Identification of Components in Music Signal by Sequential Monte Carlo

Štěpán Albrecht

Abstract— **In this article we introduce a novel approach in music transcription (i.e., in detection of pitch, loudness and timing of all sound events in the complex music signal) working without any constraints on the observed music signal in general. It follows the reverse working of music sequencers. That is, we have a bank comprising arbitrary sounds as drums, melodic sounds, etc. Given the bank of sounds and a piece of a complex music signal for analysis, the sounds in the bank (or their modifications) are identified in the music signal. The sound events are the output of the identification process. When we try to put what was identified into the track, we should obtain the same or rather similar song up to some point. The core part of the algorithm is the the sequential Monte Carlo method (SMC). In this article, the necessary theory of the SMC is introduced, the state-of-the-art in music transcription by the SMC is presented. The algorithm based on the novel approach is described and demonstration of its functionality is depicted.**

I. INTRODUCTION

The problem of music signal processing has been referred since 1970s. The state-of-the-art systems can be divided into two parts: the transcribing and separating tasks. The primary target material for both of them is usually a complex music signal where several sounds are played simultaneously. A typical example is a Western music piece containing, e.g., drums, bass guitar, keyboard or guitar tracks, and singing. *The transcribing tasks.* A complete transcription would require the pitch[1], loudness, timing and instrument of all the sound events to be resolved. These parameters capture the meaningful music information to perform or synthesize a piece of music. In Western tradition, the written music uses the note symbols to indicate these parameters. In a computational transcription system, a MIDI file is an appropriate format for musical notation. Detecting and recognizing individual sounds in music is a big part of its perception, although musical notation is primarily designed to serve for sound production and not to model hearing. It should be emphasized that we do not hear music in terms of note symbols but, as described Bregman, music often “fools” the auditory system so that we perceive simultaneous sounds as a single entity, see the book [4], page 5. And there follows a problem – when two characteristic sounds[2] played together produce another characteristic sound, which one(s) is (are) characteristic? Instruments (or sounds) are referred to as the non-percussive (pitched) or percussive (drums, …). In the transcription of pitched instruments, typical restrictions are that the number of concurrent sounds is limited, interference of drums and percussive sounds is not allowed, or only a specific instrument is considered. In percussion transcription, quite good accuracy has been achieved in the transcription of percussive tracks which comprise a limited number of instruments and no pitched instruments. Research on musical instrument classification has mostly concentrated on working with isolated sounds, although more recently this has been attempted in polyphonic audio signals. *The separating tasks.* In the input there is either one or more complex music signals (microphones). By this task we mean to separate the sound sources according to their statistical independence. The more microphones we have and the more statistically independent the simultaneous sounds are, the separation is more successful. The output is an audio wave.

In this report we introduce a novel approach in music signal processing. We follow the working of the music sequencers. They work so that we have a bank of sounds, e.g., the audio-WAV-samples[3], as the key-stones to compose a music. These sounds are put into tracks. The resulting music signal is a superposition of these sounds. In this approach we would like to simulate the inverse process. Thus, given the bank of sounds and a piece of a complex music signal, the sounds in the bank (or their modifications) are identified in the music signal. The sound events are the output of the identification process. When we try to put what was identified into the track, we should obtain the same or rather similar song up to some point.

Let us call the bank of sounds as the wave-table and its sounds as the (sound) components, the input music piece as the observed music signal. The superposition of sounds we will term as the composition, its reverse process by the identification or decomposition. See Fig. 1.

This novel approach can be classified as the transcribing tasks. It can be seen as a generalization of a transcription – by the components we can characterize what should be sought, thus this proposal is not dedicated for any special kind of music signals to work correct. Certainly, often cited polyphonic periodic sound (a piano) transcription or drum transcription algorithms can be resolved by this proposal, too.

Recently, the same novel approach was discussed in [9],

Š. Albrecht is with Faculty of Applied Sciences, Department of Computer Science and Engineering, University of West Bohemia in Pilsen, Czech Republic albrs@kiv.zcu.cz

[1] Pitch is defined as the frequency of the sine wave that is matched to the target sound by human listeners. Fundamental frequency is the corresponding physical term and is defined for periodic or nearly-periodic sounds only.

[2] meaning e.g. a sound of a particular instrument

[3] The sound samples can be arbitrary in general (arbitrary in the length, loudness and sound color) – from the tone A_1 of a piano, to a drum pattern of a popular song.

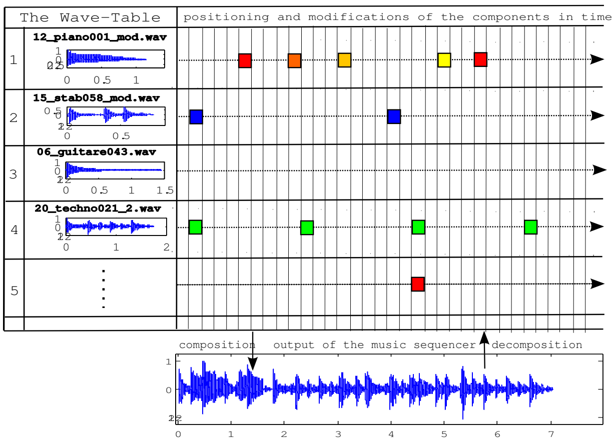

Fig. 1. Illustration of a music sequencer operation. The various colors of the component events in the tracks belong to the different component modifications. E.g., the changes between the red and yellow color could denote the pitch-shifting. Another colors could denote the various truncation of the components, or changes of the component loudness in comparison to the original in the wave-table.

