IDIAP RESEARCH REPORT

A Generative Model for Music Transcription

Ali Taylan Cemgil [a] Bert Kappen [b]
David Barber [c]

IDIAP–RR 05-89

[a] with University of Amsterdam, Informatica Instituut, Kruislaan 403, 1098 SJ Amsterdam, the Netherlands
[b] B. Kappen is with Radboud University Nijmegen, SNN, Geert Grooteplein 21, 6525 EZ Nijmegen, the Netherlands
[c] IDIAP Research Institute, Rue du Simplon 4, CH-1920 Martigny, Switzerland

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 05-89

A Generative Model for Music Transcription

Ali Taylan Cemgil Bert Kappen
David Barber

Abstract. In this paper we present a graphical model for polyphonic music transcription. Our model, formulated as a Dynamical Bayesian Network, embodies a transparent and computationally tractable approach to this acoustic analysis problem. An advantage of our approach is that it places emphasis on explicitly modelling the sound generation procedure. It provides a clear framework in which both high level (cognitive) prior information on music structure can be coupled with low level (acoustic physical) information in a principled manner to perform the analysis. The model is a special case of the, generally intractable, switching Kalman filter model. Where possible, we derive, exact polynomial time inference procedures, and otherwise efficient approximations. We argue that our generative model based approach is computationally feasible for many music applications and is readily extensible to more general auditory scene analysis scenarios.

A Generative Model for Music Transcription

Ali Taylan Cemgil, *Student Member, IEEE,* Bert Kappen, *Senior Member, IEEE,* and David Barber

***Abstract*— In this paper we present a graphical model for polyphonic music transcription. Our model, formulated as a Dynamical Bayesian Network, embodies a transparent and computationally tractable approach to this acoustic analysis problem. An advantage of our approach is that it places emphasis on explicitly modelling the sound generation procedure. It provides a clear framework in which both high level (cognitive) prior information on music structure can be coupled with low level (acoustic physical) information in a principled manner to perform the analysis. The model is a special case of the, generally intractable, switching Kalman filter model. Where possible, we derive, exact polynomial time inference procedures, and otherwise efficient approximations. We argue that our generative model based approach is computationally feasible for many music applications and is readily extensible to more general auditory scene analysis scenarios.**

***Index Terms*— music transcription, polyphonic pitch tracking, Bayesian signal processing, switching Kalman filters**

I. Introduction

When humans listen to sound, they are able to associate acoustical signals generated by different mechanisms with individual symbolic events [1]. The study and computational modelling of this human ability forms the focus of computational auditory scene analysis (CASA) and machine listening [2]. Research in this area seeks solutions to a broad range of problems such as the cocktail party problem, (for example automatically separating voices of two or more simultaneously speaking persons, see e.g. [3], [4]), identification of environmental sound objects [5] and musical scene analysis [6]. Traditionally, the focus of most research activities has been in speech applications. Recently, analysis of musical scenes is drawing increasingly more attention, primarily because of the need for content based retrieval in very large digital audio databases [7] and increasing interest in interactive music performance systems [8].

A. Music Transcription

One of the hard problems in musical scene analysis is automatic music transcription, that is, the extraction of a human readable and interpretable description from a recording of a music performance. Ultimately, we wish to infer automatically a musical notation (such as the traditional western music notation) listing the pitch levels of notes and corresponding time-stamps for a given performance. Such a representation of the surface structure of music would be very useful in a broad spectrum of applications such as interactive music performance systems, music information retrieval (Music-IR) and content description of musical material in large audio databases, as well as in the analysis of performances. In its most unconstrained form, i.e., when operating on an arbitrary polyphonic acoustical input possibly containing an unknown number of different instruments, automatic music transcription remains a great challenge. Our aim in this paper is to consider a computational framework to move us closer to a practical solution of this problem.

Music transcription has attracted significant research effort in the past – see [6] and [9] for a detailed review of early and more recent work, respectively. In speech processing, the related task of tracking the pitch of a single speaker is a fundamental problem and methods proposed in the literature are well studied[10]. However, most current pitch detection algorithms are based largely on heuristics (e.g., picking high energy peaks of a spectrogram, correlogram, auditory filter bank, etc.) and their formulation usually lacks an explicit objective function or signal model. It is often difficult to theoretically justify the merits and shortcomings of such algorithms, and compare them objectively to alternatives or extend them to more complex scenarios.

Pitch tracking is inherently related to the detection and estimation of sinusoids. The estimation and tracking of single or multiple sinusoids is a fundamental problem in many branches of applied sciences, so it is less surprising that the topic has also been deeply investigated in statistics, (e.g. see [11]). However, ideas from statistics seem to be not widely applied in the context of musical sound analysis, with only a few exceptions [12], [13] who present frequentist techniques for very detailed analysis of musical sounds with particular focus on decomposition of periodic and transient components. [14] has presented real-time monophonic pitch tracking application based on a Laplace approximation to the posterior parameter distribution of an AR(2) model [15], [11, page 19]. Their method outperforms several standard pitch tracking algorithms for speech, suggesting potential practical benefits of an approximate Bayesian treatment. For monophonic speech, a Kalman filter based pitch tracker is proposed by [16] that tracks parameters of a harmonic plus noise model (HNM). They propose the use of Laplace approximation around the predicted mean instead of the extended Kalman filter (EKF). For both methods, however, it is not obvious how to extend them to polyphony.

Kashino [17] is, to our knowledge, the first author to apply graphical models explicitly to the problem of polyphonic music transcription. Sterian [18] described a system that viewed transcription as a model driven segmentation of a time-frequency image. Walmsley [19] treats transcription and estimation in a full Bayesian framework. He employs a

Manuscript received; revised .

A. T. Cemgil is with University of Amsterdam, Informatica Instituut, Kruislaan 403, 1098 SJ Amsterdam, the Netherlands, B. Kappen is with Radboud University Nijmegen, SNN, Geert Grooteplein 21, 6525 EZ Nijmegen, the Netherlands and D. Barber is with IDIAP, CH-1920 Martigny, Switzerland.

0000–0000/00$00.00 © 2004 IEEE

frame based generalized linear model (a sinusoidal model) and proposes inference by reversible-jump Markov Chain Monte Carlo (MCMC) algorithm. The main advantage of the model is that it makes no strong assumptions about the signal generation mechanism, and views the number of sources as well as the number of harmonics as unknown model parameters. Davy and Godsill [20] address some of the shortcomings of his model and allow changing amplitudes and frequency deviations. The reported results are encouraging, although the method is computationally very expensive.

B. Approach

Musical signals have a very rich temporal structure, both on a physical (signal) and a cognitive (symbolic) level. From a statistical modelling point of view, such a hierarchical structure induces very long range correlations that are difficult to capture with conventional signal models. Moreover, in many music applications, such as transcription or score following, we are usually interested in a symbolic representation (such as a score) and not so much in the "details" of the actual waveform. To abstract away from the signal details, we define a set of intermediate variables (a sequence of indicators), somewhat analogous to a "piano-roll" representation. This intermediate layer forms the "interface" between a symbolic process and the actual signal process. Roughly, the symbolic process describes how a piece is composed and performed. We view this process as a prior distribution on the piano-roll. Conditioned on the piano-roll, the signal process describes how the actual waveform is synthesized.

Most authors view automated music transcription as an "audio to piano-roll" conversion and usually consider "piano-roll to score" a separate problem. This view is partially justified, since source separation and transcription from a polyphonic source is already a challenging task. On the other hand, automated generation of a human readable score includes nontrivial tasks such as tempo tracking, rhythm quantization, meter and key induction [21], [22], [23]. As also noted by other authors (e.g. [17], [24], [25]), we believe that a model that integrates this higher level symbolic prior knowledge can guide and potentially improve the inferences, both in terms quality of a solution and computation time.

There are many different natural generative models for piano-rolls. In [26], we proposed a realistic hierarchical prior model. In this paper, we consider computationally simpler prior models and focus more on developing efficient inference techniques of a piano-roll representation. The organization of the paper is as follows: We will first present a generative model, inspired by additive synthesis, that describes the signal generation procedure. In the sequel, we will formulate two subproblems related to music transcription: melody identification and chord identification. We will show that both problems can be easily formulated as combinatorial optimization problems in the framework of our model, merely by redefining the prior on piano-rolls. Under our model assumptions, melody identification can be solved exactly in polynomial time (in the number of samples). By deterministic pruning, we obtain a practical approximation that works in linear time. Chord identification suffers from combinatorial explosion. For this case, we propose a greedy search algorithm based on iterative improvement. Consequently, we combine both algorithms for polyphonic music transcription. Finally, we demonstrate how (hyper-)parameters of the signal process can be estimated from real data.

II. Polyphonic Model

In a statistical sense, music transcription, (as many other perceptual tasks such as visual object recognition or robot localization) can be viewed as a latent state estimation problem: given the audio signal, we wish to identify the sequence of events (e.g. notes) that gave rise to the observed audio signal.

This problem can be conveniently described in a Bayesian framework: given the audio samples, we wish to infer a piano-roll that represents the onset times (e.g. times at which a 'string' is 'plucked'), note durations and the pitch classes of individual notes. We assume that we have one microphone, so that at each time t we have a one dimensional observed quantity y_t. Multiple microphones (such as required for processing stereo recordings) would be straightforward to include in our model. We denote the temporal sequence of audio samples $\{y_1, y_2, \ldots, y_t, \ldots, y_T\}$ by the shorthand notation $y_{1:T}$. A constant sampling frequency F_s is assumed.

Our approach considers the quantities we wish to infer as a collection of 'hidden' variables, whilst acoustic recording values $y_{1:T}$ are 'visible' (observed). For each observed sample y_t, we wish to associate a higher, unobserved quantity that labels the sample y_t appropriately. Let us denote the unobserved quantities by $\mathcal{H}_{1:T}$ where each $\mathcal{H}_t$ is a vector. Our hidden variables will contain, in addition to a piano-roll, other variables required to complete the sound generation procedure. We will elucidate their meaning later. As a general inference problem, the posterior distribution is given by Bayes' rule

$$p(\mathcal{H}_{1:T}|y_{1:T}) \quad \propto \quad p(y_{1:T}|\mathcal{H}_{1:T})p(\mathcal{H}_{1:T}) \qquad (1)$$

The likelihood term $p(y_{1:T}|\mathcal{H}_{1:T})$ in (1) requires us to specify a generative process that gives rise to the observed audio samples. The prior term $p(\mathcal{H}_{1:T})$ reflects our knowledge about piano-rolls and other hidden variables. Our modelling task is therefore to specify both how, knowing the hidden variable states (essentially the piano-roll), the microphone samples will be generated, and also to state a prior on likely piano-rolls. Initially, we concentrate on the sound generation process of a single note.

