Piano Transcription in the Studio Using an Extensible Alternating Directions Framework

Sebastian Ewert, *Member, IEEE,* and Mark Sandler, *Fellow, IEEE*

***Abstract*—Given a musical audio recording, the goal of automatic music transcription is to determine a score-like representation of the piece underlying the recording. Despite significant interest within the research community, several studies have reported on a "glass ceiling" effect, an apparent limit on the transcription accuracy that current methods seem incapable of overcoming. In this paper, we explore how much this effect can be mitigated by focusing on a specific instrument class and making use of additional information on the recording conditions available in studio or home recording scenarios. In particular, exploiting the availability of single note recordings for the instrument in use we develop a novel signal model employing variable-length spectro-temporal patterns as its central building blocks – tailored for pitched percussive instruments such as the piano. Temporal dependencies between spectral templates are modeled, resembling characteristics of factorial scaled hidden Markov models (FS-HMM) and other methods combining Non-Negative Matrix Factorization with Markov processes. In contrast to FS-HMMs, our parameter estimation is developed in a global, relaxed form within the extensible alternating direction method of multipliers (ADMM) framework, which enables the systematic combination of basic regularizers propagating sparsity and local stationarity in note activity with more complex regularizers imposing temporal semantics. The proposed method achieves an f-measure of 93-95% for note onsets on pieces recorded on a Yamaha Disklavier (MAPS DB).**

***Index Terms*—Music Transcription, Alternating Direction Method of Multipliers, Markov-Regularizer, Non-Negative Matrix Factorization, Structured Sparsity.**

I. Introduction

AUTOMATIC Music Transcription (AMT) has a long history in music processing [1]. Identifying higher-level musical concepts such as notes in digital music recordings, it is often considered a key technology for a semantic analysis of music, with applications ranging from various retrieval tasks in music informatics over computational musicology and performance analysis to creative music technology [2]. Despite significant interest within the research community, transcription remains highly challenging and accuracies reported in recent years seem to have reached a plateau [3]. A promising approach to achieve higher accuracy is to provide a method with additional prior information. For example, the user can be actively involved in the transcription process providing partial transcription results to stabilize the parameter estimation process [4]. While this is an interesting direction and leads to measurable improvements in accuracy, it can be time-consuming and precludes a fully automatic process.

A central goal of this paper is to investigate if the performance of a transcription system can be improved by focusing on a specific application scenario. First, we focus on a single class of instruments, namely pitched percussive instruments such as the piano or harpsichord. Second, we assume that recordings of single notes are available for the instrument to be transcribed (at least one playing style), which mitigates many uncertainties regarding the instrument and the recording conditions (room response, recording equipment). Further, to obtain a practical transcription system, we assume that the user can play at the beginning of a recording session a note in pianissimo (low intensity), which is used by our system to derive a threshold employed to differentiate between an active note and estimation noise. Given this scenario, we can tailor our proposed signal model to precisely this instrument class, which is necessary to account for the highly non-stationary behavior of the piano sound production process. In particular, when a key on the piano is hit a mechanical, pitch-dependent, broad-band sound is produced, followed by a harmonic sound that is non-stationary due to amplitude modulation (*beating*) and differences in decay rate between the harmonics [5]. The main idea in our signal model is that this sound sequence is highly characteristic for a note event, which enables us to use the single note recordings as a blueprint to identify similar spectro-temporal patterns in the recording.

To implement this idea, existing methods are not well suited. In particular, most state of the art transcription methods employ variants of *non-negative matrix factorization (NMF)*, which treat spectral and temporal information independently and therefore cannot make use of such joint patterns (see also Section II). As an extension to NMF, *Non-Negative Matrix Deconvolution (NMD)* employs spectro-temporal patterns – however, these have a fixed length and thus are not a good match for modeling notes of variable duration. A promising candidate could be the *factorial scaled hidden Markov model (FS-HMM)* [6] and its variants [7], [8], in which a Markov process governs which spectral templates can be used in a given time frame based on the previous frame. Such models, however, were proposed in the context of modeling a few concurrent speakers, and, in a standard form, their computational complexity is exponential in the number of Markov processes [7], of which we have 88 in our case (one for each piano key). Parameter decoupling techniques such as generalized expectation maximization often used in this context (i.e. fixing some parameters while optimizing against the remaining ones) tend to converge extremely slowly with so many independent processes, and often lead to poor local minima with excessive decoupling (see also [7], [9], [10]).

The method presented in this paper employs spectro-temporal patterns of variable length to model a given time-frequency representation of a piano recording. The core idea for our

S. Ewert and M. Sandler are with Queen Mary University of London, UK.
Manuscript received Nov 2015; revised Apr 2016.

parameter estimation stems from the observation that piano sounds can be well represented using simple left-to-right Markov models (i.e. there is a clear succession of spectral templates when a piano key is hit). As we will see, this enables us, instead of strictly enforcing Markov properties as in FS-HMMs, to approximate the temporal transitions between spectral templates in a relaxed form by stating the parameter estimation problem as a structured sparse coding problem, which is controlled by simple convex regularizers. Using these regularizers we can steer the solution close to a semantically meaningful progression similar to an FS-HMM solution. The resulting problem is convex and all parameters are jointly optimized (i.e. no decoupling is necessary), such that poor local minima are typically avoided. Once we are close to a solution to this convex problem, we switch to non-convex regularizers, which can be interpreted as projections onto matrices encoding strict Markov-like transitions. While these non-convex regularizers would lead to relatively poor results without the convex initialization, we can use this combination to further refine the parameter estimate, gradually enforcing stricter, more meaningful transitions between spectral templates. The entire model is developed within the highly extensible *alternating direction method of multipliers (ADMM)* framework, which is widely used in the sparse-coding and computer vision communities but has not yet received the same amount of attention in audio processing research despite its proven usefulness in non-linear optimization of non-differentiable functions and machine learning for big data.

The remainder is organized as follows. In Section II we discuss related work, focusing on core concepts in current transcription methods. In Section III, we describe the proposed model and explain the effects of specific regularizers. Next, in Section IV, we develop an efficient parameter estimation method for our model based on the ADMM algorithmic framework, which we describe in more detail as we think it might be useful in other contexts as well. In Section VI, we discuss the results of various experiments to illustrate the performance of the proposed model as well as the influence of parameters. Finally, in Section VII we conclude the paper with an outlook on future work.

II. Related Work

As one of the central topics in music processing, automatic music transcription (AMT) has attracted considerable interest within the research community over the years. In the following, we refer to overview articles for a more comprehensive overview and focus on discussing central contributions and general concepts. In particular, many methods proposed before 2005 are described in [2], and many more recent methods in an outlook article [3].

Overall, a wide range of strategies have been used for AMT. For example, in [11], [12] the recording to be transcribed is first segmented by detecting onsets and rhythmic structures, which is then used to guide a subsequent pitch estimation. A large body of work involves elaborate methods exploiting the harmonicity of musical sounds to group detected spectral peaks into note objects [13]–[16]. Another group of methods employs probabilistic sinusoidal plus noise modeling, in which parameters such as onset and offset position, fundamental frequency, intensity and spectral envelope parameters are adjusted to match the observed signal using maximum a posteriori estimation [17] or genetic algorithms [18]. Another approach employs a hidden Markov model (HMM) [19], where each state corresponds to one possible combination of active notes, which requires elaborate heuristic state-space pruning strategies to be computationally feasible.

A further successful technique is based on the iterative estimation and subtraction of the predominant fundamental frequency and its corresponding harmonics [20]. Transcription has also been considered as a classification or regression task in the context of discriminative methods. In [21], a total of 87 support vector machines were trained to detect pitch activity in spectrogram frames, followed by temporal smoothing using an HMM. As another example, the system presented in [22] was highly successful in comparative studies [23] and uses time-delay neural networks (similar to convolutional networks in time direction) to classify the output of adaptive oscillators, which are used to track and group partials in the output of a gammatone filterbank. More recently, [24] presented a system using a multi-layer recurrent neural network based on Hochreiter and Schmidhuber's bidirectional long short term memory units, which were chosen to better model temporal dependencies in music.

Most state of the art methods, however, employ variants of non-negative matrix factorization (NMF), see [25] for a recent overview. In general, the underlying idea of NMF-based methods is to model a given time-frequency representation of a recording as a mixture of note- or sound-specific spectral template vectors and to estimate their individual activity over time. One principal advantage of NMF over many early approaches is that parameters in NMF are iteratively refined and errors made early in the process can thus be corrected later. Further, many NMF-based methods contain parameters with a clear interpretation, which can facilitate the integration of prior knowledge and thus can give an advantage over some discriminative models where this can be more difficult [26]. For example, many recent approaches have extended classic NMF [27] by forcing the spectral templates to represent only harmonic sounds. In particular, the methods presented in [23], [28] restrict the spectral templates to a linear combination of narrow-band harmonic sub-templates, each having a clear pitch association. As another example, the PreFest [29] and HTC [30] methods can be interpreted as modeling spectral templates using a set of scaled Gaussians, each representing a partial of a harmonic sound in frequency direction. To transcribe recordings of several instruments, a typical approach is to use pre-trained spectral templates for specific instruments [31], [32]. Further, a high number of spectral templates can be used to increase the representation accuracy of the model, which leads to sparse coding methods [33] where the model is encouraged to explain the audio using only a few of the available templates, see also [34] for a recent study in an AMT context.

An overarching principle of NMF-based methods is that spectral properties are decoupled from temporal ones, i.e. neither do the activations provide information about how a

note spectrally manifests nor do the templates describe when a note occurs or how it evolves. Without an enforced temporal progression of spectral templates, however, non-stationary signals like a piano note are difficult to model. For example, one typically cannot express in NMF that a certain spectral template for the sustain part of a note is expected after a certain time after the attack. Further, activations between neighboring frames are often not or only loosely coupled, such that it is difficult to express that an activation value has a certain relationship to the ones in subsequent frames.