[10]. In [9], the problem was treated by optimization methods as a part of the unsupervised sound separation algorithm of Virtanen [11]. However, the author did not deal with the component truncation there. This causes ending of the optimization in a low local minima, since the huge number of free parameters, thus the results may not be useful. In [10], the problem solution by the sequential Monte Carlo (SMC) as in this article is considered, however no results are presented there.

In the section II, the theory of the SMC is introduced, in III, the state-of-the-art in music signal processing by the SMC is presented. In IV, we define this approach thoroughly and present the algorithm based on the proposal. Demonstration of its functionality is depicted there and its parameters are discussed. In the last sections, the conclusions and the future suggestions are discussed.

II. Sequential Monte Carlo Methods (MC)

A. Monte Carlo Methods (MC)

Monte Carlo methods utilize statistical sampling and estimation techniques to evaluate the solutions to mathematical problems. Having enough samples reflecting some phenomenon, the distribution of the phenomenon can be approximated. The main concept is that the goal distribution is a posterior and we operate in the Bayesian framework.

There is a random variable $\mathbf{x}$ being a scalar or a vector in some space $\mathcal{X}$ which can be continuous or discrete. The probability, that a random variable $\mathbf{x}$ will appear, is denoted $\mathrm{p}(\mathbf{x})$ where $\mathrm{p}(.)$ is a distribution function[4].

For example, the minimum-mean square error (MMSE) estimates are determined by

$$\hat{\mathbf{x}} = \int_{\mathcal{X}} \mathbf{x}.\mathrm{p}(\mathbf{x}|\mathbf{y})d\mathbf{x} \qquad (1)$$

Whenever this estimation is not possible, a numerical integration technique should be implemented. Consider the more general integral computation case

$$I[h] = \int_{\mathcal{X}} h(\mathbf{x})\mathrm{p}(\mathbf{x}|\mathbf{y})d\mathbf{x} \qquad (2)$$

where $I[h]$ represents the expected value of the function h.

When the dimension of $\mathcal{X}$ is small (smaller than three), it can be calculated numerically via Riemann (provided $\mathcal{X}$ is a compact set). However, the dimension we encounter in music signal modeling takes the size of tens or hundreds. Since the grid size increases exponentially with the dimension, the numerical approach becomes infeasible.

Fortunately, another numerical computation technique can be applied. Assume random samples $\mathbf{x}^{(i)}, i = 1, \ldots, N$ are available, where each sample is distributed according to $\mathrm{p}(\mathbf{x}|\mathbf{y})$ ($\mathbf{x}^{(i)} \sim \mathrm{p}(\mathbf{x}|\mathbf{y})$). Then $I[h]$ can be approximated by the empirical average

$$I[h] \approx \widehat{I}_N[h] = \frac{1}{N}\sum_{i=1}^{N} h(\mathbf{x}_i) \qquad (3)$$

, which is called the *Monte Carlo estimate of* $I[h]$. Then the random samples $\mathbf{x}^{(i)}, i = 1, \ldots, N$ are referred to as the *Monte Carlo samples*. The estimate $\widehat{I}_N[h]$ is unbiased for any N and consistent. One crucial property of Monte Carlo approximation is that the estimation accuracy does not depend as much on the dimensionality of $\mathcal{X}$ but on the ability to focus on the significant locations in the posterior distribution.

The hardest thing in MC techniques is to obtain an approximating distribution from which is possible to generate the samples. Several techniques have been developed for random variable generation from any distribution. One of them is the *importance sampling*, the other is, e.g., the *Monte Carlo Markov Chain (MCMC)* method [5], [6].

B. Importance Sampling (IS)

Suppose a posterior distribution $\mathrm{p}(\mathbf{x}|\mathbf{y})$ from which it is difficult to draw the samples. Next, assume the random samples to be easily generated from another distribution $\mathbf{x}^{(i)} \sim \mathrm{q}(\mathbf{x}|\mathbf{y}), i = 1, \ldots, N$, and $\mathrm{q}(\mathbf{x}|\mathbf{y}) \neq 0$ whenever $\mathrm{p}(\mathbf{x}|\mathbf{y}) \neq 0$. In terms of the so called *importance* distribution $\mathrm{q}(.)$, the expectation of h can be rewritten as follows:

$$I[h] = \int_{\mathcal{X}} h(\mathbf{x})\mathrm{p}(\mathbf{x}|\mathbf{y})d\mathbf{x} = \int_{\mathcal{X}} h(\mathbf{x})\frac{\mathrm{p}(\mathbf{x}|\mathbf{y})}{\mathrm{q}(\mathbf{x}|\mathbf{y})}\mathrm{q}(\mathbf{x}|\mathbf{y})d\mathbf{x} \qquad (4)$$