A. Modelling a single note

Musical instruments tend to create oscillations with modes that are roughly related by integer ratios, albeit with strong damping effects and transient attack characteristics [27]. It is common to model such signals as the sum of a periodic component and a transient non-periodic component (See e.g. [28], [29], [13]). The sinusoidal model [30] is often a good approximation that provides a compact representation for the periodic component. The transient component can be modelled as a correlated Gaussian noise process [16], [20]. Our signal model is also in the same spirit, but we will define it in state

Fig. 1. A damped oscillator in state space form. Left: At each time step, the state vector s rotates by ω and its length becomes shorter. Right: The actual waveform is a one dimensional projection from the two dimensional state vector. The stochastic model assumes that there are two independent additive noise components that corrupt the state vector s and the sample y, so the resulting waveform $y_{1:T}$ is a damped sinusoid with both phase and amplitude noise.

space form, because this provides a natural way to couple the signal model with the piano-roll representation. Similar formulations are used in the econometrics literature to model seasonal fluctuations, e.g. see [31], [32]. Here we omit the transient component and focus on the periodic component. It is conceptually straightforward to include the transient component as this does not affect the complexity of our inference algorithms.

First we consider how to generate a damped sinusoid y_t through time, with angular frequency ω. Consider a Gaussian process where typical realizations $y_{1:T}$ are damped "noisy" sinusoidal signals with angular frequency ω:

$$s_t \sim \mathcal{N}(\rho_t B(\omega) s_{t-1}, Q) \quad (2)$$

$$y_t \sim \mathcal{N}(C s_t, R) \quad (3)$$

$$s_0 \sim \mathcal{N}(0, S) \quad (4)$$

$$B(\omega) = \begin{pmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{pmatrix} \quad (5)$$

We use $\mathcal{N}(\mu, \Sigma)$ to denote a multivariate Gaussian distribution with mean μ and covariance Σ. Here $B(\omega)$ is a Givens rotation matrix that rotates two dimensional vector s_t by ω degrees counterclockwise. C is a projection matrix defined as $C = [1,\ 0]$. The phase and amplitude characteristics of y_t are determined by the initial condition s_0 drawn from a prior with covariance S. The damping factor $0 \leq \rho_t \leq 1$ specifies the rate at which s_t contracts to 0. See Figure 1 for an example. The transition noise variance Q is used to model deviations from an entirely deterministic linear model. The observation noise variance R models background noise.

In reality, musical instruments (with a definite pitch) have several modes of oscillation that are roughly located at integer multiples of the fundamental frequency ω. We can model such signals by a bank of oscillators giving a block diagonal transition matrix $A_t = A(\omega, \rho_t)$ defined as

$$\begin{pmatrix} \rho_t^{(1)} B(\omega) & 0 & \dots & 0 \\ 0 & \rho_t^{(2)} B(2\omega) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \rho_t^{(H)} B(H\omega) \end{pmatrix} \quad (6)$$

where H denotes the number of *harmonics*, assumed to be known. To reduce the number of free parameters we define each harmonic damping factor $\rho^{(h)}$ in terms of a basic ρ. A possible choice is to take $\rho_t^{(h)} = \rho_t^h$, motivated by the fact that damping factors of harmonics in a vibrating string scale approximately geometrically with respect to that of the fundamental frequency, i.e. higher harmonics decay faster [33]. $A(\omega, \rho_t)$ is the transition matrix at time t and encodes the physical properties of the sound generator as a first order Markov Process. The rotation angle ω can be made time dependent for modelling pitch drifts or vibrato. However, in this paper we will restrict ourselves to sound generators that produce sounds with (almost) constant frequency. The state of the sound generator is represented by s_t, a $2H$ dimensional vector that is obtained by concatenation of all the oscillator states in (2).

B. From Piano-Roll to Microphone

A piano-roll is a collection of indicator variables $r_{j,t}$, where $j = 1 \dots M$ runs over sound generators (i.e. notes or "keys" of a piano) and $t = 1 \dots T$ runs over time. Each sound generator has a unique fundamental frequency ω_j associated with it. For example, we can choose ω_j such that we cover all notes of the tempered chromatic scale in a certain frequency range. This choice is arbitrary and for a finer pitch analysis a denser grid with smaller intervals between adjacent notes can be used.

Each indicator is binary, with values "sound" or "mute". The essential idea is that, if previously muted, $r_{j,t-1}$ = "mute" an onset for the sound generator j occurs if $r_{j,t}$ = "sound". The generator continues to sound (with a characteristic damping decay) until it is again set to "mute", when the generated signal decays to zero amplitude (much) faster. The piano-roll, being a collection of indicators $r_{1:M,1:T}$, can be viewed as a binary sequence, e.g. see Figure 2. Each row of the piano-roll $r_{j,1:T}$ controls an underlying sound generator.

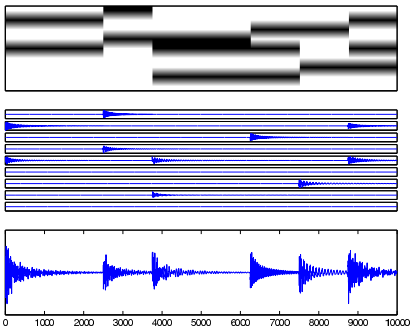


Fig. 2. Piano-roll. The vertical axis corresponds to the sound generator index j and the horizontal axis corresponds to time index t. Black and white pixels correspond to "sound" and "mute" respectively. The piano-roll can be viewed as a binary sequence that controls an underlying signal process. Each row of the piano-roll $r_{j,1:T}$ controls a sound generator. Each generator is a Gaussian process (a Kalman filter model), where typical realizations are damped periodic waveforms of a constant fundamental frequency. As in a piano, the fundamental frequency is a function of the generator index j. The actual observed signal $y_{1:T}$ is a superposition of the outputs of all generators.

The piano-roll determines the both sound onset generation, and the damping of the note. We consider first the damping effects.

1) Piano-Roll : Damping: Thanks to our simple geometrically related damping factors for each harmonic, we can characterise the damping factor for each note $j = 1, \dots, M$ by two decay coefficients ρ_{sound} and ρ_{mute} such that $1 \geq$

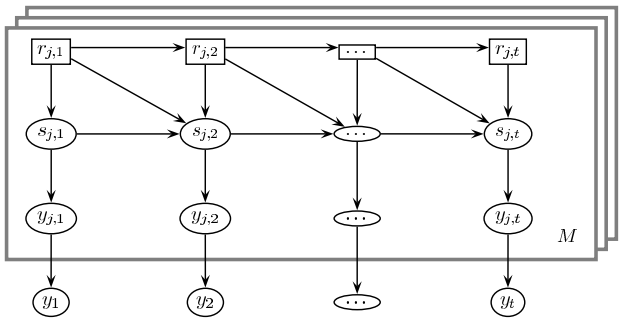


Fig. 3. Graphical Model. The rectangle box denotes "plates", M replications of the nodes inside. Each plate, $j = 1, \ldots, M$ represents the sound generator (note) variables through time.

$\rho_{\text{sound}} > \rho_{\text{mute}} > 0$. The piano-roll $r_{j,1:T}$ controls the damping coefficient $\rho_{j,t}$ of note j at time t by:

$$\rho_{j,t} = \rho_{\text{sound}}[r_{j,t} = \text{sound}] + \rho_{\text{mute}}[r_{j,t} = \text{mute}] \quad (7)$$

Here, and elsewhere in the article, the notation $[x = \text{text}]$ has value equal to 1 when variable x is in state text, and is zero otherwise. We denote the transition matrix as $A_j^{\text{mute}} \equiv A(\omega_j, \rho_{\text{mute}})$; similarly for A_j^{sound}.

2) Piano-Roll : Onsets: At each new onset, i.e. when $(r_{j,t-1} = \text{mute}) \rightarrow (r_{j,t} = \text{sound})$, the old state s_{t-1} is "forgotten" and a new state vector is drawn from a Gaussian prior distribution $\mathcal{N}(0, S)$. This models the energy injected into a sound generator at an onset (this happens, for example, when a guitar string is plucked). The amount of energy injected is proportional to the determinant of S and the covariance structure of S describes how this total energy is distributed among the harmonics. The covariance matrix S thus captures some of the timbre characteristics of the sound. The transition and observation equations are given by

$$\begin{aligned}
\text{isonset}_{j,t} &= (r_{j,t-1} = \text{mute} \wedge r_{j,t} = \text{sound}) && (8)\\
A_{j,t} &= [r_{j,t} = \text{mute}]A_j^{\text{mute}} + [r_{j,t} = \text{sound}]A_j^{\text{sound}} && (9)\\
s_{j,t} &\sim [\neg\text{isonset}_{j,,t}]\mathcal{N}(A_{j,t}\mathbf{s}_{t-1}, Q) \\
&\quad +[\text{isonset}_{j,t}]\mathcal{N}(0, S) && (10)\\
y_{j,t} &\sim \mathcal{N}(Cs_{j,t}, R) && (11)
\end{aligned}$$

In the above, C is a $1 \times 2H$ projection matrix $C = [1, 0, 1, 0, \ldots, 1, 0]$ with zero entries on the even components. Hence $y_{j,t}$ has a mean being the sum of the damped harmonic oscillators. R models the variance of the noise in the output of each sound generator. Finally, the observed audio signal is the superposition of the outputs of all sound generators,

$$y_t = \sum_j y_{j,t} \quad (12)$$

The generative model (7)-(12) can be described qualitatively by the graphical model in Figure 3. Equations (11) and (12) define $p(y_{1:T}|s_{1:M,1:T})$. Equations (7) (9) and (10) relate r and s and define $p(s_{1:M,1:T}|r_{1:M,1:T})$. In this paper, the prior model $p(r_{1:M,1:T})$ is Markovian and has the following factorial structure [1]:

$$p(r_{1:M,1:T}) = \prod_m \prod_t p(r_{m,t}|r_{m,t-1})$$

C. Inference

Given the polyphonic model described in section II, to infer the most likely piano-roll we need to compute

$$r^*_{1:M,1:T} = \operatorname*{argmax}_{r_{1:M,1:T}} p(r_{1:M,1:T}|y_{1:T}) \quad (13)$$

where the posterior is given by

$$\begin{aligned}
p(r_{1:M,1:T}|y_{1:T}) &= \frac{1}{p(y_{1:T})} \int ds_{1:M,1:T}\, p(y_{1:T}|s_{1:M,1:T}) \\
&\quad \times p(s_{1:M,1:T}|r_{1:M,1:T})p(r_{1:M,1:T})
\end{aligned}$$

The normalization constant, $p(y_{1:T})$, obtained by summing the integral term over all configurations $r_{1:M,1:T}$ is called the evidence. [2]

Unfortunately, calculating this most likely piano-roll configuration is generally intractable, and is related to the difficulty of inference in Switching Kalman Filters [35], [36]. We shall need to develop approximation schemes for this general case, to which we shall return in a later section.

As a prelude, we consider a slightly simpler, related model which aims to track the pitch (melody identification) in a monophonic instrument (playing only a single note at a time), such as a flute. The insight gained here in the inference task will guide us to a practical approximate algorithm in the more general case later.