An extension to NMF modeling such spectro-temporal dependencies was presented in [35] under the name *Non-Negative Matrix Deconvolution (NMD)*, which uses, instead of spectral templates, entire time-frequency patterns concatenating several templates over time as building blocks within the model. The use of patterns enforces a specific temporal order for the templates and effectively couples their activations. However, since these patterns have a fixed length, NMD has not been used to transcribe instruments such as the piano, where notes are of variable duration. The FS-HMM approach [6]–[8] adds more flexibility regarding the length of template sequences by employing a Markov process that governs the use of specific templates in a given frame. While such models indeed provide the necessary freedom to model a non-stationary sound such as a piano note, their computational complexity in a basic formulation is typically exponential in the number of Markov processes [7] – with 88 piano keys and corresponding Markov processes this is computationally infeasible. Updating the parameters of some Markov processes while keeping the remaining ones fixed (i.e. applying generalized expectation maximization schemes) typically converges extremely slowly with so many processes and often leads to poor local minima in the distance function between model and observed signal [9] (see [36] in a transcription scenario). A hybrid between NMD and FS-HMM for piano transcription was presented in [10], where the computational issues are approached by a combination of decoupling strategies similar to Viterbi training to generate note event candidates and global, coupled optimization over all candidates during an activation update. While this type of parameter estimation typically yields transcription results of comparatively high quality, the remaining decoupling during the candidate selection can still lead to some poor local minima, which limits the transcription performance, as we will see below.

III. Proposed Model

While the goal of our proposed method is similar to the approach presented in [10], the underlying model and parameter estimation process are fundamentally different. The design goals were to eliminate the decoupling of parameters, and as we will see, the resulting model is not only simpler but also yields improved results and is computationally more efficient. In the following, we assume that for each of the $K = 88$ piano keys a recording of a single note for the instrument to be transcribed is available. Computing a log-frequency magnitude spectrogram from each recording, we obtain K time-frequency patterns, each consisting of L spectral templates, resulting in a *pattern dictionary tensor* $P \in \mathbb{R}_{\geq 0}^{M \times L \times K}$, where M is the number of frequency bins. Each column $P(:, \ell, k) \in \mathbb{R}_{\geq 0}^{M}$ for fixed ℓ and k contains a single *spectral template vector* or *template* for short; here we used the Matlab notation *:* to refer to all elements in an index dimension, i.e. $\{1, \ldots, M\}$ in this case. In contrast to most NMF approaches, we do not normalize the templates in order to preserve their energy progression over time, which later will provide an additional indication for which part of the note pattern should be active in a given time frame.

Next, given a log-frequency magnitude spectrogram $V \in \mathbb{R}_{\geq 0}^{M \times N}$ of a recording to be transcribed, we model each entry in V as a sum over the K patterns. More precisely:

$$V(m,n) \approx (PA)(m,n) := \sum_{k} \sum_{\ell} P(m,\ell,k) \cdot A(k,\ell,n), \quad (1)$$

where $A \in \mathbb{R}_{\geq 0}^{K \times L \times N}$ is the *activity tensor*, which specifies the intensity of each template in each frame. Our goal will be to design and minimize a function of A describing a semantically meaningful distance (or divergence) between V and PA. The final distance function will consist of several terms each encouraging certain behavior in A in a soft way by penalizing unwanted behavior. That means that our model does not directly impose any structure on A, which is in stark contrast to many NMF approaches discussed in Section II. For example, a temporal order of templates is not hardly enforced within the model as in more complex approaches such as FS-HMM or in [10]. This lack of enforcement will enable us to build efficient and not hardly decoupling parameter estimation procedures in Section IV.

A. Encouraging Data Fidelity and Non-Negativity

As a first step, we include a data fidelity term in our distance function in the form of a generalized divergence between V and PA. Here, most other sparse coding methods employ the Frobenius norm [33]. For NMF-based AMT methods, however, the use of other divergences led to improved results. We use the generalized Kullback-Leibler (KL) divergence

$$f_1(PA) := D(V, PA) := \sum_{m,n} d(V(m,n), (PA)(m,n))$$

with $d(a,b) := a \cdot \log\left(\frac{a}{b}\right) - a + b$ for $a, b > 0$. These improvements were often attributed to the observation that the differences between V and PA are distributed in practice rather according to a Poisson than a Gaussian distribution, which suggests the use of the KL divergence [33]. However, the KL divergence is also meaningful from an auditory point of view as the log term represents the difference in perceived loudness as approximated by Weber's law [37]:

$$\log(V(m,n)/p_m) - \log((PA)(m,n)/p_m) = \log(V(m,n)/(PA)(m,n)),$$

where p_m is a frequency dependent constant related to the threshold of hearing. Since our divergence is only defined for positive A, we add the term

$$f_2(A) := \chi_{\mathbb{R}_{\geq 0}^{K \times L \times N}}(A)$$

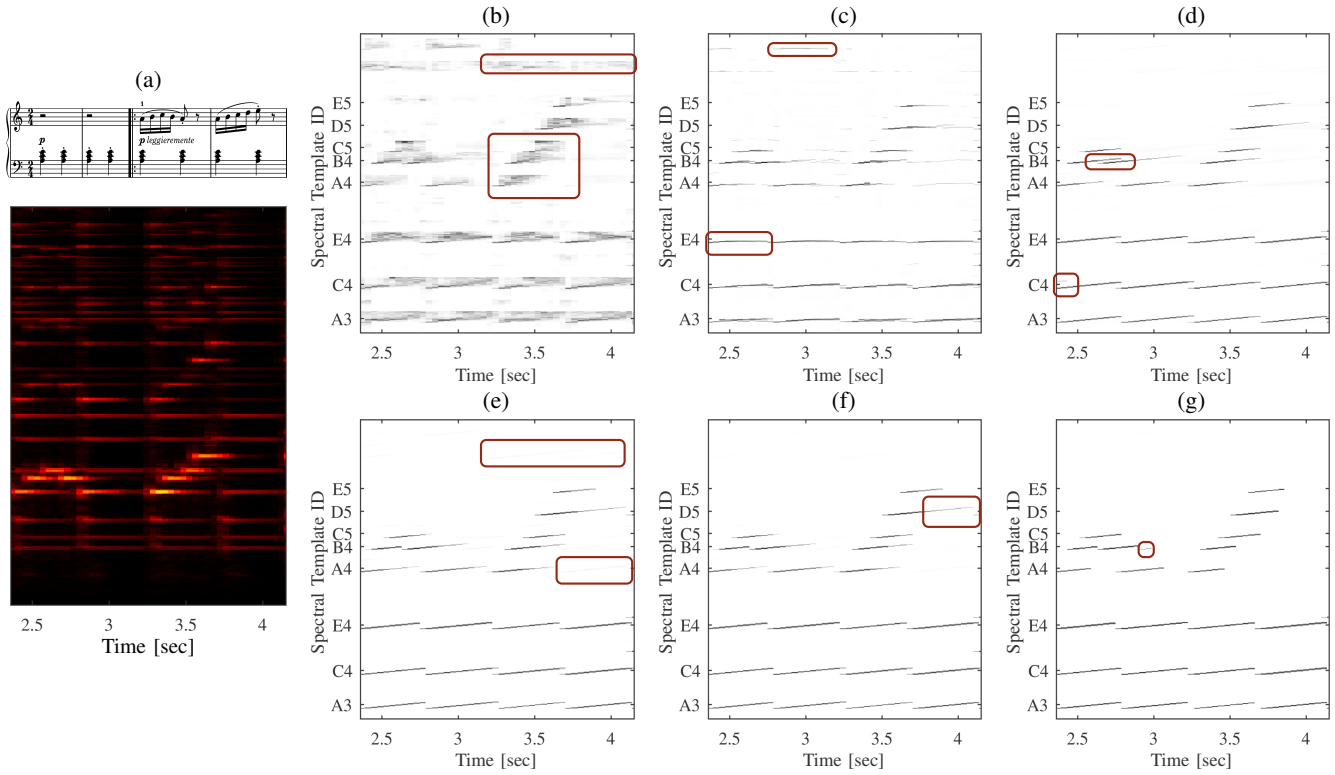

Fig. 1. **(a)** Log-frequency spectrogram of a recording of bars 3 and 4 of Burgmüller's Opus 100 Etude 2. (b)-(g) Activity tensor estimated based on various objective functions and regularizers. **Convex terms**: **(b)** Kullback-Leibler and non-negativity term, **(c)** ℓ_1 or LASSO regularizer, **(d)** total diagonal variation regularizer. **Non-convex terms**: **(e)** Markov-state regularizer, **(f)** threshold regularizer, **(g)** Binary Markov-state and strict coupling regularizer. Marked areas are discussed in the text.

to enforce non-negativity of A, where χ_S is the *characteristic function* of some set S, i.e. $\chi_S(A) = \infty$ if $A \notin S$ and zero otherwise. Thus our current objective function becomes

$$h_b(A) := f_1(PA) + f_2(A).$$

To illustrate the effects resulting from the use of various cost and regularizer terms we employ in Fig. 1 a recording of bars three and four of Burgmüller's Opus 100 Etude 2, see Fig. 1a. Subfigures b-g show several A obtained by minimizing different objective functions. For a clearer visual representation of the order-3 tensor A, we 'flattened' A by stacking the slices $A(k,:,:)$ vertically on top of each other to obtain a matrix representation of size $KL \times N$, parts of which are shown in Fig. 1b-g. For a semantically meaningful result, we expect to find for each note a diagonal line in A, with the first template for the note being activated at the onset position followed by activations for the subsequent templates in the frames after the onset. Note that Fig.1 uses different scalings for the horizontal and vertical axis such that diagonal lines in A do not have 45 degrees.