Then for the following Monte Carlo estimate it holds [3]:

$$\widehat{I}_N[h] = \sum_{i=1}^{N} \omega^{(i)} h(\mathbf{x}^{(i)}) \approx I[h], \text{with} \quad \omega^{(i)} = \frac{\mathrm{p}(\mathbf{x}^{(i)}|\mathbf{y})}{\mathrm{q}(\mathbf{x}^{(i)}|\mathbf{y})} \qquad (5)$$

The *imporatance weight* $\omega^{(i)}$ can be understand as a discrepancy between $\mathrm{q}(.)$ and $\mathrm{p}(.)$, when the samples are generated from $\mathrm{q}(.)$ instead of $\mathrm{p}(.)$. If we had enough samples from whole support of $\mathrm{p}(.)$ but generated from $\mathrm{q}(.)$, the correct estimation of $\hat{I}_N[h]$ would not be subject to the suitable importance $\mathrm{q}(.)$ selection. Thus, the importance pdf should

[4] For simplicity, the probability density function $\mathrm{p}(.)$ of a continuous variable is termed identically as a distribution $P(.)$ of a discrete variable.

be as close as possible to the posterior distribution because then we have enough samples on the locations of high posterior probability. Having the samples from q(.) the posterior probability can be approximated by

$$\mathrm{p}(\mathbf{x}|\mathbf{y}) \approx \sum_{i=1}^{N} \omega^{(i)} \delta(\mathbf{x} - \mathbf{x}^{(i)}) \qquad (6)$$

where δ the Kronecker's delta symbol.

C. Sequential Importance Sampling (SIS)

The SIS is a group of methods applying the IS into the dynamic framework of the sequential Monte Carlo (SMC).

The sequential form of the weight $\omega_t^{(i)}$ of the sample sequence up to time t, $\mathbf{x}_{0:t}^{(i)}$, is given as follows:

$$\omega_t^{(i)} = \frac{\mathrm{p}(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}{\mathrm{q}(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})} \propto \omega_{t-1}^{(i)} \frac{\mathrm{p}(\mathbf{y}_t|\mathbf{x}_t^{(i)})\mathrm{p}(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\mathrm{q}(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_t)} = \acute{\omega}_t^{(i)} \qquad (7)$$

The step towards the recursive formula $\acute{\omega}_t^{(i)}$ in (7) can be accomplished because the denominator $\mathrm{p}(\mathbf{y}_t|\mathbf{y}_{1:t})$ is constant for all $\mathbf{x}_t^{(i)}, i = 1, \ldots, N$ and because it is considered $\mathrm{q}(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t}) = \mathrm{q}(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_t)$ following from the Markov process, see [3], [2]. The weight $\omega_t^{(i)}$ is a normalized form of $\acute{\omega}_t^{(i)}$, that is $\omega_t^{(i)} = \acute{\omega}_t^{(i)} / \frac{1}{N} \sum_{j=1}^{N} \acute{\omega}^{(j)}$.

In the common case only a filtered estimate of $\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$ is required at each time step. In such scenarios, only $\mathbf{x}_t^{(i)}$ must be stored, therefore one can discard the path $\mathbf{x}_{0:t-1}^{(i)}$. The modified weight is then

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{\mathrm{p}(\mathbf{y}_t|\mathbf{x}_t^{(i)})\mathrm{p}(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\mathrm{q}(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)} \qquad (8)$$

and the posterior filtered density $\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$ can be approximated as

$$\mathrm{p}(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \hat{\mathrm{p}}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i=1}^{N} \omega_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \qquad (9)$$

D. Degeneracy Problem and Generic SIS Algorithm

A common problem with the SIS particle filter is the degeneracy where after a few iterations, all except one sample have a negligible weight. It has been shown in [1] that the variance of the importance weights can only increase over time, and thus, it is impossible to avoid the degeneracy phenomenon. A suitable measure of degeneracy of the algorithm is the effective sample size N_{eff} approximated by

$$\widehat{N_{\text{eff}}} = \frac{1}{\sum_{i=1}^{N} (\omega_t^{(i)})^2}. \qquad (10)$$

Notice, that $N_{\text{eff}} \leq N$, and small N_{eff} indicates a severe degeneracy. The brute force approach to reducing its effect is to use an increasing number of samples N as t increases or a very large N. This is impractical, therefore we rely on two other methods: First on a good sampling algorithm selection [3], [2], or on a use of resampling. The basic idea of resampling is to eliminate samples that have small weights and to concentrate on samples with large weights. The resampling step involves generating a new set $\{\mathbf{x}_t^{*(i)}\}_{i=1}^{N}$ by resampling N times from an approximate discrete representation (9) so that $\mathrm{p}(\mathbf{x}_t^{*(i)} = \mathbf{x}_t^{(j)}) = \omega_t^{(j)}$. The resulting sample is in fact an i.i.d. sample from the discrete density (9), thus, the normalized weights are now reset to $\omega_t^{(i)} = 1/N$.

