (3) and $g_{i,t}$ is the corresponding amplitude. The values of $\alpha_{i,t}$ are constrained by the parameters $\mathscr{K}_t$ and $\boldsymbol{p}_\kappa$ as follows:

- only one frame of the kth sound may be active at time t,
- when the ith frame of the kth sound was active in time $t-1$ and t is in $\boldsymbol{p}_\kappa$, the $(i+1)$th frame must be active in time t,
- no frame is active when t is out of $\boldsymbol{p}_\kappa$.

The number of possible combinations of $\alpha_{i,t}$ is still enormous, since we allow arbitrary truncations of the sounds. Therefore, we propose to relax the hard constraints above and introduce an unconstrained variable $0 \leq a_{i,t}$ such that

$$\boldsymbol{y}_t = \sum_{i=1}^{N} a_{i,t} \boldsymbol{f}_i = \boldsymbol{F} \boldsymbol{a}_t. \tag{3}$$

where $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N]$ and $\boldsymbol{a}_t = [a_{1,t}, \ldots, a_{N,t}]'$, $a_{i,t}$ being the amplitude of ith frame. This relaxation has both advantages and disadvantages.

The advantage is that model (3) is well studied in statistical literature and efficient parameter estimation methods exists for its various variants. For example, linear regression, factor analysis [3], Kalman filtering, matching pursuit [10] and independent component analysis [5] (ICA) arise from (3) by imposing different assumptions on parameters $\boldsymbol{a}_t$ and $\boldsymbol{F}$. These methods are used in music processing, e.g. ICA for blind (unsupervised) source separation (BSS) techniques in monoaural input music signals [6]. In this work, we investigate the use of the Kalman filtering approach.

The main disadvantage of the relaxation is that it allows to explain signal $\boldsymbol{y}_t$ by a combination of frames that are not valid from musical point of view (e.g. it allows to play all frames from one sound at the same time). The original restrictions can be restored in less restrictive form via transition model $p(\boldsymbol{a}_t|\boldsymbol{a}_{t-1})$ which needs to be designed. Similar approach was used in [2] where the prior was designed for the whole sequence by optimized combination of priors commonly used in the area. In this paper, we present a new model using only first-order Markov transition model which is obtained by conversion of transition model for discrete events (1) into a form suitable for the continuous model (3).

The paper is organized as follows: model of the signal transition between time frames is presented in the second Section; evaluation of the posterior density implied by the model is presented in the third Section; simulation study that assess performance on the model on music data is in the fourth section.

2. DERIVATION OF THE MODEL

Observation of the signal $\boldsymbol{y}_t$ are never perfect due to round-off errors and measurement noise. The observation model (3) is used as mean value of Gaussian likelihood function of observations:

$$p(\boldsymbol{y}_t|\boldsymbol{a}_t, \boldsymbol{F}) = \mathcal{N}(\boldsymbol{F}\boldsymbol{a}_t, \omega^{-1}\boldsymbol{I}_\phi). \tag{4}$$

Here, $\mathcal{N}$ denotes normal distribution of vector argument, ω is scalar precision parameter, $\boldsymbol{I}_\phi$ denotes identity matrix of dimensions $\phi \times \phi$.

The task is to estimate posterior density of $\boldsymbol{a}_t$ given available data, $p(\boldsymbol{a}_t|\boldsymbol{F}, \boldsymbol{Y}_t)$, where $\boldsymbol{Y}_t = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t]$. The constraints on α will be transformed into Gaussian prior $p(\boldsymbol{a}_t|\boldsymbol{a}_{t-1})$, which is parametrized by mean value of size N and covariance matrix of size $N \times N$.

2.1 Transformation between $\alpha_{i,t}$ and $a_{i,t}$

We start with a simple transformation between discrete variable $\boldsymbol{\alpha}_t$ and continuous amplitude $\boldsymbol{a}_t$, specifically

$$p(a_{i,t}|\alpha_{i,t}) = \begin{cases} \mathcal{N}(1, k\sigma_1) & \text{if } \alpha_{i,t} = 1, \\ \mathcal{N}(0, \sigma_1) & \text{otherwise.} \end{cases} \tag{5}$$

Intuitively, zero values of $\alpha_{i,t}$ (i.e., representation of silence) are mapped on $a_{i,t}$ which are 'close to zero' and $\alpha_{i,t} = 1$ (i.e., the loudest sound notation) are mapped to $a_{i,t}$ close to 1. The closeness is modeled by variance parameter σ_1. Since we allow lower amplitudes of the tone via g, we model variance of the first component of the pdf in (5) to be k times greater than that of the second component.