Looking at Fig.1b, we see that after minimizing h_b most activations indeed correspond to templates for the notes used in the example, compare Fig.1a. However, a semantically meaningful diagonal structure hardly exists, with activation energy for a note being spread across different templates for a note (see bottom marker in Fig.1b). Further, since we only have one spectral pattern corresponding to one specific playing style for each note in our dictionary tensor, the various patterns played in the actual recording somewhat differ from the ones in the dictionary. Therefore, some energy in V corresponding to a specific note is not 'explained' by the corresponding pattern which leads to additional spurious activations (see upper marker in Fig.1b). Overall, a clear discrimination between actual notes and estimation errors would be difficult using this A.

B. Encouraging Sparsity

To obtain fewer and more meaningful activation patterns, a typical approach is to encourage sparsity in A by adding its ℓ^1 norm to the objective [33]:

$$h_c(A) = h_b(A) + f_3(A),$$

where $f_3 := \lambda_1 \|\cdot\|_1$ and $\lambda_1 \geq 0$ is a parameter balancing the importance of the sparsity and the remaining terms. From a probabilistic point of view, the use of the ℓ^1 norm is equivalent to assuming that the activities A we will encounter in real recordings are distributed according to a Laplace distribution [33], [38]. The Laplace can be interpreted as a variant of the Normal distribution that contains an absolute instead of a squared difference from its mean, which makes higher and sharper peaks in A more likely compared to the Normal distribution. While we could follow this probabilistic interpretation, we rather choose an optimization point of view as not all of our regularizers will have a straightforward probabilistic interpretation – this will lead us to the well-known concept of *soft thresholding* in Section IV.

In Fig.1c we see the result of minimizing h_c with respect to A. The sparsifying and energy focusing effect of the ℓ^1 term are clearly visible. A semantically meaningful diagonal activation structure, however, can still not be found but rather horizontal lines (bottom marker). The underlying reason is that we did not normalize our spectral templates as in other sparse coding method. Due to the energy decay in piano sounds, the early templates in a pattern contain more energy than the subsequent ones. Therefore, we can often further minimize h_c by activating a wrong template with less activation intensity, rather than using the correct template with more intensity. As a result something counterintuitive happens: increasing λ_1 can lead to more random, spurious activations because the use of the wrong templates leads to unexplained residual energy which is then modeled using other patterns (upper marker).

C. Encouraging a Temporally Meaningful Template Order

To counter these negative effects of the ℓ_1 term, we next introduce a regularizer that enhances diagonal structures. To this end, we define for $(k,\ell,n) \in [1:K]\times[1:L-1]\times[1:N-1]$

$$\begin{aligned}\Delta_D[A](k,\ell,n) &:= A(k,\ell,n) - A(k,\ell+1,n+1),\\ f_4 &:= \lambda_2 \left\|\cdot\right\|_1,\\ h_d(A) &:= h_c(A) + f_4(\Delta_D[A])\end{aligned}$$

where $\lambda_2 \geq 0$ is another balancing parameter. The Δ_D operator is essentially a simple high-pass filter, which suppresses in combination with $\left\|\cdot\right\|_1$ oscillations and unnecessary changes along the diagonals of A. Similar to temporal continuity constraints as used in NMF [39], this approach corresponds to an anisotropic version of total variation image denoising, a well-studied problem in computer vision [40] where the goal is to remove various types of noise while preserving edges – a property useful for us to account for the energy drop at offsets. Due to this similarity, we refer to this regularizer as *total diagonal variation (TDV)*.

The results of minimizing $h_d(A)$ with respect to A are shown in Fig.1d. As we can see the TDV regularizer can be used to effectively attenuate not only semantically not meaningful horizontal and vertical activation structures in A but also single spurious activations. Further, while the use of unnormalized templates had detrimental effects on the ℓ^1 term, it increases the usefulness of the TDV regularizer drastically. In particular, as the energy decay of the piano sound is already accounted for in P, we expect only marginal change across a given diagonal in A. This way, we can set a high value for λ_2 for the TDV term which suppresses many semantically meaningless activations without leading to strong negative effects from a modeling point of view.

From a numerical point of view, it is important to note that h_d is convex in A. To see this, it is useful to recall which operations preserve convexity [41]. Applying these rules, we see that d is strictly convex in b, and thus D applying d in a sum to the entries in PA is strictly convex in PA. Since PA is linear in A, $D(V,PA)$ is convex in A. Further, χ_M is convex if M is a convex set which is the case for $\mathbb{R}_{\geq 0}^{K\times L\times N}$. Finally, since $\left\|A\right\|_1$ is convex and $\Delta_D[A]$ is linear in A, we see that all terms in h_d are convex in A. This convexity guarantees that our algorithms cannot get stuck in poor local minima of h_d, which leads to meaningful results even without a good initialization for A. On the downside, using just convex terms one is often limited to semantically less expressive regularizers. Therefore, as a general concept, we will use h_d primarily during a first algorithmic stage to robustly obtain a reasonable initialization for A. Then, we add more expressive, non-convex regularizers that would often lead to useless results on their own – in combination with this initialization, however, they can be used to further refine an already reasonable estimate of A.

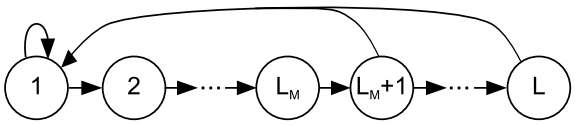

Fig. 2. Graphical model describing which transitions between templates for a specific piano key k are encouraged by the regularizer term $\chi_{\mathcal{M}}$.

D. Constraining the Concurrency of Templates

To see remaining issues, note that in Fig.1d for a single piano key k several templates can be active at the same time in parallel diagonal lines (see markers), which is semantically not meaningful and can lead to estimation errors. To mitigate such issues, we now develop a regularizer which encourages activations in A to follow transition rules for the templates described by the graphical model depicted in Fig.2, i.e. we integrate the concepts behind the FS-HMM and its variants [6]. While one of our contributions is a more robust parameter estimation process for these models (to be described in Section IV), we focus for now on the conceptual differences from a modeling point of view. In particular, instead of using a fully connected ergodic process, we employ, from a probabilistic point of view, a structured Bakis-type Markov process. Using such a process, we can express that we expect a specific sequence of templates after an onset, including a minimum duration for the entire sequence.

More precisely, for a given k, the model expresses that if we want to use the ℓ-th template in frame n, we need to use template $\ell-1$ in frame $n-1$, with an exception only for the first template. Further, to use template ℓ for $2 \leq \ell < L_M$ in a given frame we must use templates $\ell+1,\ldots,L_M$ in the subsequent frames, where L_M is referred to as the *minimum note length* in frames. To realize this model as a regularizer, we employ the characteristic function χ in combination with a specific set $\mathcal{M} \subset \mathbb{R}_{\mathbb{R}_{\geq 0}}^{K\times L\times N}$, and $A \in \mathcal{M}$ if for each k the corresponding $A(k,:,:)$ encodes, in a specific way, a valid state sequence for our graphical model. A simple encoding could be to define that state ℓ is considered active in frame n if $A(k,\ell,n) > 0$ and the remaining entries in $A(k,:,n)$ are zero. However, since our time-frequency representation V will use overlapped windows, this would be too restrictive. In particular, each part of the signal is contained in several consecutive frames and therefore we need the capability in our model to activate consecutive templates together in a given frame. Therefore, we define that $A(k,:,n)$ encodes state ℓ if $A(k,\tilde{\ell},n) \geq 0$ for all $\tilde{\ell} \in \{\ell,\ldots,\ell+\lceil w/s\rceil\} \cap \{1,\ldots,L\}$ and $A(k,\tilde{\ell},n) = 0$ for all remaining $\tilde{\ell}$, where w and s are

the window and step size in samples used to compute the spectrogram V. Due to its resemblance of Markov models, we refer to $\chi_{\mathcal{M}}(A)$ as the *Markov-State (MS) regularizer*. The results of initializing A using h_d and then refining it with

$$h_e(A) := h_d(A) + f_5(A)$$

with $f_5 := \chi_{\mathcal{M}}$ are shown in Fig.1e. As we can see, including the MS term removes the concurrent activations highlighted in Fig.1d and the structure described by the graphical model leads to semantically more meaningful activation patterns.

E. Differentiating between Estimation Noise and Note Events

Inspecting Fig.1e, however, reveals a further problem. In particular, while the sparsity and TDV terms used in h_d led to a reduction of spurious activations some still remain, see markers. To remove these, we now make use of the example recording of a note played pianissimo (low intensity). Here, the idea is to provide an intuitive way for the user to specify a threshold used to differentiate between estimation noise and actual notes, which is in contrast to many previous methods where the user needs to adjust this value and having to decide which setting works best – each time the recording level or other recording conditions change. As an additional benefit of having this threshold before the optimization process starts, we can include this knowledge as part of the optimization process. For the given low-intensity recording, we compute an activation tensor A_M using h_e in a pre-processing step and define our threshold as $a_m := \max_{k,\ell,n} A_M(k,\ell,n)$. Everything below this threshold will be considered as noise. However, instead of just thresholding A after the estimation as usually done, we make the optimization process aware of this requirement and integrate it as an additional regularizer. Thus the energy below the threshold is not truncated but can be explained within the model leading to clearer activations. To this end, we use the characteristic function in combination with the set $\mathcal{T} := \big([a_m,\infty)\cup\{0\}\big)^{K\times L\times N}$. Fig.1f shows that using h_d to obtain an initialization followed by a refinement using $h_f(A) := h_e(A) + f_6(A)$ with $f_6 := \chi_{\mathcal{T}}$ eliminates most estimation noise from A.

Activations estimated using h_f typically correspond closely to the notes being played and onset positions can be estimated to a high accuracy. The offset position, however, is not always well captured. This is an effect of using unnormalized templates with the sparsity and TDV regularizers, which sometimes shortens and sometimes extends notes to make up for estimation errors. This can often be individually corrected by adjusting the parameters λ_1 and λ_2 – however, the right value depends on the recording. In Fig.1f we chose values for λ_1 and λ_2 leading to such an issue, see marker.