Algorithm *Generic Sequential Importance Sampling Algorithm*:

```
• For $i = 1, \ldots, N$
    – Draw $\mathbf{x}_t^{(i)} \sim \mathrm{q}(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)$.
    – Assign the normalized sample weight according to (8).
• Calculate $\widehat{N_{\text{eff}}}$ using (10).
• If $\widehat{N_{\text{eff}}} < N_{\text{threshold}}$
    – Resample as written in II-D.
• Compute estimates[5]: $\hat{\mathbf{x}}_t = \sum_{i=1}^{N} \omega_t^{(i)} \mathbf{x}_t^{(i)}$.
```

III. State-of-the-art in SMC for Music Signal Processing

Music data have some common characteristics, but they may vary a lot. Even two single real sound sources, produced by identical musical instruments, are not the same even under the most equal conditions. The sources cannot be modeled deterministically. Statistical methods represent a powerful tool for modeling of sound sources.

The algorithms of the SMC are aimed at pitch detection of simultaneous tones in one segment (*frame*) of music, usually length of 92ms or 46ms. They work without any prior knowledge on the different tones and timbres of music instruments. For the sake of this they are able to operate reliably only with periodic sounds.

There are two main approaches also of MC for music signal processing – the on-line (SMC) and off-line – Monte Carlo Markov Chain [6]. The off-line approaches have a major advantage of being quite accurate. The drawback is that the computational requirements may be high. On-line approaches, on the other hand, only use the current frame at time t and information from the past estimates. It should be noted that they are quite attractive for multi-pitch detection, because they do not require a separate onset / offset detection as in the off-line [6], and they provide some flexibility to formulate completely on-line inference schemas, such as those presented in this section. Unfortunately, there is not a proper evaluation for online MC multi-pitch tracking, it is only pointed out in [4] that it is comparable to the off-line ones. For the off-line approaches in [4], Davy reports for mono-phony 100% accuracy, for polyphony 2 about 85%, for polyphony 3 about 74%, for polyphony 4 about 71% accuracy. There are also other approaches [4], however, all of them have a common property – they are aimed at periodic sound recordings and discovering the fundamental frequencies of tones in a polyphony.

A. Application Example of the SMC in MSP

Approach of Davy and Dubois [7], [8]. We have a discrete time signal denoted by $\{y_t\}$ with sampling frequency f_s.

In [8] the signal model estimate is given as follows:

$$z_t(\tau) = \sum_{j=1}^{N_t} \sum_{h=1}^{H} \left[\alpha_{h,j}(t) w(\tau) \cos\left(2\pi \frac{f_{t,h,j}}{f_s} \tau' \right) \right]. \quad (11)$$

The letter w is an analysis window (or a frame) of length $2L_w + 1$ time points with typically Gaussian, Hamming, etc. shape. The window is centered around time t, where $\mathbf{y}_t = [z_{t-L_w/f_s}, \dots, z_{t+L_w/f_s}]$. The positioning in the observation signal respecting the offset at time t of the current frame is denoted $\tau' = t + (\tau - 1 - L_w)/f_s$. H is the number of harmonics (being fixed) and N_t is the number of simultaneous tones / notes at time t.

The discrete Fourier transform (DFT) is applied on the frames of the windowed observed and estimated signal, yielding $\mathbf{y}_t^{\mathrm{DFT}} = ||\mathrm{DFT}(\mathbf{w}_t * \mathbf{y}_t)||^2$, $\mathbf{z}_t^{\mathrm{DFT}} = ||\mathrm{DFT}(\mathbf{z}_t)||^2$ respectively, thus we are getting the observation equation

$$\mathbf{y}_t^{\mathrm{DFT}} = \mathbf{z}_t^{\mathrm{DFT}}(\mathbf{f}_t, \boldsymbol{\alpha}_t, N_t) + \mathbf{v}_t. \quad (12)$$

"$*$" denotes element-wise multiplication. It determines the Gaussian likelihood $\mathrm{p}(\mathbf{y}_t^{\mathrm{DFT}} | \mathbf{f}_t, \boldsymbol{\alpha}_t, N_t) = \mathcal{N}(\mathbf{y}_t^{\mathrm{DFT}}; \mathbf{z}_t^{\mathrm{DFT}}, diag(\mathbf{r}))$ where $diag(\mathbf{r})$ is a diagonal covariance matrix with elements of value 0.05. The prior for $\mathbf{f}_t$, $\boldsymbol{\alpha}_t$ is given by a random walk

$$\begin{aligned} f_{t,h,j} &= f_{t-1,h,j} + v_{t-1,h,j}^{\mathbf{f}} \quad (13) \\ \alpha_{t,h,j} &= \alpha_{t-1,h,j} + v_{t-1,h,j}^{\boldsymbol{\alpha}} \quad (14) \end{aligned}$$

where $v_{t-1,h,j}^{(.)}$ is a zero-mean white noise with Gaussian density of variance $r_{t-1,h,j}^{(.)}$. The variance for amplitudes and frequencies is allowed to evolve according to

$$\log(r_{t,h,j}^{(.)}) = \log(r_{t-1,h,j}^{(.)}) + \varphi_{t-1,h,j}^{(.)}. \quad (15)$$

The prior for number of simultaneous tones N_t was given by a transitional table, such as in table I, on the right.

Sampling of variables to estimate was performed from the prior / transitional for N_t and by Kalman filtering [2].