III. Monophonic Model

Melody identification, or monophonic pitch tracking with onset and offset detection, can be formulated by a small modification of our general framework. Even this simplified task is still of huge practical interest, e.g. in real time MIDI conversion for controlling digital synthesizers using acoustical instruments or pitch tracking from the singing voice. One important problem in real time pitch tracking is the time/frequency tradeoff: to estimate the frequency accurately, an algorithm needs to collect statistics from a sufficiently long interval. However, this often conflicts with the real time requirements.

In our formulation, each sound generator is a dynamical system with a sequence of transition models, sound and mute.

[1] In the simulations we have fixed the transition parameter $p(r = \text{mute}|r = \text{sound}) = p(r = \text{sound}|r = \text{mute}) = 10^{-7}$

[2] It is instructive to interpret (13) from a Bayesian model selection perspective [34]. In this interpretation, we view the set of all piano-rolls, indexed by configurations of discrete indicator variables $r_{1:M,1:T}$, as the set of all models among which we search for the best model $r^*_{1:M,1:T}$. In this view, state vectors $s_{1:M,1:T}$ are the model parameters that are integrated over. It is well known that the conditional predictive density $p(y|r)$, obtained through integration over s, automatically penalizes more complex models, when evaluated at $y = y_{1:T}$. In the context of piano-roll inference, this objective will automatically prefer solutions with less notes. Intuitively, this is simply because at each note onset, the state vector s_t is reinitialized using a broad Gaussian $\mathcal{N}(0, S)$. Consequently, a configuration r with more onsets will give rise to a conditional predictive distribution $p(y|r)$ with a larger covariance. Hence, a piano-roll that claims the existence of additional onsets without support from data will get a lower likelihood.

The state s evolves first according to the sounding regime with transition matrix A^{sound} and then according to the muted regime with A^{mute}. The important difference from a general switching Kalman filter is that when the indicator r switches from mute to sound, the old state vector is "forgotten". By exploiting this fact, in the appendix I-A we derive, for a single sound generator (i.e. a single note of a fixed pitch that gets on and off), an exact polynomial time algorithm for calculating the evidence $p(y_{1:T})$ and MAP configuration $r^*_{1:T}$.

1) Monophonic pitch tracking: Here we assume that at any given time t only a single sound generator can be sounding, i.e. $r_{j,t} = \text{sound} \Rightarrow r_{j',t} = \text{mute}$ for $j' \neq j$. Hence, for practical purposes, the factorial structure of our original model is redundant; i.e. we can "share" a single state vector s among all sound generators[3]. The resulting model will have the same graphical structure as a single sound generator but with an indicator $j_t \in 1 \dots M$ which indexes the active sound generator, and $r_t \in \{\text{sound}, \text{mute}\}$ indicates sound or mute. Inference for this case turns out to be also tractable (i.e. polynomial). We allow switching to a new j' only after an onset. The full generative model using the pairs (j_t, r_t), which includes both likelihood and prior terms is given as

$$
\begin{aligned}
r_t &\sim p(r_t|r_{t-1}) \\
\text{isonset}_t &= (r_t = \text{sound} \wedge r_{t-1} = \text{mute}) \\
j_t &\sim [\neg\text{isonset}_t]\delta(j_t; j_{t-1}) + [\text{isonset}_t]u(j_t) \\
A_t &= [r_t = \text{mute}]A^{\text{mute}}_{j_t} + [r_t = \text{sound}]A^{\text{sound}}_{j_t} \\
s_t &\sim [\neg\text{isonset}_t]\mathcal{N}(A_t s_{t-1}, Q) + [\text{isonset}_t]\mathcal{N}(0, S) \\
y_t &\sim \mathcal{N}(C s_t, R)
\end{aligned}
$$

Here $u(j)$ denotes a uniform distribution on $1, \dots, M$ and $\delta(j_t; j_{t-1})$ denotes a degenerate (deterministic) distribution concentrated on j_t, i.e. unless there is an onset the active sound generator stays the same. Our choice of a uniform $u(j)$ simply reflects the fact that any new note is as likely as any other. Clearly, more informative priors, e.g. that reflect knowledge about tonality, can also be proposed. Similarly, for doing a more precise pitch analysis, we may choose a finer grid such that $\omega_{j+1}/\omega_j = \mathcal{Q}$. Here, $\mathcal{Q}$ is the quality factor, a measure of the desired frequency precision not to be confused with the transition noise Q.

The graphical model is shown in Figure 4. The derivation of the polynomial time inference algorithm is given in appendix I-C. Technically, it is a simple extension of the single note algorithm derived in appendix I-A.

In Figure 5, we illustrate the results on synthetic data sampled from the model where we show the filtering density $p(r_t, j_t|y_{1:t})$. After an onset, the posterior becomes quickly crisp, long before we observe a complete cycle. This feature is especially attractive for real time applications where a reliable pitch estimate has to be obtained as early as possible.

We conclude this subsection with an illustration on real data. We have recorded a major scale on an electric bass and downsampled from the original sampling rate of $F_s = 22050$ by a factor of $D = 10$. We have estimated parameters for a signal model with $H = 8$ harmonics. The "training set" consisted of a single note recorded from the same instrument; this procedure will be discussed in more detail in section V. We have estimated the MAP configuration $(r, j)_{1:T}$ using the algorithm described in appendix I-C. The figure shows that the estimated piano roll is quite precise. We have repeated the experiment on a pianoroll with a pitch grid of $1/4$ semitones ($\mathcal{Q} = 2^{1/48}$). The results reveal that the 5'th and 7'th degree of the scale were intonated slightly low, which did not had much effect on the estimation of the pitch class when using a coarser grid. In the last experiment we have trained the model parameters using a note sung by a vocalist. As expected, the results are poorer; in particular we observe that 5'ths or octaves are confused due to the different harmonic structure and transition characteristics.

[3]We ignore the cases when two or more generators are simultaneously in the mute state.

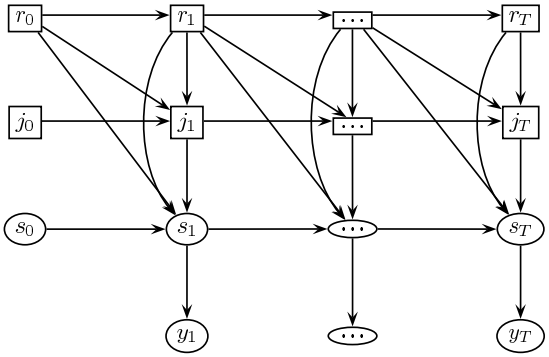


Fig. 4. Simplified Model for monophonic transcription. Since there is only a single sound generator active at any given time, we can represent a piano-roll at each time slice by the tuple (j_t, r_t) where j_t is the index of the active sound generator and $r_t \in \{\text{sound}, \text{mute}\}$ indicates the state.

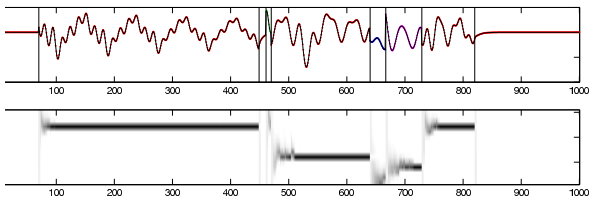
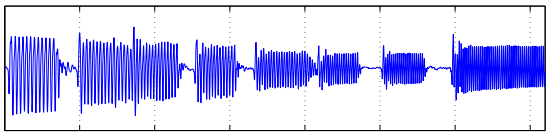
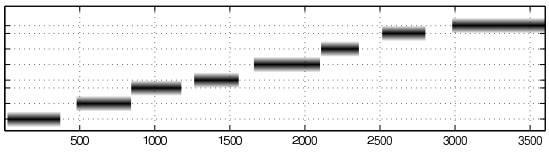


Fig. 5. Monophonic pitch tracking. (Top) Synthetic data sampled from model in Figure 4. Vertical bars denote the onset and offset times. (Bottom) The filtering density $p(r_t, j_t|y_{1:t})$. The vertical axis denotes the sound generator index j_t and the gray level denotes the posterior probability $p(r_t = \text{sound}, j_t|y_{1:t})$ where black corresponds to 1.

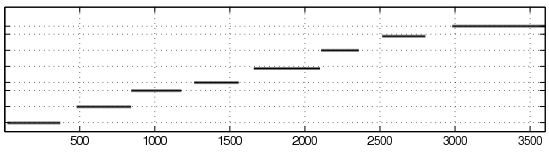
2) Extension to vibrato and legato: The monophonic model has been constructed such that the rotation angle ω remains constant. Although the the transition noise with variance Q still allows for small and independent deviations in frequencies of the harmonics, the model is not realistic for situations with systematic pitch drift or fluctuation, e.g. as is the case with vibrato. Moreover, on many musical instruments, it is possible to play *legato*, that is without an explicit onset between note boundaries. In our framework, pitch drift and legato can be modelled as a sequence of transition models. Consider the



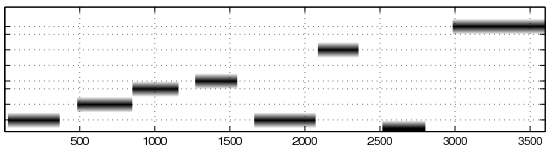
(a) F major scale played on an electric bass, sampled with $F_s = 22050$ and downsampled by a factor of $D = 10$.



(b) Estimated MAP configuration $(r, j)_{1:T}$ with with quality factor $\mathcal{Q} = 2^{1/12}$.



(c) A finer analysis with $\mathcal{Q} = 2^{1/48}$ reveals that the 5'th and 7'th degree of the scale are intonated slightly low.



(d) Estimated piano roll, when signal model is trained on a different instrument. Here, results are poorer since parameters were estimated using human voice, which has different spectral and temporal characteristics.

Fig. 6. Monophonic pitch estimation on real data.

generative process for the note index j:

$$\begin{aligned} r_t &\sim p(r_t|r_{t-1}) \\ \text{isonset}_t &= (r_t = \text{sound} \wedge r_{t-1} = \text{mute}) \\ \text{issound}_t &= (r_t = \text{sound} \wedge r_{t-1} = \text{sound}) \\ j_t &\sim [\text{issound}_t] d(j_t|j_{t-1}) + \\ &\quad [r_t = \text{mute}]\delta(j_t; j_{t-1}) + [\text{isonset}_t] u(j_t) \end{aligned}$$

Here, $d(j_t|j_{t-1})$ is a multinomial distribution reflecting our prior belief how likely is it to switch between notes. When r_t = mute, there is no regime change, reflected by the deterministic distribution $\delta(j_t; j_{t-1})$ peaked around j_{t-1}. Remember that neighbouring notes have also close fundamental frequency ω. To simulate pitch drift, we choose a fine grid such that $\omega_j/\omega_{j+1} = \mathcal{Q}$. In this case, we can simply define $d(j_t|j_{t-1})$ as a multinomial distribution with support on $[j_{t-1} - 1, j_{t-1}, j_{t-1} + 1]$ with cell probabilities $[d_{-1}\ \ d_0\ \ d_1]$.