F. Encouraging Meaningful Long-Term Note Activity

As a starting point to eliminate these effects, we can observe in Fig.1f that the activation intensity for affected notes typically drops at the correct offset position (see marking). As a countermeasure we now disallow any change in activation value while a note is active, which we implement by changing the signal model: we replace A in Eq.1 by the Hadamard product of two matrices: $B \odot G$. While A encoded both the activation intensity and length of the note, we split these responsibilities into these two matrices, which enables more direct constraints. In particular, we constrain G in such a way that, for each diagonal in $G(k,:,:)$ for a given k, the entries are only allowed to use a single, shared value, i.e. a strict coupling of values across frames. Since note-lengths cannot be modeled in G, we multiply it pointwise with the binary matrix B which is subject to constraints similar to our Markov-state regularizer (Section III-D).

More precisely, to obtain estimates for B and G we minimize

$$h_g(B,G) := D(V, P(B\odot G)) + \chi_{\widetilde{\mathcal{M}}}(B) + \chi_{\widetilde{\mathcal{T}}}(G),$$

where $\widetilde{\mathcal{M}} := \mathcal{M}\cap\{0,1\}^{K\times L\times N}$ and $\widetilde{\mathcal{T}} := \{A\in\mathcal{T}|\ \|\Delta_D[A]\|_1 = 0\}$. This is a highly non-convex objective function, so we rely on a meaningful initialization to obtain useful results. Here, we use the A obtained via h_f as follows. First, we set $B(k,\ell,n) = 1$ if $A(k,\ell,n) > a_m$, and zero otherwise. Second, we set $G(k,\ell,n) = \max\{A(k,1,n-\ell+1), A(k,2,n-\ell+2),\dots,A(k,L,n-\ell+L)\}$: the maximum over the $(n-\ell+1)$-th diagonal in $A(k,:,:)$. The effect of using these *binary Markov-state* and *strict coupling regularizers* is shown Fig.1g. In particular, the use of $\chi_{\widetilde{\mathcal{T}}}$ indeed leads to constant values on the diagonals of $A := B\odot G$. These values are typically considerably higher than the incorrect values we observed before following an offset position in A. As a result, we found that the parameter estimation for B typically switches from the value 1 to 0 on a diagonal around the correct offset, as otherwise the high values on the diagonal would lead to high error rates after the offset. As a drawback, however, we sometimes observed that a note is shortened too much and the resulting residual energy can lead to new note activations. An example is highlighted in Fig.1g. For this reason, it seems this last combination of regularizers is more useful for correcting note lengths rather than obtaining a full transcription.

IV. Parameter Estimation using the Alternating Direction Method of Multipliers

So far, we focused on the design of meaningful objective functions and their effect on the parameter estimation. In this section, we develop robust algorithms to minimize them. A major problem is that our functions contain non-differentiable (e.g. $\|\cdot\|_1$), infinite (e.g. $\chi_{\mathbb{R}^{K\times L\times N}_{\geq 0}}$) as well as non-convex terms (e.g. $\chi_{\mathcal{M}}$): many classical optimization methods are not stable under these properties. In this context, the *alternating direction method of multipliers (ADMM)* has sparked a lot of interest in recent years [42], [43]. ADMM belongs to a class referred to as *proximal algorithms*, whose importance Parikh and Boyd [44] describe as "much like Newton's method is a standard tool for solving unconstrained smooth optimization problems of modest size, proximal algorithms can be viewed as an analogous tool for non-smooth, constrained, large-scale, or distributed versions of these problems". In general, ADMM is of interest if the objective function comprises several terms that are difficult to minimize jointly but efficiently individually. In this respect, ADMM provides a scheme to split up the objective function, minimize the terms individually and still

provide convergence guarantees for the entire objective. As a result it is not only useful for complex objective functions as in our case but also in big data scenarios, as ADMM's splitting and merging operations fit perfectly into distributed computing schemes like *Map-Reduce*. For a comprehensive introduction, we refer to [42]–[44].

ADMM solves problems of the form

$$\begin{aligned} \underset{x,z}{\operatorname{argmin}} \quad & f(x)+g(z) \\ \text{subject to} \quad & Bx+Cz=c \end{aligned} \tag{2}$$

with $x \in \mathbb{R}^N$, $z \in \mathbb{R}^M$, $B \in \mathbb{R}^{P\times N}$, $C \in \mathbb{R}^{P\times M}$, $c \in \mathbb{R}^P$. Solutions for problem (2) are identical to the ones for the following problem (under some assumptions [41])

$$\underset{\beta}{\operatorname{argmax}} \inf_{x,z} L_\rho(x,z,\beta), \tag{3}$$

$$\begin{aligned} L_\rho(x,z,\beta) := f(x)+g(z)+\langle \beta, Bx+Cz-c\rangle \\ +(\rho/2)\left\|Bx+Cz-c\right\|_2^2 \end{aligned}$$

where L_ρ is referred to as the *augmented Lagrangian* for problem (2) with *penalty parameter* $\rho > 0$ and *dual variable* $\beta \in \mathbb{R}^P$ (see e.g. [41], [42]). A classical method to iteratively solve (3) is to minimize L_ρ w.r.t. x and z, followed by a maximization over β. Convergence guarantees only hold if the minimization is computed jointly in x and z, which in practice is often difficult. The main extension in ADMM is to split the minimization into two steps:

$$\begin{aligned} x^{k+1} &:= \underset{x}{\operatorname{argmin}}\, L_\rho(x, z^k, \beta^k) \\ z^{k+1} &:= \underset{z}{\operatorname{argmin}}\, L_\rho(x^{k+1}, z, \beta^k) \\ \beta^{k+1} &:= \beta^k + \rho(Bx^{k+1}+Cz^{k+1}-c), \end{aligned} \tag{4}$$

where k is the iteration number. Note that the update of β is a gradient ascent on $L_\rho(x^{k+1}, z^{k+1}, \cdot)$ with stepsize ρ. Eckstein and Bertsekas [45] showed that ADMM is equivalent to a firmly non-expansive operator, which enables the application of theorems closely related to the well-known Banach fixed-point theorem. As a consequence, despite the split, ADMM converges for any $\rho > 0$ under relatively mild conditions. In particular, there is no need for f or g to be differentiable or strictly convex or even finite (i.e. they can assume the value ∞) – they only need to be convex, as well as closed and proper. Further, the x and z updates do not even have to be exact but can be approximate to some degree. Since the initial proof [45], convergence results were greatly extended, see [42]–[44] and references therein.

A. Consensus Form ADMM

Though not obvious at first, the splitting of the minimization step in ADMM has considerable consequences and enables us to divide our objective functions into their constituent terms, which are much easier to minimize individually. First, we identify our problem to be of the form used in Eq. (2). To this end, we will treat A as an element of a vector space and do not differentiate between $\mathbb{R}^{M\times L\times K}$ and $\mathbb{R}^{MLK}$. Our objective functions h_a to h_f have the following form

$$\underset{A}{\operatorname{argmin}} \sum_{i=1}^{I} f_i(C_i A), \tag{5}$$

where C_i are linear operators corresponding to the dictionary pattern operator P, the TDV operator Δ_D or simply the identity operator. Note that, with A interpreted as an element of $\mathbb{R}^{MLK}$, each C_i could be represented by a matrix in $\mathbb{R}^{M_i\times MLK}$ for some $M_i > 0$. Problem (5) is equivalent to:

$$\begin{aligned} \underset{x_1,\ldots,x_I,A}{\operatorname{argmin}} \quad & \sum_{i=1}^{I} f_i(x_i) \\ \text{subject to} \quad & \forall i \in \{1,\ldots,I\} : x_i = C_i A, \end{aligned} \tag{6}$$

where $x_i \in \mathbb{R}^{M_i}$. This form is often referred to as the *consensus form* of problem (5) since all local variables x_i have to agree on a common solution enforced by the constraints [46]. In this context, A is called the *central collector*. By setting $X := (x_1,\ldots,x_I)$, $f(X) := \sum_{i=1}^I f_i(x_i)$, $\mathcal{A} := \{\overline{A} := (A_1,\ldots,A_I) \in \mathbb{R}^{MLKI} | A_1 = \ldots = A_I\}$ and $g = \chi_{\mathcal{A}}$, we see that problem (6) is equivalent to:

$$\begin{aligned} \underset{X,\overline{A}}{\operatorname{argmin}} \quad & f(X)+g(\overline{A}) \\ \text{subject to} \quad & X - C\overline{A} = 0 \end{aligned} \tag{7}$$

where C is a block-diagonal matrix

$$C := \begin{pmatrix} C_1 & & \\ & \ddots & \\ & & C_I \end{pmatrix}. \tag{8}$$

Here, the characteristic function $\chi_{\mathcal{A}}$ ensures that the copies of A in $\overline{A}$ are identical. Problem (7) clearly adheres to the form shown in Eq. (2) and thus we can apply ADMM.

To see advantages of this form compared to (5), we first note that the augmented Lagrangian L_ρ for problem (7) is *separable* by construction in each element of $\{x_1,\ldots,x_I\}$, i.e. $L_\rho(\cdot, Z^k, \beta^k)$ partitions its input and processes each disjunct subset independently. More precisely, the augmented Lagrangian for problem (7) can be written as

$$\begin{aligned} L_\rho(X,\overline{A},\beta) := \sum_{i=1}^{I} f_i(x_i) + \langle \beta_i, x_i - C_i A_i\rangle + \frac{\rho}{2}\left\|x_i - C_i A_i\right\|_2^2 \\ +\chi_{\mathcal{A}}(A_1,\ldots,A_I) \end{aligned}$$

where the partition of $\beta = (\beta_1,\ldots,\beta_I)$ is defined analogously to the partition of X. As we can see, a specific x_i appears only in exactly three terms, and every other term is independent of x_i. Thus, we can update each x_i individually, which enables us to develop specialized, efficient minimizers for each term – which can even run in parallel. More precisely, the X update for our problem is equivalent to (for all $i \in \{1,\ldots,I\}$):

$$x_i^{k+1} = \underset{x_i}{\operatorname{argmin}}\, f_i(x_i) + \langle \beta_i^k, x_i - C_i A_i^k\rangle + \frac{\rho}{2}\left\|x_i - C_i A_i^k\right\|_2^2 \tag{9}$$

We develop specific solvers for problem (9) for f_1 to f_6 in Section V.