Inverse mapping of $\boldsymbol{a}_t$ to $\boldsymbol{\alpha}_t$ can be obtained by the Bayes rule:

$$p(\alpha_{i,t}|a_{i,t}) = p(a_{i,t}|\alpha_{i,t})p(\alpha_{i,t})/p(a_{i,t}).$$

There is no information on prior of $\alpha_{i,t}$, thus $p(\alpha_{i,t})$ is uniform, and for a particular component:

$$\begin{aligned} p(\alpha_{i,t} = 0|a_{i,t}) &\propto \\ &\frac{\frac{1}{2}\sigma_1^{-0.5}\exp(-\frac{1}{2\sigma_1}a_{i,t}^2)}{\frac{1}{2}\sigma_1^{-0.5}\left[\exp(-\frac{1}{2\sigma_1}a_{i,t}^2) + k^{-0.5}\exp(-\frac{1}{2k\sigma_1}(1-a_{i,t})^2)\right]} \\ &= \frac{1}{1 + k^{-0.5}\exp(-\frac{1}{2k\sigma_1}((1-k)a_{i,t}^2 - 2a_{i,t} + 1))} \end{aligned} \tag{6}$$

2.2 Parameter evolution model

In the discrete parametrization (1), the transition between frames can be modeled by a simple Markov transition:

$p(\alpha_{i,t}\|\alpha_{i-1,t-1})$	$\alpha_{i-1,t-1} = 0$	$\alpha_{i-1,t-1} = 1$
$\alpha_{i,1} = 0$	τ_0	$1-\tau_0$
$\alpha_{i,1} = 1$	$1-\tau_1$	τ_1

where τ_0, τ_1 are constant probabilities that the discrete amplitude is not changed by the transition from $t-1$ to t . This transition model can be combined with (6) as follows:

$$\begin{aligned} p(a_{i,t}|a_{i-1,t-1}) = &\sum_{\alpha_{i,t-1}} \\ &\sum_{\alpha_{i-1,t-1}} p(a_{i,t}|\alpha_{i,t})p(\alpha_{i,t}|\alpha_{i-1,t-1})p(\alpha_{i-1,t-1}|a_{i-1,t-1}) \end{aligned} \tag{7}$$

However, direct application of this rule would result in prior $p(\boldsymbol{a}_t)$ being a mixture of 4^{NT} components which is not computationally tractable. Hence, we project (7) into a single Gaussian density

$$p(a_{i,t}|a_{i-1,t-1}) = \mathcal{N}(\mu_{i,t-1}, \sigma_{i,t-1}) \tag{8}$$

using geometric merging of probabilities [7], which yields

$$\begin{aligned} \sigma_{i,t-1}^{-1} &= \hat{\alpha}_{i,t}\frac{(k-1)\tau_0 + 1}{k\sigma_1} + (1-\hat{\alpha}_{i,t})\frac{(k-1)(1-\tau_1)+1}{k\sigma_1}, \\ &= \frac{\hat{\alpha}_{i,t}(k-1)(\tau_0+\tau_1-1) + (k-1)(1-\tau_1)+1}{k\sigma_1} \end{aligned} \tag{9}$$

$$\begin{aligned} \mu_{i,t-1} &= \sigma_{i,t-1}\left(\hat{\alpha}_{i,t}\frac{(1-\tau_0)}{k\sigma_1} + (1-\hat{\alpha}_{i,t})\frac{\tau_1}{k\sigma_1}\right). \\ &= \frac{\hat{\alpha}_{i,t}(1-\tau_0+\tau_1)+1}{\hat{\alpha}_{i,t}(k-1)(\tau_0+\tau_1-1)+(k-1)(1-\tau_1)+1} \end{aligned} \tag{10}$$