Number of particles was $M = 500$, $H = 6$, N_t from 1 to 3, frame length (Hanning) 256 and $\sigma_\varphi = 0.01$. There is no information about time the processing takes.

The observed material (artificially created recording) was tested by a root means square (RMS) error $\mathrm{Err}_{\mathrm{RMS}} = \sqrt{\frac{\sum_{k=1}^{K} \omega_t^{(i)} ||\mathbf{y}_t^{\mathrm{DFT}} - \mathbf{z}_t^{\mathrm{DFT}}||^2}{L_w}}$.

IV. Proposal Solution by SMC Methods

A. Introduction

There is an observed music signal represented by a matrix $\mathbf{Y}$. Its columns, $\mathbf{y}_t^{\mathrm{DFT}}$, are the magnitude DFT values of the windowed signal in one frame $\mathbf{y}_t$.

All frames of all components are also windowed and transformed by the DFT. The absolute values (i.e., the magnitude spectrum) yield $\mathbf{x}_c^{\mathrm{DFT}}$. They build column by column a matrix $\mathbf{X} = \left[\mathbf{x}_1^{\mathrm{DFT}} | \mathbf{x}_2^{\mathrm{DFT}} | .. | \mathbf{x}_C^{\mathrm{DFT}} \right]$, where C is the number of frames of all components. E.g., the first component captures 1–6th column, the second 7–18th column, etc. Within one component all frames keep the time order with the increasing column index.

In order to prevent the singular cases, we consider all frames of all components to be non-silent, that is $||\mathbf{x}_c^{\mathrm{DFT}}|| > threshold$.

We consider a linear signal model. For every observation $\mathbf{y}_t$, matrix $\mathbf{X}$ we have

$$\mathbf{y}_t^{\mathrm{DFT}} \approx \mathbf{z}_t^{\mathrm{DFT}} = \mathbf{X}\mathbf{s}_t \quad (16)$$

where $\mathbf{s}_t$ is the vector of presence or non-presence. Its elements are either zero, or one according to if the c-th frame is present in $\mathbf{y}_t^{\mathrm{DFT}}$ or not. Its every c^{th} element corresponds to $\mathbf{x}_c^{\mathrm{DFT}}, c = 1, \dots, C$. We also define a vector of indices of frames which are present at time t, $\mathbf{n}_t$. This is an equivalent of $\mathbf{s}_t$. Its length is represented by N_t. In the following text we often deal with n_t^j or n_t^i, which is referred to as a value on the i-th or j-th position of the vector of indices $\mathbf{n}_t$. When the vector size is only of one, there is no point to express the index position i or j in vector of indices, hence, this is denoted by n_t. The number of active frames at t is then $N_t = \sum_{c=1}^{C} s_t^c$, and their maximal number is given beforehand ($N_{\max}$).

The reason why we apply stochastic methods is an *over-completeness*[6]. Since we deal with a huge amount of data (parameters of all frames) and we want to find just a few of them, common optimization algorithms fail. However, stochastic modeling allows to incorporate any heuristics we know about the problem[7], in such a way that the right subspace of the parameters is focused. Our task is to estimate parameters according to their posterior distribution.

B. Estimated Parameters

Desired parameters at time t are the number of polyphony N_t and the vector of indices, that is, vector of frame presence $\mathbf{n}_t \equiv \mathbf{s}_t$. We note that a side product of the estimation are the bounds of the component sounds (they are not sampled) – since we work with individual frames of the sound components, it is allowed to identify an arbitrary length of the sound components. The detection of the arbitrary length can be understood as the component modification (section I).

C. Likelihood and Transitional Distribution

Likelihood is given as $\mathbf{L}_t(\mathbf{y}_t^{\mathrm{DFT}} | \mathbf{s}_t, \sigma_{1,t}^2) = \mathcal{N}(\mathbf{y}_t^{\mathrm{DFT}}; \mathbf{X}.\mathbf{s}_t, \boldsymbol{\Sigma}_{\sigma_1})$, $\boldsymbol{\Sigma}_{\sigma_1} = \frac{1}{\sigma_{1,t}^2}.\mathbf{I}$.

For the vector of presence $\mathbf{n}_t \equiv \mathbf{s}_t$ and the number of present components N_t the transitionals (or the priors) are:

1) $\mathrm{p}(\mathbf{n}_t | \mathbf{n}_{t-1}, N_t)$*:* Transition probabilities are determined by a matrix $\mathbf{H}$. The values are denoting the succession between all frames. See example on table I.

Discrete distribution for sampling of one frame n_t presence conditioned the frame combination $\mathbf{n}_{t-1}$ is as follows:

$$\mathrm{p}(n_t | \mathbf{n}_{t-1}, N_t) \propto k_1 \sum_{j=1}^{N_{t-1}} \mathbf{h}_{(n_{t-1}^j)} + k_2 \sum_{c=1}^{C} \mathbf{h}_{(c)} \quad (17)$$

[6] Meaning "too many parameters to assess".

[7] E.g., heuristic of the number of simultaneous components, or, the silent parts in the components which are good candidates for the start/end points of the component, or a likelihood of the estimated component combination.