B. Linearized ADMM

Once all x_i are computed, we can update $\overline{A}$, which corresponds to the z update in Eq. (4), i.e. we need to solve:

$$\operatorname*{argmin}_{\overline{A}} \chi_{\mathcal{A}}(\overline{A}) + \langle \beta^k, X^{k+1} - C\overline{A} \rangle + \frac{\rho}{2} \left\| X^{k+1} - C\overline{A} \right\|_2^2. \quad (10)$$

We can pull the inner product into the norm to obtain the equivalent *proximal form* (leaving out terms independent of $\overline{A}$)

$$\operatorname*{argmin}_{\overline{A}} \chi_{\mathcal{A}}(\overline{A}) + \frac{\rho}{2} \left\| C\overline{A} - (X^{k+1} + (1/\rho)\beta^k) \right\|_2^2.$$

We will use this form repeatedly in the next section. A problem of this form can be identified as a quadratic programming problem with linear equality constraints, for which specific solvers exist [41]. Such methods, however, compute exact solutions and are computationally too expensive for our purposes.

As discussed in [47], problem (10) is easily solved if C would be the identity matrix – in this case we only need to compute an orthogonal projection of $X^{k+1} + (1/\rho)\beta^k$ onto $\mathcal{A}$ which turns out to be straightforward. Following this idea, we now try to extract C out of the norm in (10). To this end, we use here a concept similar to the one presented in [47], referred to as *Linearized ADMM*, and thus our approach can be considered as a combined *Linearized Consensus ADMM*. More precisely, we replace the term $(\rho/2)\left\| X^{k+1} - C\overline{A} \right\|_2^2$ with

$$\rho \langle C^\top C\overline{A}^k - C^\top X^{k+1}, \overline{A} \rangle + (\mu/2) \left\| \overline{A} - \overline{A}^k \right\|_2^2,$$

for some $\mu \geq \rho \left\| C \right\|^2$ where $\left\| \cdot \right\|$ is the spectral norm, and $\overline{A}^k$ and X^{k+1} denote estimates computed in the k-th and $k+1$-th iteration, respectively. This change can be interpreted as adding additional regularizers to the augmented Lagrangian and is sometimes referred to as the *inexact Uzawa* approach – see [47] for a more in-depth discussion. With this change, the $\overline{A}$ update becomes (again pulling all inner products into the norm as above)

$$\operatorname*{argmin}_{\overline{A}} \chi_{\mathcal{A}}(\overline{A}) + \frac{\rho}{2} \left\| \overline{A} - (\overline{A}^k - \frac{\rho}{\mu} C^\top (C\overline{A}^k - X^{k+1} - \frac{1}{\rho}\beta^k)) \right\|_2^2.$$

With our variable $\overline{A}$ freed of C, the solution simply becomes the orthogonal projection π of $\overline{A}^k - \frac{\rho}{\mu} C^\top (C\overline{A}^k - X^{k+1} - \frac{1}{\rho}\beta^k)$ onto the set $\mathcal{A}$:

$$\overline{A}^{k+1} = \pi_{\mathcal{A}}(\overline{A}^k - \frac{\rho}{\mu} C^\top (C\overline{A}^k - X^{k+1} - \frac{1}{\rho}\beta^k)),$$

which due to the definition of $\mathcal{A}$ corresponds to simply taking the average over its components, i.e. with

$$A^{k+1} := \frac{1}{I} \sum_{i=1}^{I} A_i^k - \frac{\rho}{\mu} C_i^\top (C_i A_i^k - x_i^{k+1} - (1/\rho)\beta_i^k)$$

the updated variable is then $\overline{A}^{k+1} := (A^{k+1}, \ldots, A^{k+1})$. Note that $A^{k+1} = A_1^{k+1} = \ldots = A_I^{k+1}$ is guaranteed, which allows us to leave out the index i on A^k in the following and simplify the update even further:

$$A^{k+1} := \frac{1}{I} \sum_{i=1}^{I} A^k - \frac{\rho}{\mu} C_i^\top (C_i A^k - x_i^{k+1} - (1/\rho)\beta_i^k) \quad (11)$$

As shown in [47] convergence results also hold for this ADMM variant.

As a drawback of this modification, however, we found that the condition $\mu \geq \rho \left\| C \right\|^2$ required to prove convergence is too restrictive in practice and the resulting method might require an excessive amount of iterations to converge. As an alternative, we developed a second update method for $\overline{A}$, which is based on the introduction of additional slack variables. With this method, we achieved an improved convergence rate compared to linearized ADMM. Existing convergence proofs, however, do not hold for this approach anymore. Due to space constraints this variant is presented in an external annex [48].

Finally, to complete our ADMM framework for problem (7), we give the update rules for β. Here, we can exploit the block structure of C and express the update more compactly for all $i \in \{1, \ldots, I\}$ as

$$\beta_i^{k+1} := \beta_i^k + \rho(x_i^{k+1} - C_i A^{k+1}). \quad (12)$$

V. Minimizing the Individual Terms

With the general framework in place, we now develop methods for minimizing the individual terms in our objective function, i.e. implement Eq. 9 for each regularizer or data fidelity term. For a lack of space we omit most proofs but point to relevant work for those terms that had been introduced previously in similar forms.

A. Kullback-Leibler Data Fidelity Term

We start with the Kullback-Leibler term D, for which Eq. 9 has the following form:

$$x_1^{k+1} := \operatorname*{argmin}_{x_1} D(V, x_1) + \langle \beta_1^k, x_1 - PA^k \rangle + \frac{\rho}{2} \left\| x_1 - PA^k \right\|_2^2.$$

Without a linear transform in the arguments of D (result of our construction), the calculation becomes straightforward and x_1^{k+1} can be found analytically: The gradient of the right hand side is $\beta_1^k - \frac{V}{x_1} + \rho(x_1 - PA^k) + 1$, where the division is element-wise and 1 is the all-one vector. To find the minimizing argument, we set the gradient to zero and obtain

$$x_1^{k+1} = \frac{\rho PA^k - \beta_1^k - 1 + \sqrt{(\rho PA^k - \beta_1^k - 1)^2 + 4\rho V}}{2\rho}, \quad (S_1)$$

where the square and square root are again element-wise.

B. Non-negativity Term

The non-negativity term does not include a linear transformation, i.e. C_2 is the identity matrix, and problem (9) has the following form

$$x_2^{k+1} := \operatorname*{argmin}_{x_2} \chi_{\mathbb{R}_{\geq 0}^{K \times L \times N}}(x_2) + \langle \beta_2^k, x_2 - A^k \rangle + \frac{\rho}{2} \left\| x_2 - A^k \right\|_2^2,$$

which in proximal form is

$$x_2^{k+1} := \operatorname*{argmin}_{x_2} \chi_{\mathbb{R}_{\geq 0}^{K \times L \times N}}(x_2) + \frac{\rho}{2} \left\| x_2 - (A^k - \frac{1}{\rho}\beta_2^k) \right\|_2^2.$$

As already seen above, the solution here is an orthogonal projection of $A^k - \frac{1}{\rho}\beta_2^k$ onto the set $\mathbb{R}_{\geq 0}^{K\times L\times N}$, i.e. the solution is a half-wave rectifier [42]:

$$x_2^{k+1} := \pi_{\mathbb{R}_{\geq 0}^{K\times L\times N}}(A^k - \frac{1}{\rho}\beta_2^k) = \max(A^k - \frac{1}{\rho}\beta_2^k, 0), \quad (S_2)$$

where $\max$ is entrywise and 0 is the all-zero matrix.

C. LASSO and Total Diagonal Variation Terms

Problem (9) in proximal form for the LASSO or ℓ_1 term is

$$x_3^{k+1} := \underset{x_3}{\operatorname{argmin}} \left\|x_3\right\|_1 + \frac{\rho}{2\lambda_1}\left\|x_3 - (A^k - \frac{1}{\rho}\beta_3^k)\right\|_2^2.$$

A problem in computing x_3^{k+1} is that $\left\|\cdot\right\|_1$ is not differentiable everywhere and thus we cannot simply set the gradient to zero as before. However, similar strategies are possible when we use the sub-derivative instead, which is set valued and, compared to the derivative, contains all possible linearizations of the function that do not locally exceed the function value [49]. For example, the sub-derivative for the absolute value function is either $\{-1\}$ or $\{1\}$ everywhere, except for zero where it is the closed interval $[-1, 1]$. In this context, analogue to the derivative, a minimum of a convex function is characterized by its sub-derivative containing the value zero. The solution to the above problem derived this way is commonly referred to as *soft-thresholding* [43]:

$$x_3^{k+1} := \operatorname{sign}(v)\max(|v| - \frac{\lambda_1}{\rho}, 0), \quad (S_3)$$

where $v := A^k - \frac{1}{\rho}\beta_3^k$ and all operations are element-wise.

The main difference between the LASSO and our new Total Diagonal Variation term is the application of the Δ_D operator. This operator is linear in its argument and thus could be represented by a matrix, i.e. it takes the role of the matrix C_4 in the last section. Due to our construction the Δ_D operator is not applied to x_4 (but to the central collector A), which again turns out to simplify our minimization problem considerably: The solution for the TDV term is almost identical to the LASSO term, with

$$x_4^{k+1} := \operatorname{sign}(v)\max(|v| - \frac{\lambda_2}{\rho}, 0) \quad (S_4)$$

and $v := \Delta_D A^k - \frac{1}{\rho}\beta_4^k$.