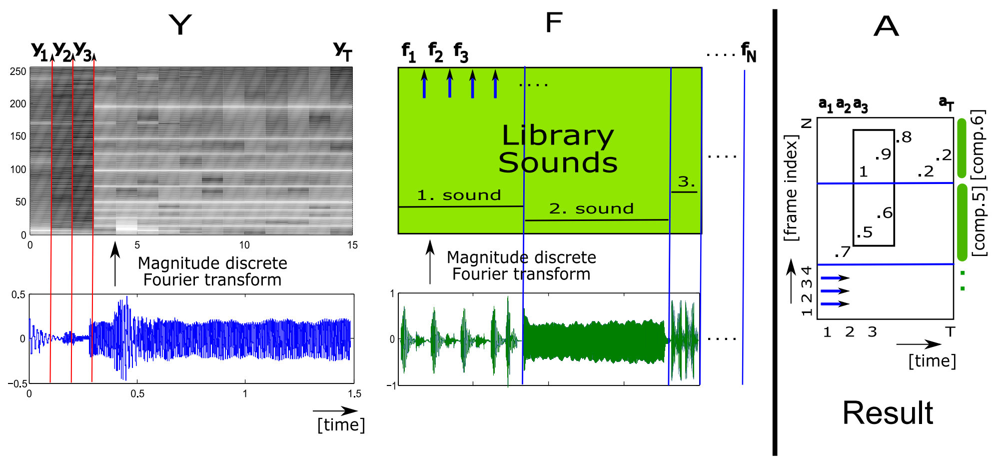

Figure 2: Visualization of matrix notation of the audio model.

where $\hat{\alpha}_{i,t} = p(\alpha_{i,t} = 0|a_{i,t})$ from (6).

Prior (7) is valid only when frame with index $i-1$ is in the same sound. First frames in the sound are treated in a special way. The probability of the whole vector is then:

$$p(\boldsymbol{a}_t|\boldsymbol{a}_{t-1}) = \mathcal{N}(h(\boldsymbol{a}_{t-1}), \boldsymbol{Q}_t(\boldsymbol{a}_{t-1})), \quad (11)$$
$$p(\boldsymbol{y}_t|\boldsymbol{a}_t) = \mathcal{N}(\boldsymbol{F}\boldsymbol{a}_t, \omega^{-1}\boldsymbol{I}_\phi). \quad (12)$$

Here, $h(\boldsymbol{a}_{t-1})$ is a vector valued function,

$$h_i(\boldsymbol{a}_{t-1}) = \begin{cases} \mu_{i,t}(\boldsymbol{a}_{t-1}) & \text{if } i, i-1 \text{ in the same sound} \\ c & \text{otherwise} \end{cases}, \quad (13)$$

and $\boldsymbol{Q}_t$ is a diagonal matrix

$$Q_{i,i,t}(\boldsymbol{a}_{t-1}) = \begin{cases} \sigma_{i,t}(\boldsymbol{a}_{t-1}) & \text{if } i, i-1 \text{ in the same sound} \\ q & \text{otherwise} \end{cases}. \quad (14)$$

Here c, q denote constants on the positions of first frames of the library sounds, for illustration of the relation between i and sounds see Fig. 1.

3. BAYESIAN FILTERING OF THE MODEL

The state-space model derived in Section 2 is strongly nonlinear model with Gaussian disturbances. There is a range of techniques for Bayesian filtering, such as particle filters [4], extended Kalman filters, and others. The model was derived using projection into Gaussian densities, hence a filter designed for Gaussian disturbances seems to be appropriate choice. For the purpose of this paper, we will use a version of the extended Kalman filter (EKF).

The task is to recursively compute posterior density $p(\boldsymbol{a}_t|\boldsymbol{Y}_t)$ which is, in the EKF, approximated by a Gaussian

$$p(\boldsymbol{a}_t|\boldsymbol{Y}_t) = \mathcal{N}(h(\hat{\boldsymbol{a}}_{t-1}) - \boldsymbol{K}(\boldsymbol{y}_t - \boldsymbol{F}\hat{\boldsymbol{a}}_{t-1}), \boldsymbol{P}_{t,t}),$$

where $\hat{\boldsymbol{a}}_{t-1}$ is a mean value of the previous density $p(\boldsymbol{a}_{t-1}|\boldsymbol{Y}_{t-1})$ and matrices $\boldsymbol{K}$ and $\boldsymbol{P}_{t,t}$ are computed using the standard EKF as follows:

$$\begin{aligned} \boldsymbol{R}_y &= \boldsymbol{F}'\boldsymbol{P}_{t-1}\boldsymbol{F} + \omega^{-1}\boldsymbol{I}_\phi, \\ \boldsymbol{K} &= \boldsymbol{P}_{t-1}\boldsymbol{F}\boldsymbol{R}_y^{-1} \\ \boldsymbol{P}_{t|t} &= \boldsymbol{P}_{t-1} - \boldsymbol{P}_{t-1}\boldsymbol{F}'\boldsymbol{R}_y^{-1}\boldsymbol{F}\boldsymbol{P}_{t-1}, \\ \boldsymbol{P}_t &= \boldsymbol{A}\boldsymbol{P}_{t|t}\boldsymbol{A} + \boldsymbol{Q}_t(\hat{\boldsymbol{a}}_{t-1}). \end{aligned} \quad (15)$$