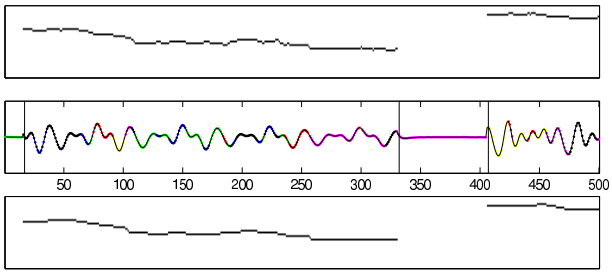


Fig. 7. Tracking varying pitch. Top and middle panel show the true piano-roll and the sampled signal. The estimated piano-roll is shown below.

We can take a larger support for $d(j_t|j_{t-1})$, but in practice we would rather reduce the frequency precision $\mathcal{Q}$ to avoid additional computational cost.

Unfortunately, the terms included by the drift mechanism render an exact inference procedure intractable. We derive the details of the resulting algorithm in the appendix I-D. A simple deterministic pruning method is described in appendix II-A. In Figure 7, we show the estimated MAP trajectory $r^*_{1:T}$ for drifting pitch. We use a model where the quality factor is $\mathcal{Q} = 2^{-120}$, (120 generators per octave) with drift probability $d_{-1} = d_1 = 0.1$. A fine pitch contour, that is accurate to sample precision, can be estimated.

IV. Polyphonic Inference

In this section we return to the central goal of inference in the general polyphonic model described in section II. To infer the most likely piano-roll we need to compute $\operatorname{argmax}_{r_{1:M,1:T}} p(r_{1:M,1:T}|y_{1:T})$ defined in (13). Unfortunately, the calculation of (13) is intractable. Indeed, even the calculation of the Gaussian integral conditioned on a particular configuration $r_{1:M,1:T}$ using standard Kalman filtering equations is prohibitive since the dimension of the state vector is $|s| = 2H \times M$, where H is the number of harmonics. For a realistic application we may have $M \approx 50$ and $H \approx 10$. It is clear that unless we are able to develop efficient approximation techniques, the model will be only of theoretical interest.

A. Vertical Problem: Chord identification

Chord identification is the simplest polyphonic transcription task. Here we assume that a given audio signal $y_{1:T}$ is generated by a piano-roll where $r_{j,t} = r_j$ for all[4] $j = 1 \ldots M$. The task is to find the MAP configuration

$$r^*_{1:M} = \operatorname*{argmax}_{r_{1:M}} p(y_{1:T}, r_{1:M})$$

Each configuration corresponds to a chord. The two extreme cases are "silence" and "cacophony" that correspond to configurations $r_{1:M}$[mute mute ... mute] and [sound sound ... sound] respectively. The size of the search space in this case 2^M, which is prohibitive for direct computation.

[4]We will assume that initially we start from silence where $r_{j,0}$ = mute for all $j = 1 \ldots M$

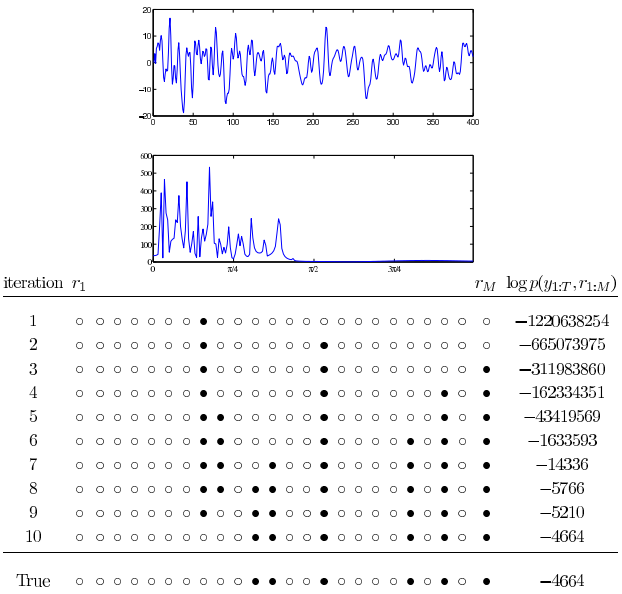


Fig. 8. We have first drawn a random piano-roll configuration (a random chord) $r_{1:M}$. Given $r_{1:M}$, we generate a signal of length 400 samples with a sampling frequency $F_s = 4000$ from $p(y_{1:T}|r_{1:M})$. We assume 24 notes (2 octaves). The synthesized signal from the generative model and its discrete time Fourier transform modulus are shown above. The true chord configuration and the associated log probability is at the bottom of the table. For the iterative algorithm, the initial configuration in this example was silence. At this point we compute the probability for each single note configurations (all one flip neighbours of silence). The first note that is added is actually not present in the chord. Until iteration 9, all iterations add extra notes. Iteration 9 and 10 turn out to be removing the extra notes and iterations converge to the true chord. The intermediate configurations visited by the algorithm are shown in the table below. Here, sound and mute states are represented by •'s and ○'s.

A simple approximation is based on greedy search: we start iterative improvement from an initial configuration $r^{(0)}_{1:M}$ (silence, or randomly drawn from the prior). At each iteration i, we evaluate the probability $p(y_{1:T}, r_{1:M})$ of all neighbouring configurations of $r^{(i-1)}_{1:M}$. We denote this set by $\mathrm{neigh}(r^{(i-1)}_{1:M})$. A configuration $r' \in \mathrm{neigh}(r)$, if r' can be reached from r within a single flip (i.e., we add or remove single notes). If $r^{(i-1)}_{1:M}$ has a higher probability than all its neighbours, the algorithm terminates, having found a local maximum. Otherwise, we pick the neighbour with the highest probability and set

$$r^{(i)}_{1:M} = \operatorname*{argmax}_{r_{1:M} \in \mathrm{neigh}(r^{(i-1)}_{1:M})} p(y_{1:T}, r_{1:M})$$

and iterate until convergence. We illustrate the algorithm on a signal sampled from the generative model, see Figure 8. This procedure is guaranteed to converge to a (possibly local) maxima. Nevertheless, we observe that for many examples this procedure is able to identify the correct chord. Using multiple restarts from different initial configurations will improve the quality of the solution at the expense of computational cost.

One of the advantages of our generative model based approach is that we can in principle infer a chord given any subset of data. For example, we can simply downsample $y_{1:T}$ (without any preprocessing) by an integer factor of D and view the discarded samples as missing values. Of course, when D is large, i.e. when we throw away many samples, due to diminishing likelihood contribution, we obtain a diffuse posterior on the piano-roll and eventually the results will be poorer.

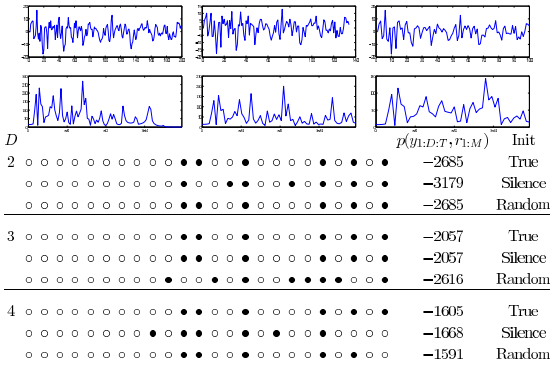


Fig. 9. Iterative improvement results when data are subsampled by a factor of $D = 2, 3$ and 4, respectively. For each factor D, the top line shows the true configuration and the corresponding probability. The second line is the solution found by starting from silence and the third line is starting from a random configuration drawn form the prior (best of 3 independent runs).

In Figure 9, we show the results of such an experiment. We have downsampled $y_{1:T}$ with factor $D = 2, 3$ and 4. The energy spectrum is quite coarse due to the short length of the data. Consequently many harmonics are not resolved, e.g. we can not identify the underlying line spectrum by visual inspection. Methods based on template matching or identification of peaks may have serious problems for such examples. On the other hand, our model driven approach is able to identify the true chord. We note that, the presented results are illustrative only and the actual behaviour of the algorithm (sensitivity to D, importance of starting configuration) will depend on the details of the signal model.

B. Piano-Roll inference Problem: Joint Chord and Melody identification

The piano-roll estimation problem can be viewed as an extension of chord identification in that we also detect onsets and offsets for each note within the analysis frame. A practical approach is to analyze the signal in sufficiently short time windows and assume that for each note, at most one changepoint can occur within the window.

Consider data in a short window, say $y_{1:W}$. We start iterative improvement from a configuration $r^{(0)}_{1:M,1:W}$, where each time slice $r^{(0)}_{1:M,t}$ for $t = 1 \ldots W$ is equal to a "chord" $r_{1:M,0}$. The chord $r_{1:M,0}$ can be silence or, during a frame by frame analysis, the last time slice of the best configuration found in the previous analysis window. Let the configuration at $i-1$'th iteration be denoted as $r^{(i-1)}_{1:M,1:W}$. At each new iteration i, we evaluate the posterior probability $p(y_{1,W}, r_{1:M,1:W})$, where $r_{1:M,1:W}$ runs over all neighbouring configurations of $r^{(i-1)}_{1:M,1:W}$. Each member $r_{1:M,1:W}$ of the neighbourhood is generated as follows: For each $j = 1 \ldots M$, we clamp all the other rows, i.e. we set $r_{j',1:W} = r^{(i-1)}_{j',1:W}$ for $j' \neq j$. For each time step $t = 1 \ldots W$, we generate a new configuration such that the switches up to time t are equal to the initial switch $r_{j,0}$, and its opposite $\neg r_{j,0}$ after t, i.e.

$r_{j,t'} = r_{j,0}[t' < t] + \neg r_{j,0}[t' \geq t]$. This is equivalent to saying that a sounding note may get muted, or a muted note may start to sound. The computational advantage of allowing only one changepoint at each row is that the probability of all neighbouring configurations for a fixed j can be computed by a single backward, forward pass [23], [36]. Finally, we pick the neighbour with the maximum probability. The algorithm is illustrated in Figure 10.

The analysis for the whole sequence proceeds as follows: Consider two successive analysis windows $Y_{\text{prev}} \equiv y_{1:W}$ and $Y \equiv y_{W+1:2W}$. Suppose we have obtained a solution $R^*_{\text{prev}} \equiv r^*_{1:M,1:W}$ obtained by iterative improvement. Conditioned on R^*_{prev}, we compute the posterior $p(s_{1:M,W}|Y_{\text{prev}}, R^*_{\text{prev}})$ by Kalman filtering. This density is the prior of s for the current analysis window Y. The search starts from a chord equal to the last time slice of R^*_{prev}. In Fig. 11 we show an illustrative result obtained by this algorithm on synthetic data. In similar experiments with synthetic data, we are often able to identify the correct piano-roll.