D. Markov-State Regularizer

For our new Markov-State regularizer we have to solve the following problem (proximal form)

$$\begin{aligned} x_5^{k+1} &:= \underset{x_5}{\operatorname{argmin}}\, \chi_{\mathcal{M}}(x_5) + \frac{\rho}{2}\left\|x_5 - (A^k - \frac{1}{\rho}\beta_5^k)\right\|_2^2 \quad (13)\\ &= \pi_{\mathcal{M}}(A^k - \frac{1}{\rho}\beta_5^k) \end{aligned}$$

To compute this orthogonal projection, we can use dynamic programming. Setting $v := A^k - \frac{1}{\rho}\beta_5^k \in \mathbb{R}_{\geq 0}^{K\times L\times N}$, we define for each k a cost matrix $C_k \in \mathbb{R}^{L\times N}$

$$C_k(\ell, n) = \sum_{\tilde{\ell}\in\{1,\ldots,L\}\setminus\{\ell,\ldots,\ell+\lceil w/s\rceil\}} v(k, \tilde{\ell}, n)^2. \quad (14)$$

Algorithm 1 ADMM for Minimizing h_b to h_f

Initialization:
  Set penalty ρ to a positive value.
  Set I to the number of terms in the objective function.
  Initialize A with random values.
  For $i = 1$ **to** I**:**
    Set x_i to $C_i A$.
    Set β_i to 0. *// all-zero vector*
Repeat Until Convergence:
  For $i = 1$ **to** I**:**
    Update x_i using Eq. S_i.
  Update A using Eq. 11.
  For $i = 1$ **to** I**:**
    Update β_i using Eq. 12.

This is the error (w.r.t. the squared Euclidean norm in Eq. 13) we make in frame n if we encode state ℓ in that frame for key k (compare also Section III-D). Using C_k and dynamic programming, we can find the state sequence $\ell_1^k, \ldots, \ell_N^k$ minimizing the total cost $\sum_n C_k(\ell_n^k, n)$ among all sequences valid under the graphical model shown in Fig. 2. To this end, we recursively define an accumulated cost matrix $D_k \in \mathbb{R}_{\geq 0}^{L\times N}$ and a step matrix $E_k \in \{1, \ldots, L\}^{L\times N}$ as

$$D_k(\ell, n) := C_k(\ell, n) + \begin{cases} D_k(\ell - 1, n - 1), & \ell > 1 \\ \min\limits_{\tilde{\ell}\in\mathcal{S}}(D_k(\tilde{l}, n - 1)) & \ell = 1 \end{cases} \quad (15)$$

$$E_k(\ell, n) := \begin{cases} \ell - 1, & \ell > 1 \\ \operatorname{argmin}\limits_{\tilde{\ell}\in\mathcal{S}}(D_k(\tilde{\ell}, n - 1)) & \ell = 1 \end{cases} \quad (16)$$

where $\mathcal{S} := \{1, L_M, L_M + 1, \ldots, L\}$ is the set of states that allow a return to the first state and $D_k(\ell, 1) := C_k(\ell, 1)$ for all ℓ. We start by setting $\ell_N^k = \operatorname{argmin}_\ell D_k(\ell, N)$ and set $\ell_n^k = E_k(\ell_{n+1}^k, n + 1)$ for $n \in \{1, \ldots, N - 1\}$. Note that the definition of $\mathcal{S}$ only allows state transition that are valid according to our graphical model (including the minimum note duration L_M). Having the state sequence, x_5^{k+1} is

$$x_5^{k+1}(k, \ell, n) := \begin{cases} v(k, \ell, n), & \ell \in \{\ell_n^k, \ldots, \ell_n^k + \lceil w/s\rceil\} \\ 0 & \text{otherwise} \end{cases} \quad (S_5)$$

Due to the simplicity of our graphical model, implementations of Eqns. 14 to 16 can be highly efficient. First, all computations can be parallelized over k. Second, C_k can be computed by one element-wise multiplication of v followed by one matrix multiplication (for the sum). Third, for D_k and E_k we only need to compute the minimizer for $\ell = 1$ and the other entries can be computed independently given the results for the previous frame. Therefore, the computation can be almost perfectly parallelized over ℓ as well. Overall, we found computing the solution for this regularizer to be only marginally slower compared to the other regularizers.

E. Thresholding Set Term

For the thresholding set term, we need to solve

$$x_6^{k+1} := \operatorname*{argmin}_{x_6} \chi_{\mathcal{T}}(x_6) + \frac{\rho}{2}\|x_6 - (A^k - \frac{1}{\rho}\beta_6^k)\|_2^2 \quad (17)$$
$$= \pi_{\mathcal{T}}(A^k - \frac{1}{\rho}\beta_6^k),$$

where $\mathcal{T} := \big([a_m, \infty) \cup \{0\}\big)^{K\times L\times N}$ (repeated from Section III-E). Using again v as a shorthand, setting $v := A^k - \frac{1}{\rho}\beta_6^k$, the solution is

$$x_6^{k+1}(k,\ell,n) := \begin{cases} v(k,\ell,n), & a_m < v(k,\ell,n) \\ a_m & \frac{a_m}{2} \le v(k,\ell,n) \le a_m \\ 0 & \text{otherwise.} \end{cases} \quad (S_6)$$

This completes our method for minimizing h_b to h_f. Algorithm 1 summarizes the individual steps in pseudo code. Further improvements are described in the external annex [48], where an alternative update rule for A is derived, which converged quicker in practice than the linear ADMM based solution presented here, and presents an adaptive scheme, which updates the penalty parameter ρ after each iteration to increase the convergence speed (using similar ideas as presented in [42], where ADMM is discussed as a primal-dual algorithm [49], which leads to an effective heuristic for adjusting ρ in such a way that the primal and dual residual are balanced). Additionally, a Matlab implementation is available online[1].

F. Binary Markov-State and Strict Coupling Regularizer

For the binary Markov-state and strict coupling regularizers we do not use ADMM but minimize the augmented Lagrangian directly. We do this for two reasons. First, it shows that the augmented Lagrangian is often also useful without ADMM. Second, since our objective function h_g is non-convex, the convergence guarantees provided by ADMM do not hold anymore. Due to space constraints, we only present a summary of the iterative updates here, with details on the derivation in an external annex [48].

In general, we use the augmented Lagrangian for the following problem, which is equivalent to our objective function h_g,

$$\begin{aligned} \text{argmin} \quad & D(V, X_1) + \chi_{\widetilde{\mathcal{M}}}(X_5) + \chi_{\widetilde{\mathcal{T}}}(X_6) \\ \text{subject to} \quad & X_1 = PX_2, \quad X_2 = X_3 \odot X_4 \\ & X_3 = X_5, \quad X_4 = X_6 \end{aligned} \quad (18)$$

The additional slack variables in the equality constraints are introduced to simplify the decoupling of the actual variables. Minimizing for X_1 to X_6 individually, we obtain the following update rules

$$X_1 = \frac{\rho PX_2 - \beta_{X_1} - 1 + \sqrt{(\rho PX_2 - \beta_{X_1} - 1)^2 + 4\rho V}}{2\rho}$$

$$X_2 = (P^\top P + I)^{-1}(P^\top X_1 + X_3 \odot X_4 + (1/\rho)(P^\top \beta_{X_1} - \beta_{X_2}))$$

$$X_3 = (\frac{1}{\rho}\beta_{X_2} \odot X_4 + X_4 \odot X_2 - \frac{1}{\rho}\beta_{X_3} + X_5)/(X_4 \odot X_4 + 1)$$

[1]https://code.soundsoftware.ac.uk/projects/adpt

TABLE I
DESCRIPTION OF DATASET UM-NI.

ID	Composer	Piece	Performer
U01	Bach	BWV. 851	Colafelice
U02	Beethoven	Op. 10 No. 3	Wang
U03	Chopin	Op. 25 No. 11	Kim
U04	Haydn	HobXVI 52	Mizumoto
U05	Liszt	Polonaise E-Maj	Denisova
U06	Mendelssohn	Op. 54	Sham
U07	Mozart	K284-01	Ozaki
U08	Ravel	Alb. D. Grac.	Teo
U09	Schubert	Op. 142 No. 3	Chon
U10	Stravinsky	Op. 7 No. 4	Lin

$$X_4 = (\frac{1}{\rho}\beta_{X_2} \odot X_3 + X_3 \odot X_2 - \frac{1}{\rho}\beta_{X_4} + X_6)/(X_3 \odot X_3 + 1)$$

$X_5 =$ algorithm used for Markov-State term above, using a modified cost matrix $\tilde{C}_k$, see below

$$X_6(k,\ell,n) := \begin{cases} w(k,\ell,n), & a_m < w(k,\ell,n) \\ a_m & \frac{a_m}{2} \le w(k,\ell,n) \le a_m \\ 0 & \text{otherwise} \end{cases}$$

Here, $P^\top$ denotes the adjoint of P, the division is element-wise, the shorthands $w(k,\ell,n) := \frac{1}{L}\sum_{\tilde{\ell}=1}^{L} u(k,\tilde{\ell},n+\ell-\tilde{\ell})$, $u := X_4 + \frac{1}{\rho}\beta_{X_4}$ and $v := X_3 + \frac{1}{\rho}\beta_{X_3}$, as well as the cost matrix $\tilde{C}_k$

$$\tilde{C}_k(\ell,n) = \sum_{\substack{\tilde{\ell}\in\{1,\dots,L\}\setminus \\ \{\ell,\dots,\ell+\lceil w/s\rceil\}}} v(k,\tilde{\ell},n)^2 + \sum_{\tilde{\ell}\in\{\ell,\dots,\ell+\lceil w/s\rceil\}} (v(k,\tilde{\ell},n)-1)^2 \quad (19)$$

After the minimization, we maximize over the dual variables (as we have done in ADMM):

$$\beta_{X_1} = \beta_{X_1} + \rho(X_1 - PX_2) \qquad \beta_{X_2} = \beta_{X_2} + \rho(X_2 - X_3 \odot X_4)$$

$$\beta_{X_3} = \beta_{X_3} + \rho(X_3 - X_5) \qquad \beta_{X_4} = \beta_{X_4} + \rho(X_4 - X_6)$$

Iterating these updates, X_5 and X_6 contain estimates for B and G.