Here, $\boldsymbol{A} = \frac{d}{d\boldsymbol{a}_{t-1}} h(\boldsymbol{a}_{t-1})$ which is a sparse matrix composed of derivatives of $\mu_{i,t}$ (10)

$$\frac{d}{da_{i,t-1}}\mu_{i,t-1} = \frac{(-1+t_0+t_1)\sqrt{k}\,((k-1)a_{i,t-1}+1)\,\varepsilon\sigma_1{}^{-1}}{\left(k^{3/2}t_0 + \sqrt{k} - \sqrt{k}t_0 + \varepsilon(k - kt_1 + t_1)\right)^2}$$

$$\varepsilon = \exp\frac{1}{2}\frac{(k-1)a_{i,t-1}^2 + 2a_{i,t-1} - 1}{k\sigma_1}$$

Note that $Q_t(\boldsymbol{a}_{t-1})$ in (11) was replaced by $Q_t(\hat{\boldsymbol{a}}_{t-1})$ in (15). This change is required since EKF does not allow covariance matrices to be function of the state variable. We conjecture that this is an acceptable approximation.

4. EXPERIMENT

The simulated data were generated from piano midi files. Each note was represented by pitch, onset time, duration and offset in the sound library. The offset is a non-standard extension of the midi format. The corresponding amplitude matrix and the observed audio signal were generated using model (1). Midi notes that were not available in the library of sounds were omitted. For testing purposes, 61 library sounds (corresponding to midi notes 36—96) were used, each of them having 10 frames. Each frame contained 4096 samples at 44.1 kHz sample rate, represented by the magnitude spectrum. For training of the nuisance parameters, only 36 (midi notes 45—80) sounds were considered. Thus, there were 610 and 360 frames in the testing and training library, respectively. The sounds assigned to the piano midi events were obtained by a harmonic tone synthesizer [1] which produce tones with sharp attack and inner frames of different loudness, however, the frames were significantly similar to each other. Hence, the audio signal generated by the first frame at low amplitude is remarkably similar e.g. to that of the third frame at higher amplitude. This is a challenge for estimation, since the likelihood model alone can not properly distinguish those two cases and good model of the prior is required to obtain good performance.

The proposed model contains nuisance parameters $\delta_1 = [\sigma_1, k, \tau_0, \tau_1, c, q]$ in the apriori part and ω_1 in the likelihood. These were optimized by Matlab function `fminsearch` using the following criteria: (i) a measure similar to the total relative sound-to-distortion ratio [8] that read:

$$SDR = 10log_{10}\frac{\sum_t [b \cdot \boldsymbol{F}_{acoust}\boldsymbol{a}_t]^2}{\sum_t [\boldsymbol{y}_t - b \cdot \boldsymbol{F}_{acoust}\boldsymbol{a}_t]^2}, \qquad (16)$$

where b is a scalar fitting $b \cdot \boldsymbol{A} = \boldsymbol{A}_{reference} + noise$ according to MMSE, and F_{acoust} is the matrix of frames in acoustic form; and (ii) a hit-measure: $m = hits - 0.5 \cdot (falsepositive + falsenegative)$. Model nuisance parameters were trained on 51 frame long signal of one of Debussy's preludes and tested on 582 frame long concatenation of short excerpts of Mozart and Debussy. In the training phase, 51 units were filtered by the Kalman filter to a selected optimization criteria value, frame by frame with no overlap. The SDR criteria was found to be more suitable for optimization since the hit-measure is too coarse for the `fminsearch` optimization. Moreover, the hit-measure depends on the amplitude threshold to distinguish active from non-active $a_{i,t}$ amplitudes whereas the SDR does not. All results presented in the paper are based on nuisance parameters optimized for the SDR criterion. In the testing phase, 58 seconds of music audio signal containing 1325 active frames were estimated by the Kalman filter.