From To	1	2	3	4
1	0	0	0	1
2	1	0	0	0
3	0	1	0	0
4	0	0	1	0

N_{t-1} k_t	0	1	2
+2	1/10	0	0
+1	2/10	1/9	0
0	7/10	7/9	7/10
-1	0	1/9	2/10
-2	0	0	1/10

TABLE I

LEFT TABLE REPRESENTS THE EXAMPLE OF TRANSITION PROBABILITIES OF 4 FRAMES. RIGHT TABLE IS BOTH PRIOR (TRANSITIONAL) AND IMPORTANCE DISTRIBUTION FOR NUMBER OF SIMULTANEOUS ACTIVE FRAMES N_t, WHICH IS GIVEN BY $k_t \sim \mathrm{p}(k_t|N_{t-1})$ AND BY $N_t = N_{t-1} + k_t$. WE CONSIDER HERE $N_{\max} = 2$.

The left term of "+" models the highlighted probability of the expected following frames while the right term represents a vector of all frames presence. The former is multiplied by $k_1 = \sum_i \sum_{j=1}^{N_{t-1}} h_{(n_{t-1}^j)}$. The latter is the case of "sampling from silence", i.e., sampling from information excluding the previous present frames, it is multiplied by $k_2 = \sum_{i,j} h_{i,j}$. By k_1, k_2 multiplication it is implied that the prior (transitional) distribution of n_t is equally probable for the left or right term. The total probability of the predicted frame combination is calculated by $\mathrm{p}(\mathbf{n}_t|\mathbf{n}_{t-1}, N_t) \propto \frac{1}{N_t}\sum_{i=1}^{N_t} \mathrm{p}(n_t|\mathbf{n}_{t-1}, N_t)$.

2) $\mathrm{p}(N_t|N_{t-1}) \equiv \mathrm{p}(k_t|N_{t-1})$: Given by a table I. The aim is to allow N_t to increase or decrease but the probability to keep it constant must be prominent.

D. Importance Distributions

The importance distributions should be as close as possible to the posterior distribution and they have to be sampleable. There is not a closer distribution for N_t than the prior (transitional) one, thus this is sampled as the importance distribution. The transitional distribution will not appear in the weight formula (8) since this is reduced by a fraction. For the presence vector $\mathbf{n}_t$ calculation a heuristical approach is applied.

1) Calculating and Sampling from $\mathrm{p}(\mathbf{n}_t|\mathbf{y}_t^{\mathrm{DFT}}, N_t, \mathbf{n}_{t-1})$: The *similarity measure* of j-th frame n_t^j to the observation $\mathbf{y}_t^{\mathrm{DFT}}$ corresponds to a temporary likelihood $\mathrm{p}(\mathbf{y}_t^{\mathrm{DFT}}|n_t^j)$. The measure should be fast and effective to calculate. If $||\mathbf{y}_t^{\mathrm{DFT}}|| < \mathit{threshold}$, i.e., if there is a silence in the observed signal, then $\forall c \in C : \mathrm{p}(\mathbf{y}_t^{\mathrm{DFT}}|n_t^c) = 0$. We tried these simple similarity measures:

$$\begin{aligned}\mathrm{p}(\mathbf{y}_t^{\mathrm{DFT}}|n_t^j) &= \cos\varphi = \frac{\mathbf{y}_t^{\mathrm{DFT}}.\mathbf{x}_{n_t^j}^{\mathrm{DFT}}}{||\mathbf{y}_t^{\mathrm{DFT}}||.||\mathbf{x}_{n_t^j}^{\mathrm{DFT}}||} \\ &\propto 1 - \min_\beta ||\mathbf{y}_t^{\mathrm{DFT}} - \beta.\mathbf{x}_c^{\mathrm{DFT}}|| \\ &= \mathcal{N}(\mathbf{y}_t^{\mathrm{DFT}}; \hat{\beta}.\mathbf{x}_c^{\mathrm{DFT}}, \mathbf{\Sigma}_x) \qquad (18)\end{aligned}$$