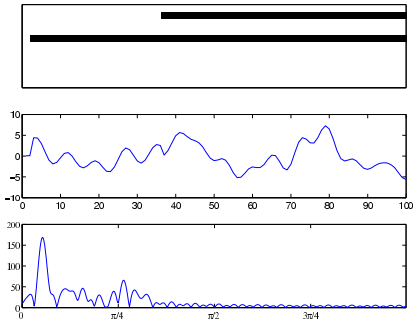
This simple greedy search procedure is somewhat sensitive to location of onsets within the analysis window. Especially, when an onset occurs near the end of an analysis window, it may be associated with an incorrect pitch. The correct pitch is often identified in the next analysis window, when a longer portion of the signal is observed. However, since the basic algorithm does not allow for correcting the previous estimate by retrospection, this introduces some artifacts. A possible method to overcome this problem is to use a fixed lag smoothing approach, where we simply carry out the analysis on overlapping windows. For example, for an analysis window $Y_{\text{prev}} \equiv y_{1:W}$, we find $r^*_{1:M,1:W}$. The next analysis window is taken as $y_{L+1:W+L}$ where $L \leq W$. We find the prior $p(s_{1:M,L}|y_{1:L}, r^*_{1:M,1:L})$ by Kalman filtering. On the other hand, obviously, the algorithm becomes slower by a factor of L/W.

An optimal choice for L and W will depend upon many factors such as signal characteristics, sampling frequency, downsampling factor D, onset/offset positions, number of active sound generators at a given time as well as the amount of CPU time available. In practice, these values may be critical and they need to be determined by trial and error. On the other hand, it is important to note that L and W just determine how the approximation is made but not enter the underlying model.

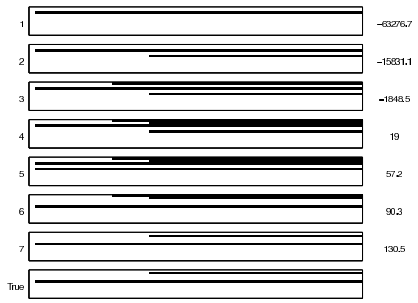
V. Learning

In the previous sections, we assumed that the correct signal model parameters $\theta = (S, \rho, Q, R)$ were known. These include in particular the damping coefficients $\rho_{\text{sound}}, \rho_{\text{mute}}$, transition noise variance Q, observation noise R and the initial prior covariance matrix S after an onset. In practice, for an instrument class (e.g. plucked string instruments) a reasonable range for θ can be specified a-priori. We may safely assume that θ will be static (not time dependent) during a given performance. However, exact values for these quantities will vary among different instruments (e.g. old and new strings) and recording/performance conditions.

One of the well-known advantages of Bayesian inference is that, when uncertainty about parameters is incorporated



(a)



(b)

Fig. 10. Iterative improvement with changepoint detection. The true piano-roll, the signal and its Fourier transform magnitude are shown in Figure 10.(a). In Figure 10.(b), configurations $r^{(i)}$ visited during iterative improvement steps. Iteration numbers i are shown left and the corresponding probability is shown on the right. The initial configuration (i.e. "chord") $r_{1:M,0}$ is set to silence. At the first step, the algorithm searches all single note configurations with a single onset. The winning configuration is shown on top panel of Figure 10.(b). At the next iteration, we clamp the configuration for this note and search in a subset of two note configurations. This procedure adds and removes notes from the piano-roll and converges to a local maxima. Typically, the convergence is quite fast and the procedure is able to identify the true chord without making a "detour" as in (b).

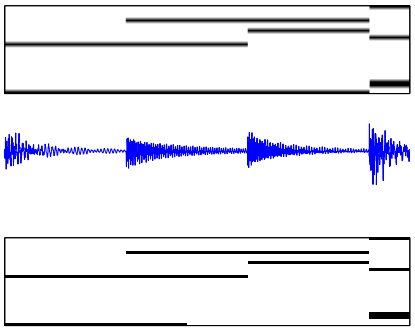


Fig. 11. A typical example for Polyphonic piano-roll inference from synthetic data. We generate a realistic piano-roll (top) and render a signal using the polyphonic model (middle). Given only the signal, we estimate the piano-roll by iterative improvement in successive windows (bottom). In this example, only the offset time of the lowest note is not estimated correctly. This is a consequence that, for long notes, the state vector s converges to zero before the generator switches to the mute state.

in a model, this leads in a natural way to the formulation of a learning algorithm. The piano-roll estimation problem, omitting the time indices, can be stated as follows:

$$r^* = \underset{r}{\operatorname{argmax}} \int d\theta \int ds\; p(y|s,\theta)p(s|r,\theta)p(\theta)p(r) \qquad (14)$$

In other words, we wish to find the best piano-roll by taking into account all possible settings of the parameter θ, weighted by the prior. Note that (14) becomes equivalent to (13), if we knew the "best" parameter θ^*, i.e. $p(\theta) = \delta(\theta - \theta^*)$. Unfortunately, the integration on θ can not be calculated analytically and approximation methods must be used [37]. A crude but computationally cheap approximation replaces the integration on θ in (14) with maximization:

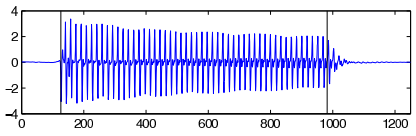
$$r^* = \underset{r}{\operatorname{argmax}} \max_{\theta} \int ds\; p(y|s,\theta)p(s|r,\theta)p(\theta)p(r)$$

Essentially, this is a joint optimization problem on piano-rolls and parameters which we solve by a greedy coordinate ascent algorithm. The algorithm we propose is a double loop algorithm where we iterate in the outer loop between maximization over r and maximization over θ. The latter maximization itself is calculated with an iterative algorithm:

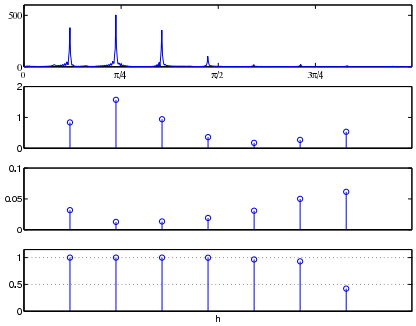
$$r^{(i)} = \underset{r}{\operatorname{argmax}} \int ds p(y|s,\theta^{(i-1)})p(s|r,\theta^{(i-1)})p(\theta^{(i-1)})p(r)$$
$$\theta^{(i)} = \underset{\theta}{\operatorname{argmax}} \int ds p(y|s,\theta)p(s|r^{(i)},\theta)p(\theta)p(r^{(i)})$$

For a single note, conditioned on a fixed $\theta^{(i-1)}$, $r^{(i)}$ can be calculated exactly, using the message propagation algorithm derived in appendix I-B. Conditioned on $r^{(i)}$, maximization on the θ coordinate becomes equivalent to parameter estimation in linear dynamical systems, for which no closed form solution is known. Nevertheless, this step can be calculated by an iterative expectation maximization (EM) algorithm [36], [38]. In practice, we observe that for realistic starting conditions $\theta^{(0)}$, the $r^{(i)}$ are identical, suggesting that the best segmentation r^* is not very sensitive to variations in θ near to a local optimum. In Figure 12, we show the results of training the signal model based on a single note (a C from the low register) of an electric bass.

In an experiment with real data, we illustrate the performance of the model for two and three note polyphony (See Fig.13). We have recorded three separate monophonic melodies; ascending modes of the major scale starting from the root, 3'rd and 5'th degree of a major scale. We have estimated model parameters using a single note from the same register. For each monophonic melody, we have calculated the ground truth $r^{\text{true}}_{1:M,1:T}$ by the algorithm described in section III-.1. We have constructed the two note example by adding the first two melodies. The analysis is carried out using a window length of $W = 200$ samples, without overlap between analysis frames (i.e. $L = W$). We were able to identify the correct pitch classes for the two note polyphony case. However, especially some note offsets are not detected correctly. In the three note case, pitch classes are correct, but there are also more artifacts, e.g. the chord around sample index 500 is identified incorrect. We expect results to go worse with increasing polyphony; this



(a) A single note from an electric bass. Original sampling rate of 22050 Hz is reduced by downsampling with factor $D = 20$. Vertical lines show the changepoints of the MAP trajectory $r_{1:K}$.



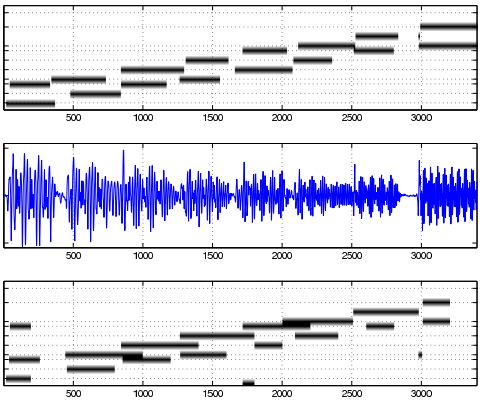
(b) Top to Bottom: Fourier transform of the downsampled signal and diagonal entries of S, Q and damping coefficients ρ_{sound} for each harmonic.

Fig. 12. Training the signal model with EM from a single note from an electric bass using a sampling rate of 22050 Hz. The original signal is downsampled by a factor of $D = 20$. Given some crude first estimate for model parameters $\theta^{(0)}(S,\rho,Q,R)$, we estimate $r^{(1)}$, shown in (a). Conditioned on $r^{(1)}$, we estimate the model parameters $\theta^{(1)}$ and so on. Let S_h denote the 2×2 block matrix from the diagonal S, corresponding to the h'th harmonic, similarly for Q_h. In (b), we show the estimated parameters for each harmonic sum of diagonal elements, i.e. $\mathbf{Tr}\, S_h$ and $\mathbf{Tr}\, Q_h$. The damping coefficient is found as $\rho_{\text{sound}} = (\det A_h A_h^T)^{1/4}$ where A_h is a 2×2 diagonal block matrix of transition matrix A^{sound}. For reference, we also show the Fourier transform modulus of the downsampled signal. We can see, that on the low frequency bands, S mimics the average energy distribution of the note. However, transient phenomena, such as the strongly damped 7'th harmonic with relatively high transition noise, is hardly visible in the frequency spectrum. On the other hand for online pitch detection, such high frequency components are important to generate a crisp estimate as early as possible.

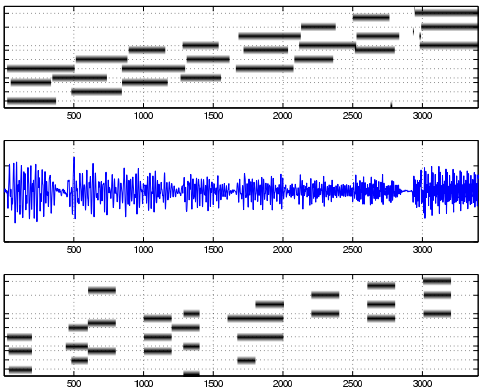
behaviour is qualitatively similar to other methods reported in the literature, e.g. [18], [19], but clearly, more simulation studies have to be carried out for an objective comparison. Investigating the log likelihood ratio

$$\log \frac{p(y_{1:T}|r^{\text{true}}_{1:M,1:T})p(r^{\text{true}}_{1:M,1:T})}{p(y_{1:T}|r^*_{1:M,1:T})p(r^*_{1:M,1:T})} \gg 0$$

suggests that the failure is due to the suboptimal estimation procedure, i.e. the model prefers the true solution but our greedy algorithm is unable to locate it and gets stuck in $r^*_{1:M,1:T}$, where r^* denotes here the configuration found by the algorithm. In the conclusions section, we will discuss some alternative approximation methods to improve results.