For comparison, two previously published methods have been applied to the same data. The first approach, labeled 'maxent', is based on model (3) with different prior [2]. The prior is obtained by optimized combination of four phenomena: A) sparsity; B), C) temporal dependence; D) dissimilarity of simultaneous sounds. Combination of these phenomena was governed by nuisance parameters $\delta_2 = [\lambda, \gamma, c, \nu_1, \nu_2]$ and ω_2, which were optimized using the same `fminsearch` procedure. The original δ_2 from [2] contained additional parameter ζ, which was found to be redundant. The second compared method, labeled 'NMF', is non-negative matrix factorization of the measurement matrix $\boldsymbol{Y}$ [6], where the matrix of bases corresponds to F that is known. Even though there are 'NMFs' with various restrictions on amplitude matrix, the considered 'NMF' transcription uses no prior knowledge (i.e., no restrictions) to demonstrate the informativeness of the independent measurements.

Resulting transcriptions of all three tested methods are displayed via piano-roll schematics in Figures 3 (detail of note E 64) and 4 (initial 25 samples of the testing set). Note that the above mentioned ambiguity of the likelihood is well manifested in the results of NMF approach (Fig. 4, bottom-right) which is based only on the likelihood. Models of prior information (maxent or the current model) improve the estimation results by sharpening around the most likely path. However, the ambiguity is affecting these methods as well, since one missed frame may lead to a postponement of the whole tone, see detail of the posterior in Fig. 3. This shift in time has negative influence on the hits factor of the current method as summarized in Table 1. The maxent model was poor in estimation of the length of a sounding note. Almost all lengths were estimated identically despite their variablity.

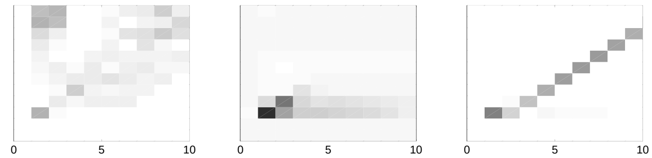

Figure 3: Focus of the note E 64 from Fig.4: posterior density of matrix A from NMF (left), using maxent model (middle), and the current model (right). The simulated value was a straight line with one vertical step increase in each horizontal step.

Table 1: Comparison of the presented model with previous methods.

	active frames total	hits	false positive	false negative	SDR [dB]
current model	1325	1219	293	106	10.59
maxent model	1325	1038	257	557	8.72
NMF	1325	1007	1367	218	3.53

This led either to their over-estimation or under-estimation according to the tested data. Over-estimation of the lengths causes only minor decrease of the SDR values since amplitudes of the tones at their ends are small.

Using library of those 61 sounds, one time unit processing ranged from 1.5 to 2 with Kalman filter on Core Duo or Quad Core CPU. Hundred of iterations of the 'NMF' algorithm took about 30 seconds, thus the implementation of the problem solution by the 'NMF' was about 50 times faster than the problem solution by Kalman filter.

5. DISCUSSION AND CONCLUSION

We have presented a new model with continuous parametrization for automatic music transcription. The main motivation of the new prior is on-line transcription of the signal using only first-order Markov transition model. The underlying model of discrete events was transformed into continuous version via Gaussian mixture models. Projection of these mixtures into a single Gaussian density yields non-linear state-space model with Gaussian disturbances. Music transcription is then converted into estimation of the state variable which is achieved by the extended Kalman filter with a minor modification. The nuisance parameters were tuned on a small training set, while the final comparison was performed on a significantly larger data-set. The results compare favorably to the previously published approaches. Note that the transcription is obtained on-line, i.e. each point was estimated using only data available up to the time of the analysis. It can be expected that extensions using Bayesian smoothing would further improve on the quality of the estimates. At present, the Kalman gain calculation is rather expensive—one step takes about one second in this case—but there is a lot of space for optimization or approximate evaluation employing e.g. the ensemble Kalman filter. Further improvement can be obtained by extension of the prior to higher-order Markov model. In this paper, we have considered only transition between the