where $\hat{\beta} = \min_\beta ||\mathbf{y}_t^{\mathrm{DFT}} - \beta.\mathbf{x}_c^{\mathrm{DFT}}||$. The best of them appeared to be the first one, since this was reflecting the presence of a frame.

```
Algorithm Algorithm of SMC:
• Initialization:
  – For $i = 1, \ldots, M$ sample $\tilde{N_0}^{(i)} \sim \mathrm{p}(N_t|N_{t-1} = 0)$
  – For $i = 1, \ldots, M$
      ∗ For $j = 1, \ldots, N_0$ sample $n_0^j \sim \mathrm{p}(n_0) \propto \sum_{c=1}^{C} \mathbf{h}_c$
      ∗ Sample $\tilde{\mathbf{n}}_0^{(i)} \sim \mathrm{p}(\mathbf{n}_0) = \frac{1}{N_0}\sum_{j=1}^{N_0} \mathrm{p}(n_0^j)$
• Iterations:
  – For $t = 1, 2, \ldots, T$
      ∗ For $i = 1, \ldots, M$
          · sample $\tilde{N_t}^{(i)} \sim \mathrm{p}(N_t|N_{t-1}^{(i)})$
          · sample $\tilde{\mathbf{s}_t}^{(i)} \equiv \tilde{\mathbf{n}_t}^{(i)} \sim \mathrm{p}(\mathbf{n}_t|\mathbf{y}_t^{\mathrm{DFT}}, \tilde{N_t}^{(i)}, \mathbf{n}_{t-1}^{(i)})$
          · compute the weight:
              $\tilde{\omega}_t^i = \omega_{t-1}^i \frac{\mathbf{L}_t(\mathbf{y}_t^{\mathrm{DFT}}|\tilde{\mathbf{s}}_t, \sigma_{1,t}^2)\mathrm{p}(\tilde{\mathbf{n}_t}^{(i)}|\mathbf{n}_{t-1}^{(i)}, \tilde{N_t}^{(i)})}{\mathrm{p}(\tilde{\mathbf{n}_t}^{(i)}|\mathbf{y}_t^{\mathrm{DFT}}, \tilde{N_t}^{(i)}, \mathbf{n}_{t-1}^{(i)})}$   (21)
      ∗ Normalize the weights $\tilde{\omega}_t^{(i)}$ so that $\sum_{i=1}^{N} \tilde{\omega}_t^{(i)} = 1$
      ∗ Compute estimates of $\mathbf{n}_t$:
              $N_t : \quad \widehat{N_t} \approx \sum_{i=1}^{M} \tilde{\omega}_t^{(i)} \tilde{N_t}^{(i)}$
              $\mathbf{n}_t : \quad \widehat{\mathbf{n}_t} \approx \mathrm{argmax}_{\mathbf{n}_t} \sum_{i=1}^{M} \tilde{\omega}_t^{(i)} \mathbb{I}(\tilde{\mathbf{n}_t}^{(i)} = \mathbf{n}_t, \tilde{N_t}^{(i)} = \widehat{N_t})$   (22)
      ∗ If $\left[\sum_{i=1}^{M} \tilde{\omega}_t^{2(i)}\right]^{-1} \leq \eta M$, then resample, i.e. duplicate
        the particles with large weight and remove the particles with
        small weight. The new particles lose the tilde sign and have
        weight $\omega_t^{(i)} = 1/M$.
      ∗ Else, assign the particles $\mathbf{n}_t^{(i)} \leftarrow \tilde{\mathbf{n}}_t^{(i)}$, $N_t^{(i)} \leftarrow \tilde{N_t}^{(i)}$.
```

Drawing from $\mathrm{p}(\mathbf{n}_t|\mathbf{y}_t^{\mathrm{DFT}}, N_t, \mathbf{n}_{t-1})$. For $j = 1 \ldots N_t$ we sample[8] from the discrete distribution

$$\mathrm{p}(n_t^j|\mathbf{y}_t^{\mathrm{DFT}}, N_t, \mathbf{n}_{t-1}) \propto \mathrm{p}(\mathbf{y}_t^{\mathrm{DFT}}|n_t^j) * \mathrm{p}(n_t^j|\mathbf{n}_{t-1}, N_t) \qquad (19)$$

where "*" denotes element-wise multiplication of the two discrete distributions represented by vectors. Hence, we result in a total presence of the frame-combination importance probability

$$\mathrm{p}(\mathbf{n}_t|\mathbf{y}_t^{\mathrm{DFT}}, N_t, \mathbf{n}_{t-1}) \propto \frac{1}{N_t}\sum_{j=1}^{N_t} \mathrm{p}(n_t^j|\mathbf{y}_t^{\mathrm{DFT}}, N_t, \mathbf{n}_{t-1}). \qquad (20)$$

Note that in (20) the probability of sampled $\mathbf{n}_t$ combination is calculated up to a normalizing constant which can be omitted since it is constant for all frame combinations.

It must be mentioned here, that in the case when we drew two identical n_t^j, the sampling is repeated.

In a singular case the zero distribution should be sampled – this can happen when the sound loudness in the observation signal did not get over a given threshold (see $\mathrm{p}(\mathbf{y}_t^{\mathrm{DFT}}|n_t^j)$ heuristics above). Then an arbitrary N_t is replaced by $N_t = 0$ and some negligible probability is assigned to this[9].

E. Testing

The testing material was similar to [9] – we had downloaded 21 sound components of arbitrary length, together about 30 seconds of complex music signal, sampled at 44.1 kHz. They represented the wave-table and counted miscellaneous sounds from drum patterns to individual tones of bass

[8] N_t is given by sampling from $\mathrm{p}(N_t|N_{t-1})$.

[9] resulting in a large particle weight.