(a) (Top) The ground truth estimated when all melodies are transcribed separately. (Middle) The superposition of first two melodies downsampled by a factor of $D = 10$. (Bottom) Piano-roll estimated with an analysis window of size $W = 200$ samples, without overlap between analysis frames.



(b) Result for the same experiment with three notes polyphony.

Fig. 13. Experiment with two and three note polyphony. Three monophonic melodies (ascending modes of the F major scale starting from the root, 3'rd and 5'th degree) are recorded and transcribed separately to obtain the ground truth. Model parameters are learned using a single note recorded from the same register. Finally, melodies are added to create two- and three-note polyphonic examples.

VI. Discussion

We have presented a model driven approach where transcription is viewed as a Bayesian inference problem. In this respect, at least, our approach parallels the previous work of [19], [20], [39]. We believe, however, that our formulation, based on a switching state space model, has several advantages. We can remove the assumption of a frame based model and this enables us to analyse music online and to sample precision. Practical approximations to an eventually intractable exact posterior can be carried out frame-by-frame, such as by using a fixed time-lag smoother. This, however, is merely a computational issue (albeit a very important one). We may also discard samples to reduce computational burden, and account for this correctly in our model.

An additional advantage of our formulation is that we can still deliver a pitch estimate even when the fundamental and lower harmonics of the frequency band are missing. This is related to so called *virtual pitch* perception [40]: we tend to associate notes with a pitch class depending on the relationship between harmonics rather than the frequency of the fundamental component itself.

There is a strong link between model selection and polyphonic music transcription. In chord identification we need to compare models with different number of notes, and in melody identification we need to deduce the number of onsets. Model selection becomes conceptually harder when one needs to compare models of different size. We partially circumvent this difficulty by using switch variables, which implicitly represent the number of components.

Following the established signal processing jargon, we may call our approach a time-domain method, since we are not explicitly calculating a discrete-time Fourier transform. On the other hand, the signal model presented here has close links to the Fourier analysis and sinusoidal modelling. Our analysis can be interpreted as a search procedure for a sparse representation on a set of basis vectors. In contrast to Fourier analysis, where the basis vectors are sinusoids (e.g. see [41] for a Bayesian treatment), we represent the observed signal implicitly using signals drawn from a stochastic process which typically generates decaying periodic oscillations (e.g. notes) with occasional changepoints. The sparsity of this representation is a consequence of the onset mechanism, that effectively puts a mixture prior over the hidden state vector s. This prior is peaked around zero and has broad tails, indicating that most of the sources are muted and only a few are sounding. It is well known that such Gaussian mixture priors induce sparse representations, e.g. see [42], [43] for applications in the context of source separation.

A. Future work

Although our approach has many desirable features (automatically deducing number of correct notes, high temporal resolution e.t.c.), one of the main disadvantage of our method is computational cost associated with updating large covariance matrices in Kalman filtering. It would be very desirable to investigate approximation schemas that employ fast transformations such as the FFT to accelerate computations.

When transcribing music, human experts rely heavily on prior knowledge about the musical structure – harmony, tempo or expression. Such structure can be captured by training probabilistic generative models on a corpus of compositions and performances by collecting statistics over selected features (e.g. [44]). One of the important advantages of our approach is that such prior knowledge about the musical structure can be formulated as an informative prior on a piano-roll; thus can be integrated in signal analysis in a consistent manner. We believe that investigation of this direction is important in designing robust and practical music transcription systems.

Our signal model considered here is inspired by additive synthesis. An advantage of our linear formulation is that

we can use the Kalman filter recursions to integrate out the continuous latent state analytically. An alternative would be to formulate a nonlinear dynamical system that implements a nonlinear synthesis model (e.g. FM synthesis, waveshaping synthesis, or even a physical model[45]). Such an approach would reduce the dimensionality of the latent state space but force us to use approximate integration methods such as particle filters or EKF/UKF [46]. It remains an interesting open question whether, in practice, one should trade-off analytical tractability versus reduced latent state dimension.

In this paper, for polyphonic transcription, we have used a relatively simple deterministic inference method based on iterative improvement. The basic greedy algorithm, whilst still potentially useful in practice, may get stuck in poor solutions. We believe that, using our model as a framework, better polyphonic transcriptions can be achieved using more elaborate inference or search methods. For example, computation time associated with exhaustive search of the neighbourhood for all visited configuations could be significantly reduced by randomizing the local search (e.g. by Metropolis-Hastings moves [5]) or use heuristic proposal distributions derived from easy-to-compute features such as the energy spectrum. Alternatively, sequential Monte Carlo methods or deterministic message propagation algorithms such as Expectation propagation (EP) [47] could be also used.

We have not yet tested our model for more general scenarios, such as music fragments containing percussive instruments or bell sounds with inharmonic spectra. Our simple periodic signal model would be clearly inadequate for such a scenario. On the other hand, we stress the fact that the framework presented here is not limited to the analysis of signals with harmonic spectra, and in principle applicable to any family of signals that can be represented by a switching state space model. This is already a large class since many real-world acoustic processes can be approximated well with piecewise linear regimes. We can also formulate a joint estimation schema for unknown parameters as in (14) and integrate them out (e.g. see [20]). However, this is currently a hard and computationally expensive task. If efficient and accurate approximate integration methods can be developed, our model will be applicable to mixtures of many different types of acoustical signals and may be useful in more general auditory scene analysis problems.

1) Acknowledgments: This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Dutch Ministry of Economic Affairs. We would like to thank the the anonymous reviewers for their comments and suggestions.

APPENDIX I
DERIVATION OF MESSAGE PROPAGATION ALGORITHMS

In the appendix, we derive several exact message propagation algorithms. Our derivation closely follows the standard derivation of recursive prediction and update equations for the Kalman filter [48]. First we focus on a single sound generator. In appendix I-A and I-B, we derive polynomial time algorithms for calculating the evidence $p(y_{1:T})$ and MAP configuration $r^*_{1:T} = \operatorname{argmax}_{r_{1:T}} p(y_{1:T}, r_{1:T})$ respectively. The MAP configuration is useful for onset/offset detection. In the following section, we extend the onset/offset detection algorithms to monophonic pitch tracking with constant frequency. We derive a polynomial time algorithm for this case in appendix I-C. The case for varying fundamental frequency is derived in the following appendix I-D. In appendix II we describe heuristics to reduce the amount of computations.

A. Computation of the evidence $p(y_{1:T})$ for a single sound generator by forward filtering

We assume a Markovian prior on the indicators r_t where $p(r_t = i | r_{t-1} = j) \equiv p_{i,j}$. For convenience, we repeat the generative model for a single sound generator by omitting the note index j.

$$\begin{aligned} r_t &\sim p(r_t|r_{t-1}) \\ \text{isonset}_t &= (r_t = \text{sound} \wedge r_{t-1} = \text{mute}) \\ s_t &\sim [\neg \text{isonset}_t]\mathcal{N}(A_{r_t} s_{t-1}, Q) + [\text{isonset}_t]\mathcal{N}(0, S) \\ y_t &\sim \mathcal{N}(C s_t, R) \end{aligned}$$

For simplicity, we will sometime use the labels 1 and 2 to denote sound and mute respectively. We enumerate the transition models as $f_{r_t}(s_t|s_{t-1}) = \mathcal{N}(A_{r_t} s_{t-1}, Q)$. We define the filtering potential as

$$\alpha_t \equiv p(y_{1:t}, s_t, r_t, r_{t-1}) = \sum_{r_{1:t-2}} \int ds_{0:t-1} p(y_{1:t}, s_{0:t}, r_{1:t})$$

We assume that y is always observed, hence we use the term potential to indicate the fact that $p(y_{1:t}, s_t, r_t, r_{t-1})$ is not normalized. The filtering potential is in general a conditional Gaussian mixture, i.e. a mixture of Gaussians for each configuration of $r_{t-1:t}$. We will highlight this data structure by using the following notation

$$\alpha_t \equiv \left\{ \begin{array}{cc} \alpha_t^{1,1} & \alpha_t^{1,2} \\ \alpha_t^{2,1} & \alpha_t^{2,2} \end{array} \right\}$$

where each $\alpha_t^{i,j} = p(y_{1:t}, s_t, r_t = i, r_{t-1} = j)$ for $i, j = 1 \ldots 2$ are also Gaussian mixture potentials. We will denote the conditional normalization constants as

$$Z_t^i \equiv p(y_{1:t}, r_t = i) = \sum_{r_{t-1}} \int ds_t \alpha_t^{i, r_{t-1}}$$

Consequently the evidence is given by

$$Z_t \equiv p(y_{1:t}) = \sum_{r_t} \sum_{r_{t-1}} \int ds_t \alpha_t = \sum_i Z_t^i$$

We also define the predictive density

$$\begin{aligned} \alpha_{t|t-1} &\equiv p(y_{1:t-1}, s_t, r_t, r_{t-1}) \\ &= \sum_{r_{t-2}} \int ds_{t-1}\, p(s_t|s_{t-1}, r_t, r_{t-1}) p(r_t|r_{t-1}) \alpha_{t-1} \end{aligned}$$

In general, for switching Kalman filters, calculating exact posterior features, such as the evidence $Z_t = p(y_{1:t})$, is not tractable. This is a consequence of the fact that the number of

[5]This improvement is suggested by one of the anonymous reviewers

mixture components to required to represent the exact filtering density α_t grows exponentially with time step k (i.e. one Gaussian for each of the exponentially many configurations $r_{1:t}$). Luckily, for the model we are considering here, the growth is polynomial in k only. See also [49].

To see this, suppose we have the filtering density available at time $t-1$ as α_{t-1}. The transition models can be organized also in a table where i'th row and j'th column correspond to $p(s_t|s_{t-1}, r_t = i, r_{t-1} = j)$

$$p(s_t|s_{t-1}, r_t, r_{t-1}) = \left\{ \begin{array}{cc} f_1(s_t|s_{t-1}) & \pi(s_t) \\ f_2(s_t|s_{t-1}) & f_2(s_t|s_{t-1}) \end{array} \right\}$$

Calculation of the predictive potential is straightforward. First, summation over r_{t-2} yields

$$\sum_{r_{t-2}} \alpha_{t-1} = \left\{ \begin{array}{c} \alpha^{1,1}_{t-1} + \alpha^{1,2}_{t-1} \\ \alpha^{2,1}_{t-1} + \alpha^{2,2}_{t-1} \end{array} \right\} \equiv \left\{ \begin{array}{c} \xi^1_{t-1} \\ \xi^2_{t-1} \end{array} \right\}$$

Integration over s_{t-1} and multiplication by $p(r_t|r_{t-1})$ yields the predictive potential

$$\alpha_{t|t-1} = \left\{ \begin{array}{cc} p_{1,1}\psi^1_1(s_t) & p_{1,2}Z^2_{t-1}\pi(s_t) \\ p_{2,1}\psi^1_2(s_t) & p_{2,2}\psi^2_2(s_t) \end{array} \right\}$$

where we define

$$\begin{aligned} Z^2_{t-1} &\equiv \int ds_{t-1} \xi^2_{t-1} \\ \psi^j_i(s_t) &\equiv \int ds_{t-1} f_i(s_t|s_{t-1}) \xi^j_{t-1} \end{aligned}$$

The potentials ψ^j_i can be computed by applying the standard Kalman prediction equations to each component of ξ^j_{t-1}. The updated potential is given by $\alpha_t = p(y_t|s_t)\alpha_{t|t-1}$. This quantity can be computed by applying standard Kalman update equations to each component of $\alpha_{t|t-1}$.