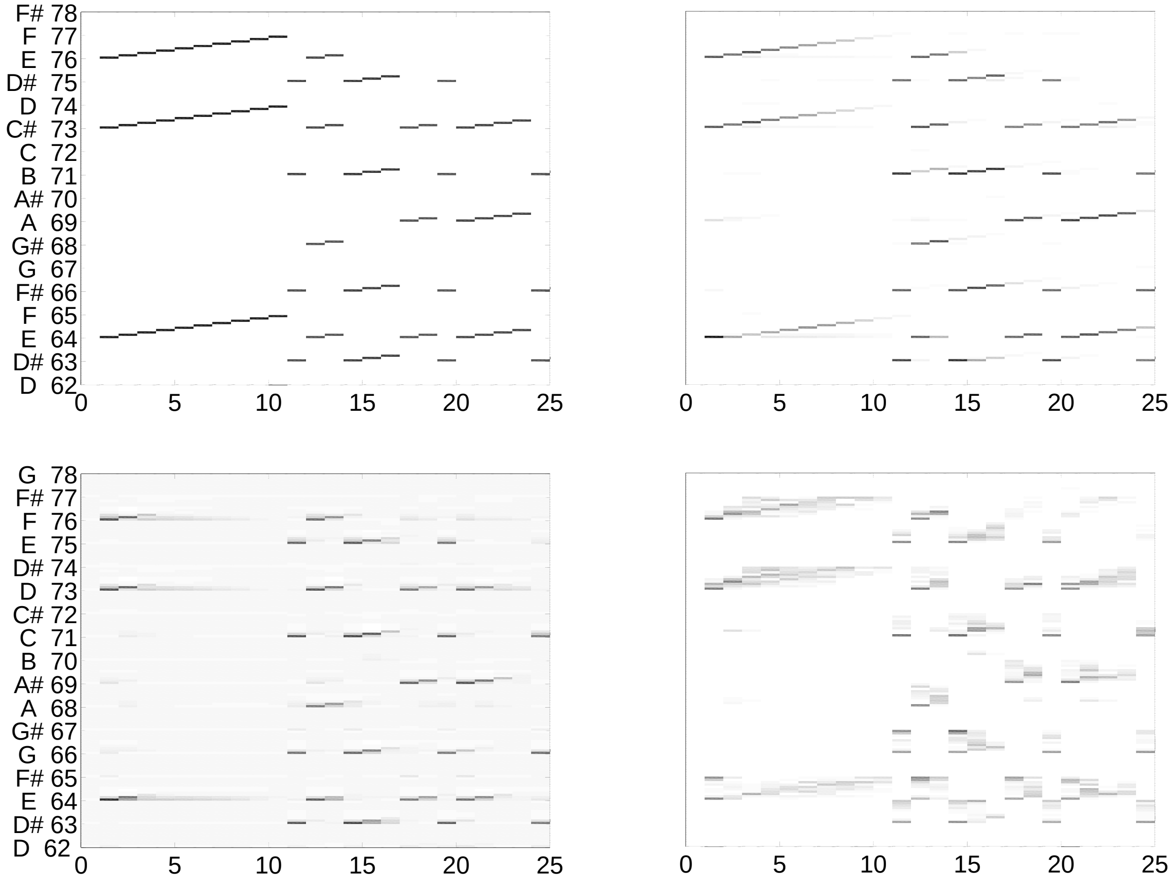

Figure 4: Example of simulated and transcribed piece of polyphonic music. **Top-left**: original music excerpt; **top-right**: the excerpt transcribed by the current model with optimized nuisance $\delta_1 = [4, 3 \cdot 10^{-4}, 59, 0.88, 0.93, 0.26, 0.17]$, $\omega_1 = 0.69$; **bottom-left**: the maxent model, optimized nuisance $\delta_2 = [0.005, 1.0, 4 \cdot 10^{-6}, 0.26, 10^{-4}]$, $\omega_2 = 1.0$; **bottom-right**: transcription by NMF without any constraints. Vertical axis denotes tone with the due midi keys. The horizontal axis denotes discrete time (time units). Focused depiction of one note is displayed in Fig. 3.

two consecutive frames in the bank of sounds. Clearly, the approach can be extended for 3 and more frames.

Acknowledgment

Support of grant GA ČR 102/08/P250 is gratefully acknowledged.

REFERENCES

[1] http://www.kenschutte.com/midi.

[2] Š. Albrecht and V. Šmídl. Model considerations for memory-based automatic music transcription. In *29th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, Oxford, Mississipi, US, 2009.

[3] T.W. Anderson. *An Introduction to Multivariate Statistical Analysis*. John Wiley, 1958.

[4] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 50(2):174–188, 2002.

[5] P Comon. Independent component analysis: A new concept? *Signal Processing*, 36:287–314, 1994.

[6] M. Davy and A. Klapuri, editors. *Signal Processing Methods For Music Transcription*. Springer, 2006.

[7] Ch. Genest and J. V. Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1:114–148, 1986.

[8] R. Gribonval, L. Benaroya, E. Vincent, and C. Févotte. Proposals for performance measurement in source separation. In *Proc. 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 763–768, Nara, Japan, 2003.

[9] K. Kashino and H. Tanaka. A sound source separation system with the ability of automatic tone modeling. In *International Computer Music Conference (ICMC)*, August 1993.

[10] SG Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.