VI. EXPERIMENTS

To illustrate the performance of our proposed method, we conducted a series of experiments. Since our method relies on the existence of individual recordings for each piano key, the set of available datasets is more limited than in most transcription scenarios. Overall, we use three datasets, each having specific properties.

A. University of Minnesota / Native Instruments (UM-NI) Dataset

The first dataset was also used in [10] and comprises ten MIDI files downloaded from the University of Minnesota piano-e-competition website[2]. These were recorded using a Yamaha Disklavier during an international piano playing competition and therefore closely capture the actual, real-world performance of skilled pianists. The pieces were selected to cover a broad range of composers and performers but were otherwise

[2]http://www.piano-e-competition.com

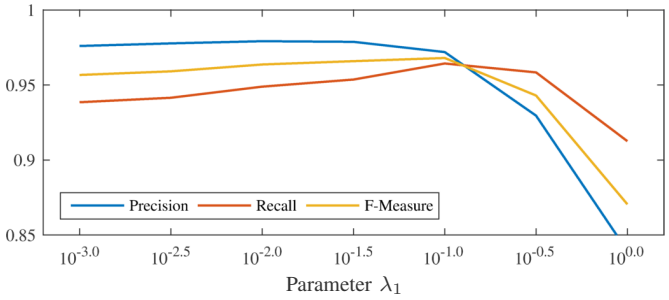

Fig. 3. Averaged evaluation results using the UM-NI dataset for various values of the sparsity parameter λ_1.

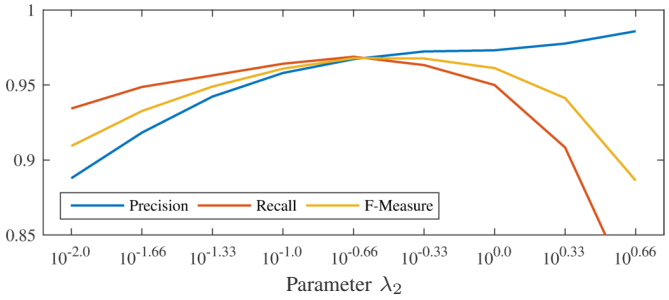

Fig. 4. Averaged evaluation results using the UM-NI dataset for various values of the total diagonal variation parameter λ_2.

selected randomly, see Table I for an overview. To create high quality audio versions from these MIDI files, we employed Native Instruments' Vienna Concert Grand VST plugin, which comprises samples of a Boesendorfer 290 concert grand with an uncompressed size of almost 14 GB. Additionally, we used the plugin to create recordings of single notes for that piano, each 6 seconds long and played in mezzo forte (MIDI velocity 75). We refer to this dataset in the following as *UM-NI* and use it to investigate the influence of parameters on our proposed method and to compare our results with the system presented in [10].

To evaluate a method, we employ precision (P), recall (R), and F-measure (F) values as used in the MIREX evaluation campaigns. A detected note is considered correct if there is a note in a corresponding ground truth MIDI file having the same MIDI pitch, with on onset position up to 50ms apart from the detected note. Every ground truth note can only validate up to one detected note. By counting the number of correctly detected notes (TP), incorrectly detected extra notes (FP) and incorrectly missed notes (FN), we can define the precision P := TP / (TP + FP), recall R := TP / (TP + FN) and f-measure F := 2 · P · R / (P + R).

Overall, our proposed system has two main parameter for sparsity λ_1 and total diagonal variation λ_2. Informal tests using the UM-NI showed a good performance setting $\lambda_1 = 0.1$ and $\lambda_2 = 0.5$. With our first two experiments we investigate the effect these two parameters have in more detail. We first set up the dictionary tensor P using the single note recordings. Then, we use our system (without the binary Markov-state and strict coupling regularizer for now) to obtain an activation value for a middle C played pianississimo (very softly) – this activation

TABLE II
EVALUATION RESULTS: PRECISION, RECALL AND F-MEASURE FOR DETECTED NOTE ONSETS FOR UM-NI DATASET.

		U01	U02	U03	U04	U05	U06	U07	U08	U09	U10	Av
[10]	P	91	86	76	86	90	94	83	86	92	84	**87**
	R	93	89	83	87	92	84	89	87	91	96	**89**
	F	92	87	79	86	91	89	86	86	91	89	**88**
Prop.	P	100	97	94	98	96	98	98	97	98	95	**97**
	R	100	97	94	97	93	94	96	98	100	99	**97**
	F	100	97	94	97	95	96	97	97	99	97	**97**

value minus 10% is used to set the minimum activation value a_m. After estimating A using our proposed method (300 iterations minimizing h_d, which is used as initialization for 300 iterations with h_f), we detect an onset for key k in frame n if $A(k,1,n) \geq a_m$ and $A(k,1,n) \geq A(k,1,\tilde{n})$ for all $\tilde{n} \in \{n - L_M, \ldots, n + L_M\}$. Fig. 3 shows our evaluation results for various values of λ_1 fixing $\lambda_2 = 0.5$, and Fig. 4 for various values of λ_2 fixing $\lambda_1 = 0.1$. In Fig. 3 we can observe that lower values for λ_1 actually yield higher precision values, which is counter-intuitive at first as a higher weight for the sparsity term typically yields fewer and more meaningful activations. The underlying cause here is the use of a dictionary containing unnormalized templates, as already discussed in Section III. Further, we can see that the f-measure is reasonably stable for $\lambda_1 \leq 10^{-1}$. The break-even point is around 10^{-1}, which also coincides with a small local maximum for the f-measure. Therefore, we keep $\lambda_1 = 0.1$ in the following. For λ_2, the f-measure is quite stable and above 0.94 with $\lambda_2 \in [10^{-1.33}, 10^{0.33}]$. The break-even point and the maximum of the f-measure are between $10^{-0.66}$ and $10^{-0.33}$, which made us adjust the λ_2 value slightly to 0.4 in the following. Evaluation results for each piece in UM-NI using these settings are shown in Table II, which enables a comparison with the method presented in [10]. As discussed in Section II, this system is a hybrid between NMD and FS-HMM and employs the same spectro-temporal patterns we use as internal dictionary. As we can see, the proposed system clearly outperforms [10] on every recording. An informal manual investigation showed that the reason for this considerable difference was in many cases that the method proposed in [10] generated note object candidates for a given pitch by actually modelling energy for another pitch – an effect resulting from the decoupled parameter estimation (as discussed at the end of Section II). Our proposed method does not decouple parameters for different pitches during the important first step using convex terms and thus is less prone to run into less meaningful local minima overall.

B. MIDI Aligned Piano Sounds (MAPS) Dataset

While the UM-NI dataset contains MIDI files of real performances, the corresponding audio recordings are synthesized, which can have a great influence on evaluation results. Therefore, we conducted additional experiments using the MAPS dataset [17], which contains audio recordings of MIDI files played on a Yamaha Disklavier. In contrast to UM-NI, MAPS contains only score-like MIDI files and is divided into two subsets, which correspond to different microphone placements. For the ENSTDkCl subset a close miking configuration was

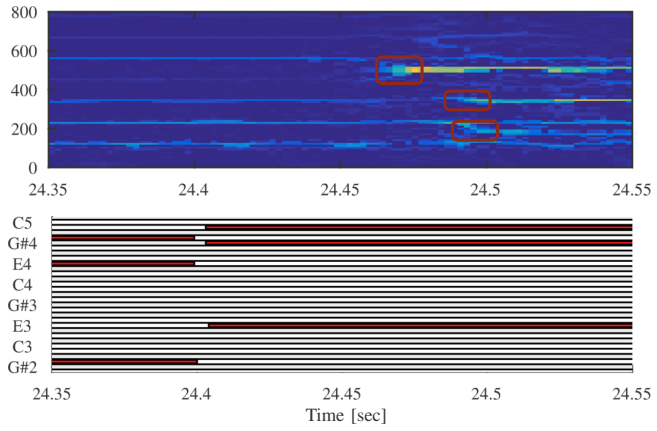

Fig. 5. Spectrogram for a section of the recording *MAPS_MUS-mendel_op62_5_ENSTDkAm* and corresponding annotation MIDI file as contained in the MAPS dataset. Markers indicate onset positions in the recording.

TABLE III
PRECISION, RECALL AND F-MEASURE IN PERCENT FOR VARIOUS METHODS. THE RESULTS ARE GIVEN SEPARATELY FOR THE TWO SUBSETS RECORDED USING A YAMAHA DISKLAVIER AS PROVIDED BY THE MAPS DATASET.

Subset	Method	P	R	F
Close	Benetos/Ewert/Weyde [32]	—	—	63
	FitzGerald/Cranitch/Coyle [50]	60	59	58
	O'Hanlon/Nagano/Plumbley [34]	79	77	78
	Tolonen/Karjalainen [51]	62	22	32
	Vincent/Bertin/Badeau [52]	72	66	67
	Proposed (±50ms)	**91**	**88**	**89**
	Proposed (±100ms)	**96**	**93**	**95**
Ambient	Bertin/Badeau/Vincent [23]	47	45	45
	Carabias-Orti et al. [53]	—	—	52
	Klapuri [54]	—	—	55
	Marolt [22]	64	54	58
	Virtanen [39]	34	35	34
	Proposed (±50ms)	**86**	**85**	**85**
	Proposed (±100ms)	**93**	**92**	**93**

used, while the ENSTDkAm recordings were made using an ambient configuration. Due to room acoustics, the latter contains a considerable amount of reverberation and thus is generally considered as the more difficult to transcribe.