From the above derivation, it is clear that $\alpha^{1,2}_t$ has only a single Gaussian component. This has the consequence that the number of Gaussian components in $\alpha^{1,1}_t$ increases only linearly (the first row-sum terms ξ^1_{t-1} propagated through f_1). The second row sum term ξ^2_t is more costly; it increases at every time slice by the number of components in ξ^1_{t-1}. Since the size of ξ^1_{t-1} grows linearly, the size of ξ^2_t grows quadratically with time t.

B. Computation of MAP configuration $r^*_{1:T}$

The MAP state is defined as

$$\begin{aligned} r^*_{1:T} &= \operatorname*{argmax}_{r_{1:T}} \int ds_{0:T} p(y_{1:T}, s_{0:T}, r_{1:T}) \\ &\equiv \operatorname*{argmax}_{r_{1:T}} \int ds_{0:T} \phi(s_{0:T}, r_{1:T}) \end{aligned}$$

For finding the MAP state, we replace summations over r_t by maximization. One potential technical difficulty is that, unlike in the case for evidence calculation, maximization and integration do not commute. Consider a conditional Gaussian potential

$$\phi(s, r) \equiv \{\phi(s, r = 1), \phi(s, r = 2)\}$$

where $\phi(s, r)$ are Gaussian potentials for each configuration of r. We can compute the MAP configuration

$$r^* = \operatorname*{argmax}_r \int ds\, \phi(s, r) = \operatorname{argmax} \{Z^1, Z^2\}$$

where $Z^j = \int ds\, \phi(s, r = j)$. We evaluate the normalization of each component (i.e. integrate over the continuous hidden variable s first) and finally find the maximum of all normalization constants.

However, direct calculation of $r^*_{1:T}$ is not feasible because of exponential explosion in the number of distinct configurations. Fortunately, for our model, we can introduce a deterministic pruning schema that reduces the number of kernels to a polynomial order and meanwhile guarantees that we will never eliminate the MAP configuration. This exact pruning method hinges on the factorization of the posterior for the assignment of variables $r_t = 1, r_{t-1} = 2$ (mute to sound transition) that breaks the direct link between s_t and s_{t-1}:

$$\begin{aligned} &\phi(s_{0:T}, r_{1:t-2}, r_{t-1} = 2, r_t = 1, r_{t+1:T}) = \\ &\phi(s_{0:t-1}, r_{1:t-2}, r_{t-1} = 2)\phi(s_{t:T}, r_{t+1:T}, r_t = 1|r_{t-1} = 2) \end{aligned} \tag{15}$$

In this case:

$$\begin{aligned} &\max_{r_{1:T}} \int ds_{0:T}\, \phi(s_{0:T}, r_{1:t-2}, r_{t-1} = 2, r_t = 1, r_{t+1:T}) \\ = &\max_{r_{1:t-1}} \int ds_{0:t-1}\, \phi(s_{0:t-1}, r_{1:t-2}, r_{t-1} = 2) \\ &\times \max_{r_{t:T}} \int ds_{t:T}\, \phi(s_{t:T}, r_{t+1:T}, r_t = 1|r_{t-1} = 2) \\ = &\; Z^2_t \times \max_{r_{t+1:T}} \int ds_{t:T}\, \phi(s_{t:T}, r_{t+1:T}, r_t = 1|r_{t-1} = 2) \end{aligned} \tag{16}$$

This Equation shows that whenever we have an onset, we can calculate the maximum over the past and future configurations separately. Put differently, provided that the MAP configuration has the form $r^*_{1:T} = [r^*_{1:t-3}, r_{t-1} = 2, r_t = 1, r^*_{t+1:T}]$, the prefix $[r^*_{1:t-3}, r_{t-1} = 2]$ will be the solution for the reduced maximization problem $\arg\max_{r_{1:t-1}} \int ds_{0:t-1}\, \phi(s_{0:t-1}, r_{1:t-1})$.

1) Forward pass: Suppose we have a collection of Gaussian potentials

$$\delta_{t-1} \equiv \left\{ \begin{array}{cc} \delta^{1,1}_{t-1} & \delta^{1,2}_{t-1} \\ \delta^{2,1}_{t-1} & \delta^{2,2}_{t-1} \end{array} \right\} \equiv \left\{ \begin{array}{c} \delta^1_{t-1} \\ \delta^2_{t-1} \end{array} \right\}$$

with the property that the Gaussian kernel corresponding the prefix $r^*_{1:t-1}$ of the MAP state is a member of δ_{t-1}, i.e. $\phi(s_{t-1}, r^*_{1:t-1}) \in \delta_{t-1}$ s.t. $r^*_{1:T} = [r^*_{1:t-1}, r^*_{t:T}]$. We also define the subsets

$$\begin{aligned} \delta^{i,j}_{t-1} &= \{\phi(s_{t-1}, r_{1:t-1}) : \phi \in \delta_{t-1} \text{and } r_{t-1} = i, r_{t-2} = j\} \\ \delta^i_{t-1} &= \bigcup_j \delta^{i,j}_{t-1} \end{aligned}$$

We show how we find δ_t. The prediction is given by

$$\delta_{t|t-1} = \int ds_{t-1}\, p(s_t|s_{t-1}, r_t, r_{t-1}) p(r_t|r_{t-1}) \delta_{t-1}$$

The multiplication by $p(r_t|r_{t-1})$ and integration over s_{t-1} yields the predictive potential $\delta_{t|t-1}$

$$\left\{ \begin{array}{cc} p_{1,1} \int ds_{t-1}\, f_1(s_t|s_{t-1}) \delta^1_{t-1} & p_{1,2}\pi(s_t) \int ds_{t-1}\, \delta^2_{t-1} \\ p_{2,1} \int ds_{t-1}\, f_2(s_t|s_{t-1}) \delta^1_{t-1} & p_{2,2} \int ds_{t-1}\, f_2(s_t|s_{t-1}) \delta^2_{t-1} \end{array} \right\}$$

By the (16), we can replace the collection of numbers $\int ds_{t-1}\ \delta_{t-1}^2$ with with the scalar $Z_{t-1}^2 \equiv \max \int ds_{t-1}\ \delta_{t-1}^2$ without changing the optimum solution:

$$\delta_{t|t-1}^{1,2} \quad = \quad p_{1,2} Z_{t-1}^2 \pi(s_t)$$

The updated potential is given by $\delta_t = p(y_t|s_t)\delta_{t|t-1}$. The analysis of the number of kernels proceeds as in the previous section.

2) Decoding: During the forward pass, we tag each Gaussian component of δ_t with its past history of $r_{1:t}$. The MAP state can be found by a simple search in the collection of polynomially many numbers and reporting the associated tag:

$$r_{1:T}^* \quad = \quad \underset{r_{1:T}}{\operatorname{argmax}} \int ds_T\ \delta_T$$

We finally conclude that the forward filtering and MAP (Viterbi path) estimation algorithms are essentially identical with summation replaced by maximization and an additional tagging required for decoding.

C. Inference for monophonic pitch tracking

In this section we derive an exact message propagation algorithm for monophonic pitch tracking. Perhaps surprisingly, inference in this case turns out to be still tractable. Even though the size of the configuration space $r_{1:M,1:T}$ is of size $(M+1)^T = O(2^{T \log M})$, the space complexity of an exact algorithm remains quadratic in t. First, we define a "mega" indicator node $z_t = (j_t, r_t)$ where $j_t \in 1 \dots M$ indicates the index of the active sound generator and $r_t \in \{\text{sound}, \text{mute}\}$ indicates its state. The transition model $p(z_t|z_{t-1})$ is a large sparse transition table with probabilities

$$\left(\begin{array}{ccc|ccc} p_{1,1} & & & p_{1,2}/M & \dots & p_{1,2}/M \\ & \ddots & & \vdots & \ddots & \vdots \\ & & p_{1,1} & p_{1,2}/M & \dots & p_{1,2}/M \\ \hline p_{2,1} & & & p_{2,2} & & \\ & \ddots & & & \ddots & \\ & & p_{2,1} & & & p_{2,2} \end{array}\right) \tag{17}$$

where the transitions $p(z_t = (j,r)|z_{t-1} = (j',r'))$ are organized at the n'th row and m'th column where $n = r \times M + j - 1$ and $m = r' \times M + j' - 1$. (17). The transition models $p(s_t|s_{t-1}, z_t = (j,r), z_{t-1} = (j',r'))$ can be organized similarly:

$$\left(\begin{array}{ccc|ccc} f_{1,1} & & & \pi(s_t) & \dots & \pi(s_t) \\ & \ddots & & \vdots & \ddots & \vdots \\ & & f_{1,M} & \pi(s_t) & \dots & \pi(s_t) \\ \hline f_{2,1} & & & f_{2,1} & & \\ & \ddots & & & \ddots & \\ & & f_{2,M} & & & f_{2,M} \end{array}\right)$$

Here, $f_{r,j} \equiv f_{r,j}(s_t|s_{t-1})$ denotes the transition model of the j'th sound generator when in state r. The derivation for filtering follows the same lines as the onset/offset detection model, with only slightly more tedious indexing. Suppose we have the filtering density available at time $t-1$ as α_{t-1}. We first calculate the predictive potential. Summation over z_{t-2} yields the row sums

$$\xi_{t-1}^{(r,j)} \quad = \quad \sum_{r',j'} \alpha_{t-1}^{(r,j),(r',j')}$$

Integration over s_{t-1} and multiplication by $p(z_t|z_{t-1})$ yields the predictive potential $\alpha_{t|t-1}$. The components are given as $\alpha_{t|t-1}^{(r,j)(r',j')} =$

$$\left\{\begin{array}{ll} (1/M) p_{r,r'} \pi(s_t) Z_{t-1}^{(r',j')} & r = 1 \wedge r' = 2 \\ [j = j'] \times p_{r,r'} \psi_t^{(r,j)(r',j')} & \text{otherwise} \end{array}\right. \tag{18}$$

where we define

$$\begin{aligned} Z_{t-1}^{(r',j')} \quad &\equiv \quad \int ds_{t-1}\ \xi_{t-1}^{(r',j')} \\ \psi_t^{(r,j)(r',j')} \quad &\equiv \quad \int ds_{t-1}\ f_{r,j}(s_t|s_{t-1}) \xi_{t-1}^{(r',j')} \end{aligned}$$

The potentials ψ can be computed by applying the standard Kalman prediction equations to each component of ξ. Note that the forward messages have the same sparsity structure as the prior, i.e. $\alpha_{t-1}^{(r,j)(r',j')} \neq 0$ when $p(r_t = r, j_t = j | r_{t-1} = r', j_t = j')$ is nonzero. The updated potential is given by $\alpha_t = p(y_t|s_t)\alpha_{t|t-1}$. This quantity can be computed by applying standard Kalman update equations to each nonzero component of $\alpha_{t|t-1}$.