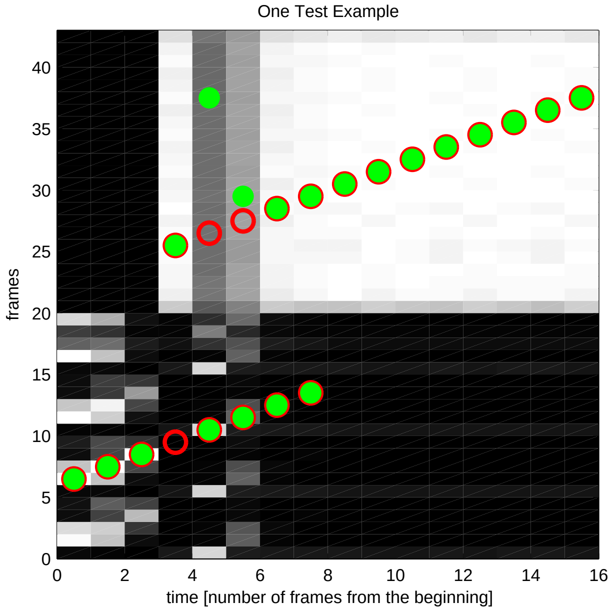

Fig. 2. The figure should represent the hits regarding the similarity measure. The black and white fields denotes the similarity measure (i.e., the part of the heuristics to express the importance distribution). The fields correspond to all frames of these two components – the first – a drum pattern ("disco-funk"), the second – an organ sound. On the y-axis, every element corresponds to one frame of one of the two components. Therefore the consecutive frames of one component produce the diagonals. The first is starting by frame 7, finishing by frame 14, the second similarly, they overlap in 5 frames. The red circles represent positions of frames which should be identified, the filled circles denote the positions of what was really identified. We mention, that the frame-wise identification runs on frames of all components, for clarity we are focused only on the two components in this figure.

or various synthesizers, however, due to the singularities, all components did not contained silent parts longer than 93ms, which was the length of one frame. The aforementioned matrix $\mathbf{X}$ counted 354 not overlapping frames. One frame of the DFT contained 2048 spectral bins[10]. The algorithm is designed to be able to operate with a silent in $\mathbf{y}_t^{\mathrm{DFT}}$.

The testing sound was created from two components of the wave-table. Both were truncated and the bounds were recorded. The sound components were overlapped so that the longer component started at the half of the shorter – they were summed resulting in a testing sound, thus, we had exact information about the bounds (truncations), the number of the components and their times. An example can be seen on Fig. 2.

If there is $N_{t-1} = 1$ at the time $t-1$ and the change plus-minus one is happened, there are 20% of the correct number at time t. If we intended to cover all frame combination ($N_{\max} = 2$) we would need more than 3.10^5. In this SMC algorithm we tried to test only 2000 samples (particles).

We have found out that better was to choose greater $\sigma_{t,1}^2$ since the frame combination of two having correct only one inner frame was assigned to a greater weight. $M = 2000$ samples was enough to cover 354 individual frames. We set $\sigma_{t,1}^2 = 150$ when the amplitude of a sound wave was maximally one. Remaining settings are presented in the previous subsection.

V. CONCLUSIONS AND FUTURE WORKS

A. *Conclusions*

The aim of this work was to introduce a novel system (concept) for detection of sounds presented in a complex music signal. The sequential Monte Carlo was proposed to be the core part. The theoretical background of the SMC was introduced. The state-of-the-art the SMC for music signal processing were outlined. In the testing part, we explained which parameter values we choose and provided a figure of a representative test example. We explained where this presents its challenge and its utilization.

B. *Future Works*

1) Algorithm: The results could be improved by application of the *smoothing* [3], [2]. Next, we would try to work also with frame differences as another heuristics "inside" the importance distribution. A smart similarity measure selection may cause the improvement in the identification as well.

2) Testing: We propose a combinations of, e.g., the root means square error between the truth and the estimate, and error following from the hitting or non-hitting the exact frame. The testing material could contain the combinations of only non-percussion music pieces, only percussion music pieces, or their combinations. Different level of a distortion could be applied onto the synthesized recordings (e.g., obtained from songs based on MIDI) in order to obtain various quality testing material with the exact information about its content.

[10] A half of 4096.

References

[1] A. Kong, J. S. Liu, W. H. Wong, "Sequential Imputations and Bayesian Missing Data Problems", J. Amer. Statist. Assoc., vol. 89, pp. 278-288, 1994

[2] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A Tutorial on Partile Filters for Online Nonlinear / Non-Gaussian Bayesian Tracking", *IEEE Transactions on Signal Processing*, vol. 50., no. 2, 2002

[3] Z. Chen, "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond"

[4] M. Davy, A. Klapuri, "Signal Processing Methods For Music Transcription", Springer 2006

[5] C. Andrieu, N. D. Freitas, A. Doucet, M. I. Jordan, "An Introduction To MCMC from Machine Learning", Kluwer Academic Publishers 2003

[6] M. Davy, S. Godsill, J. Idier, "Bayesian Analysis of Polyphonic Western Tonal Music", Ac. Soc. of Am., 2006

[7] M. Davy, C. Dubois, "A Fast Particle Filtering Approach to Bayesian Tonal Music Transcription", 2007

[8] M. Davy, C. Dubois, "Harmonic Tracking Using Sequential Monte Carlo", 2005

[9] S. Albrecht, "Music Signal Decomposition Approaches Based On Unsupervised Source Separation Methods", Ph.D. Workshop, Balatonf*ü*hred, Hungary, 2007

[10] S. Albrecht, V. Matousek, "Music Signal Decomposition Based on Identification and Subtraction of Components", DAGA 2008, Dresden

[11] T. Virtanen, "Separation of Sound Sources by Convolutive Sparse Coding", SAPA-2004, Jeju, Korea, 2004