D. Monophonic pitch tracking with varying fundamental frequency

We model pitch drift by a sequence of transition models. We choose a grid such that $\omega_j/\omega_{j+1} = \mathcal{Q}$, where $\mathcal{Q}$ is close to one. Unfortunately, the subdiagonal terms introduced to the prior transition matrix $p(z_t = (1, j_t)|z_{t-1} = (1, j_{t-1}))$

$$p_{1,1} \times \left(\begin{array}{cccccc} (d_0 + d_1) & d_{-1} & & & & \\ d_1 & d_0 & d_{-1} & & & \\ & d_1 & \ddots & \ddots & & \\ & & \ddots & d_0 & d_{-1} & \\ & & & d_1 & (d_0 + d_{-1}) \end{array}\right) \tag{19}$$

render an exact algorithm exponential in t. The recursive update equations, starting with α_{t-1}, are obtained by summing over z_{t-2}, integration over s_{t-1} and multiplication by $p(z_t|z_{t-1})$. The only difference is that the prediction equation (18) needs to be changed to $\alpha_{t|t-1}^{(r,j)(r',j')} =$

$$\left\{\begin{array}{ll} d(j - j') \times p_{r,r'} \psi_t^{(r,j)(r',j')} & r = 1 \wedge r' = 1 \\ (1/M) p_{r,r'} \pi(s_t) Z_{t-1}^{(r',j')} & r = 1 \wedge r' = 2 \\ [j = j'] \times p_{r,r'} \psi_t^{(r,j)(r',j')} & r = 2 \end{array}\right.$$

where ψ and Z are defined in (19). The reason for the exponential growth is the following: Remember that each $\psi^{(r,j)(r',j')}$ has as many components as an entire row sum of $\xi_{t-1}^{(r,j)} = \sum_{r',j'} \alpha_{t-1}^{(r,j),(r',j')}$. Unlike the inference for piecewise constant pitch estimation, now at some rows there are two or more messages (e.g. $\alpha_{t|t-1}^{(1,j)(1,j)}$ and $\alpha_{t|t-1}^{(1,j)(1,j+1)}$) that depend on ψ.

APPENDIX II
COMPUTATIONAL SIMPLIFICATIONS

A. Pruning

Exponential growth in message size renders an algorithm useless in practice. Even in special cases, where the message size increases only polynomially in T, this growth is still prohibitive for many applications. A cheaper approximate algorithm can be obtained by pruning the messages. To keep the size of messages bounded, we limit the number of components to N and store only components with the highest evidence. An alternative is discarding components of a message that contribute less than a given fraction (e.g. 0.0001) to the total evidence. More sophisticated pruning methods with profound theoretical justification, such as resampling [23] or collapsation [50], are viable alternatives but these are computationally more expensive. In our simulations, we observe that using a simple pruning method with the maximum number of components per message set to $N = 100$, we can obtain results very close to an exact algorithm.

B. Kalman filtering in a reduced dimension

Kalman filtering with a large state dimension $|s|$ at typical audio sampling rates $F_s \approx 40$ kHz may be prohibitive with generic hardware. This problem becomes more severe when the number of notes M is large, (which is typically around $50 - 60$), than even conditioned on a particular configuration $r_{1:M}$, the calculation of the filtering density is expensive. Hence, in an implementation, tricks of precomputing the covariance matrices can be considered [48] to further reduce the computational burden.

Another important simplification is less obvious from the graphical structure and is a consequence of the inherent asymmetry between the sound and mute states. Typically, when a note switches and stays for a short period in the mute state, i.e. $r_{j,t} =$ mute for some period, the marginal posterior over the state vector $s_{j,t}$ will converge quickly to a zero mean Gaussian with a small covariance matrix *regardless* of observations y. We exploit this property to save computations by clamping the hidden states for sequences of $s_{j,t:t'}$ to zero for $r_{j,t:t'} =$ "mute". This reduces the hidden state dimension, since typically, only a few sound generators will be in sound state.

REFERENCES

[1] A. Bregman, *Auditory Scene Analysis*. MIT Press, 1990.

[2] G. J. Brown and M. Cooke, "Computational auditory scene analysis," *Computer Speech and Language*, vol. 8, no. 2, pp. 297–336, 1994.

[3] M. Weintraub, "A theory and computational model of auditory monaural sound separation," Ph.D. dissertation, Stanford University Dept. of Electrical Engineering, 1985.

[4] S. Roweis, "One microphone source separation," in *Neural Information Processing Systems, NIPS*2000*, 2001.

[5] D. P. W. Ellis, "Prediction-driven computational auditory scene analysis," Ph.D. dissertation, MIT, Dept. of Electrical Engineering and Computer Science, Cambridge MA, 1996.

[6] E. D. Scheirer, "Music-listening systems," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[7] G. Tzanetakis, "Manipulation, analysis and retrieval systems for audio signals," Ph.D. dissertation, Princeton University, 2002.

[8] R. Rowe, *Machine Musichanship*. MIT Press, 2001.

[9] M. Plumbley, S. Abdallah, J. P. Bello, M. Davies, G. Monti, and M. Sandler, "Automatic music transcription and audio source separation," *Cybernetics and Systems*, vol. 33, no. 6, pp. 603–627, 2002.

[10] W. J. Hess, *Pitch Determination of Speech Signal*. New York: Springer, 1983.

[11] B. G. Quinn and E. J. Hannan, *The Estimation and Tracking of Frequency*. Cambridge University Press, 2001.

[12] R. A. Irizarry, "Local harmonic estimation in musical sound signals," *Journal of the American Statistical Association, to appear*, 2001.

[13] ——, "Weighted estimation of harmonic components in a musical sound signal," *Journal of Time Series Analysis*, vol. 23, 2002.

[14] K. L. Saul, D. D. Lee, C. L. Isbell, and Y. LeCun, "Real time voice processing with audiovisual feedback: toward autonomous agents with perfect pitch," in *Neural Information Processing Systems, NIPS*2002*, Vancouver, 2002.

[15] B. Truong-Van, "A new approach to frequency analysis with amplified harmonics," *J. Royal Statistics Society B*, no. 52, pp. 203–222, 1990.

[16] L. Parra and U. Jain, "Approximate Kalman filtering for the harmonic plus noise model," in *Proc. of IEEE WASPAA*, New Paltz, 2001.

[17] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka, "Application of bayesian probability network to music scene analysis," in *Proc. IJCAI Workshop on CASA*, Montreal, 1995, pp. 52–59.

[18] A. Sterian, "Model-based segmentation of time-frequency images for musical transcription," Ph.D. dissertation, University of Michigan, Ann Arbor, 1999.

[19] P. J. Walmsley, "Signal separation of musical instruments," Ph.D. dissertation, University of Cambridge, 2000.

[20] M. Davy and S. J. Godsill, "Bayesian harmonic models for musical signal analysis," in *Bayesian Statistics 7*, 2003.

[21] C. Raphael, "A mixed graphical model for rhythmic parsing," in *Proc. of 17th Conf. on Uncertainty in Artif. Int.* Morgan Kaufmann, 2001.

[22] D. Temperley, *The Cognition of Basic Musical Structures*. MIT Press, 2001.

[23] A. T. Cemgil and H. J. Kappen, "Monte Carlo methods for tempo tracking and rhythm quantization," *Journal of Artificial Intelligence Research*, vol. 18, pp. 45–81, 2003.

[24] K. Martin, "Sound-source recognition," Ph.D. dissertation, MIT, 1999.

[25] A. Klapuri, T. Virtanen, and J.-M. Holm, "Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals," in *COST-G6, Conference on Digital Audio Effects*, 2000.

[26] A. T. Cemgil, H. J. Kappen, and D. Barber, "Generative model based polyphonic music transcription," in *Proc. of IEEE WASPAA*. New Paltz, NY: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, October 2003.

[27] N. H. Fletcher and T. Rossing, *The Physics of Musical Instruments*. Springer, 1998.

[28] X. Serra and J. O. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1991.

[29] X. Rodet, "Musical sound signals analysis/synthesis: Sinusoidal + residual and elementary waveform models," *Applied Signal Processing*, 1998.

[30] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[31] A. C. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge, U.K.: Cambridge Univ. Press, 1989.

[32] M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models*, 2nd ed. Springer-Verlag, 1997.

[33] V. Valimaki, J. Huopaniemi, Karjaleinen, and Z. Janosy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *J. Audio Eng. Society*, vol. 44, no. 5, pp. 331–353, 1996.

[34] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[35] K. P. Murphy, "Switching Kalman filters," Dept. of Computer Science, University of California, Berkeley, Tech. Rep., 1998.

[36] ——, "Dynamic Bayesian networks: Representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, 2002.

[37] Z. Ghahramani and M. Beal, "Propagation algorithms for variational Bayesian learning," in *Neural Information Processing Systems 13*, 2000.

[38] Z. Ghahramani and G. E. Hinton, "Parameter estimation for linear dynamical systems. (crg-tr-96-2)," University of Totronto. Dept. of Computer Science., Tech. Rep., 1996.

[39] C. Raphael, "Automatic transcription of piano music," in *Proc. ISMIR*, 2002.

[40] E. Terhardt, "Pitch, consonance and harmony," *Journal of the Acoustical Society of America*, vol. 55, no. 5, pp. 1061–1069, 1974.

[41] Y. Qi, T. P. Minka, and R. W. Picard, "Bayesian spectrum estimation of unevenly sampled nonstationary data," MIT Media Lab, Tech. Rep. Vismod-TR-556, 2002.
[42] H. Attias, "Independent factor analysis," *Neural Computation*, vol. 11, no. 4, pp. 803–851, 1999.
[43] B. Olshausen and J. Millman, "Learning sparse codes with a mixture-of-Gaussians prior," in *NIPS*, vol. 12. MIT Press, 2000, pp. 841–847.
[44] C. Raphael and J. Stoddard, "Harmonic analysis with probabilistic graphical models," in *Proc. ISMIR*, 2003.
[45] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, no. 4, pp. 74–87, 1992.
[46] A. Doucet, N. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
[47] T. Minka, "Expectation propagation for approximate Bayesian inference," Ph.D. dissertation, MIT, 2001.
[48] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques and Software*. Boston: Artech House, 1993.
[49] P. Fearnhead, "Exact and efficient bayesian inference for multiple changepoint problems," Dept. of Math. and Stat., Lancaster University, Tech. Rep., 2003.
[50] O. Heskes, T. Zoeter, "Expectation propagation for approximate inference in dynamic Bayesian networks," in *Proceedings UAI*, 2002.