For both datasets, we decided to provide results using two different temporal tolerances: one using the usual ±50ms and one using ±100ms. We include the greater tolerance as we found that the actual temporal accuracy of the MIDI-based annotations included in MAPS sometimes exceed the documented accuracy of 20ms by an order of magnitude. This is illustrated in Fig. 5, where we see three notes of an E-major chord that are supposed to start concurrently at 24.404sec in the MIDI file. In the recording, however, the corresponding onsets are approximately at 24.47sec (B), 24.49sec ($G^\sharp$) and 24.495sec (E). Overall, in this example, the difference is 65-90ms. Manually inspecting the dataset using our own MIDI parser and verifying the findings using the Audacity audio editor, we observed that these differences are often not simple offsets. Rather, some seem to be caused by slow drifts which could be related to differences in clock speed between the MIDI and the audio equipment. More importantly though we also observed a considerable temporal jitter which, for instance, leads to concurrent notes being played asynchronously in the recording (as in Fig. 5). This could be related to a limitation of the Disklavier series and player pianos in general. In particular, when a Disklavier is controlled via direct MIDI input, the instrument plays each note as quickly as possible to minimize delays. Depending on the piano key and the velocity, however, each hammer needs a different amount of time to hit the strings, which consequently can lead to different onset times in a recording. Since the MAPS documentation suggests that the recordings were made using direct MIDI input, this could explain the apparent temporal jitter we observed. Therefore, we believe that the greater temporal tolerance might provide a more realistic impression of the transcription performance.

The results for our proposed method as well as various previously published methods ([23], [32], [34], [53], [55]) are given in Table III, with our method using the parameters found using the UM-NI dataset as discussed above. We remark that many of these methods do not use single note recordings as training material as we do, and thus the numbers typically cannot fairly be compared with our method. Some general observations, however, might be possible. In particular, the considerable jump in f-measure from previous methods might indicate that our proposed method modeling both spectral and temporal signal properties could be capable of exploiting the provided prior knowledge in the form of single note recordings to a high degree. Even under reverberated conditions (ambient dataset in Table III), the drop in performance for our method (from 95% f-measure to 93%) is less pronounced than for the remaining methods (average f-measure for the close subset is 60% and 49% for the ambient subset). Overall, combining the results from the UM-NI and the MAPS subsets could potentially indicate that our proposed method can be used to obtain transcription results of high accuracy even under real world conditions.

C. Influence of Individual Regularizers and Error Analysis

To indicate the influence of each regularizer in our method, we conducted a series of experiments using the close-miking subset of the MAPS collection – the ambient subset showed a similar behavior. To this end, we start with our objective function h_b, which contains only the Kullback-Leibler and the non-negativity terms. By adding the remaining regularizers successively, we illustrate the relative change in performance. A challenge, however, is that our proposed method contains components inside its parameter estimation procedure that most previous methods implement as an additional post-processing step. For example, many methods employ a Hidden Markov Model to decode NMF activations into a set of note objects [2] – a task undertaken by our Markov-State regularizer inside the parameter estimation. Therefore, we provide two different results: one using the same decision logic as used before (essentially binarization, compare Section VI-A) and one using an HMM that decodes note objects from the final activations similar to existing methods [2] (but following the graphical

TABLE IV
F-MEASURE IN PERCENT FOR DETECTED NOTE ONSETS FOR FOR SEVERAL VARIANTS OF OUR PROPOSED METHOD ELIMINATING INDIVIDUAL REGULARIZERS USING THE CLOSE-MIKING SUBSET OF THE MAPS DATASET. THE *Direct* AND *HMM* COLUMNS REFER TO DIFFERENT METHODS FOR PRODUCING THE FINAL NOTE OBJECTS AS DESCRIBED IN THE TEXT.

Regularizer	**Direct**	**HMM**
Kullback-Leibler & Non-Negativity	31	56
+ Sparsity	74	81
+ Total Diagonal Variation	90	91
+ Markov-State Reg.	94	94
+ Thresholding Reg.	95	95
+ Bin. Markov-State & Strict Coupling Reg.	92	93

model shown in Fig. 2 to account for the fact that we use a structured dictionary tensor and not a flat matrix). The results are shown in Table. IV.

As we can see, using only the Kullback-Leibler divergence and the non-negativity term results are significantly worse (31/56%) – worse than many of the methods shown in Table III despite our use of instrument-specific single note recordings. Here, without any additional regularizers, the resulting activations are often unstructured and contain many spurious entries. In particular, onset sounds for notes with velocities vastly different from the ones used for the single notes are synthesized using what seems like random combinations of templates from unrelated piano keys. Including the sparsity term improves the performance considerably (74/81%). As already discussed in Section III-B, the sparsity term alone already eliminates many spurious activations but also introduces additional ones due to our use of unnormalized templates. Another considerable jump in performance is the result of using our TDV term (90/91%) – this term is particularly useful due to our approach of using unnormalized dictionaries which already capture the amplitude progression of a piano sound over time, including any non-stationary changes in spectral properties. This leads to very little fluctuation of activity over time and thus we can set the relative importance of the TDV term very high. The TDV term also leads to similar semantics as Markov terms, which is possible due to the Bakis-type structure in our graphical model – however, in a completely convex formulation eliminating any numerical problems related to usual Markov-type constraints. This is evident in the fact that the 'HMM' value is not much higher than the 'Direct' f-measure.

With respect to the additional non-convex terms, including the Markov constraints within instead of outside the parameter estimation introduces additional harder semantics, which eliminate some of the remaining uncertainties (94/94%). Further, including the thresholding term inside the parameter estimation allows explaining some energy in the signal which would remain unexplained otherwise, which leads to another small improvement (95/95%). Finally, including the binary Markov-state and strict coupling regularizers reduces again the onset f-measure, as already discussed in Section III-F. Here the reason is that the strict coupling of activations (similar to NMD in concept) is too restrictive and can lead to additional, incorrect note detections – the usefulness of these terms thus remains limited to the estimation of note lengths.

Overall, one might argue that the non-convex terms do not considerably increase the performance anymore compared to the results obtained from the convex terms alone, with the f-measure only increasing from 90% to 95%. However, this small change means that we actually halved the number of incorrect notes in our result using the non-convex terms and only expect five instead of ten wrong notes in a hundred notes.

In another experiment, we investigate the capability of our method to additionally compute the duration of each note. As evaluation measure, we use the procedure used in MIREX, i.e. a detected note is considered as correct only if the onset is correct as described above and the detected note duration is within 20% of the corresponding note-length in the ground truth MIDI file. To obtain the correct note length from a given ground truth MIDI file, we parsed the events related to the sustain pedal in each MIDI file and adjusted the note lengths accordingly. We evaluate our proposed method in two configurations: first, using the activity A obtained by minimizing h_d followed by h_f, and second, using B and G resulting from a minimization of h_g, with the solution initialized based on the A obtained from the first step (compare Section V-F). Using the resulting A and B, we determine a duration for each onset, which we detect as before. More precisely, for a detected onset for key k in frame n, we find ℓ with $A(k, \ell, n + \ell - 1) \geq a_m$ and $A(k, \ell + 1, n + \ell) < a_m$ and set the duration for the note to the time in seconds corresponding to ℓ frames. For the second method, we do the same but find ℓ with $B(k, \ell, n + \ell - 1) = 1$ and $B(k, \ell + 1, n + \ell) = 0$. Using a tolerance of $\pm$100ms for the onset, we compute precision, recall and f-measure using the duration-based error measure. For the first method, using the close miking and ambient subsets in MAPS, we obtain an f-measure of 46% and 38%, respectively. Using the second method, we yield an f-measure of 59% and 55%, respectively. This illustrates, that while using h_g for onset detection can decrease the performance due to unrealistic assumptions expressed by the regularisers, it can improve the note length estimation by a considerable amount.

As a final step, we manually tried to identify some general sources of error to find out why and when our method failed. First, our approach of using the same threshold for all piano keys, and choosing this threshold based on a single note, was sometimes not precise enough. Therefore, the performance of our method could be improved if the minimum activity a_m would be available for more piano keys, for example by providing additional recordings of other softly played notes. As a second cause, when the sustain pedal is used, some strings are cross-excited when strings in a neighborhood are hit – depending on the velocity and instrument specific factors. It is a matter of definition if such detections should indeed be counted as transcription errors. Third, depending on the playing style and the velocity in particular, the temporal decay rates for partials and the spectral envelope change sometimes drastically – with considerable differences between different instruments and also for individual keys on the same piano. Since we use only one time-frequency pattern for each key, this can lead to pattern mismatches, leaving residual energy which in some cases is modeled using wrong patterns. These pattern mismatches had a stronger effect under reverberant conditions

(which explain the lower results for the ambient subset).

We also investigated sources of error for the note length detection. First, the duration for softly played notes is more often incorrect than for forte note, which is caused by a difference in the energy decay between our mezzo-forte pattern and the actual note. Second, the use of the sparsity term during the initialization can lead to activation diagonals that are too short (as later templates in a pattern contain less energy and are not activated anymore). While the sparsity term is not used in h_g, it is used to obtain an initialization and we observed that the update rules presented in Section V-F cannot always recover from this error. Third, after the offset, the string is still vibrating for a short while (release state in ADSR model) introducing a principal temporal uncertainty where the actual note end is.

VII. Conclusions

We presented a method for transcribing pitched-percussive instruments such as the piano in controlled recording conditions. Compared to NMF, where time and frequency are strictly separated, our method employs spectro-temporal patterns to model the temporal evolution for each individual piano key. In contrast to non-negative matrix deconvolution, these patterns can be of variable length if activated. From a numerical point of view, our approach employs a combination of convex and non-convex regularizers, which penalize unwanted behavior in an otherwise loosely defined signal model. This is in contrast to the FS-HMM and similar approaches, where the temporal evolution of spectra is directly enforced by a graphical model. The result is a highly accurate parameter estimation which does not rely on unstable parameter decoupling techniques and thus is less prone to poor local minima. The optimization framework ADMM that we used is highly extensible and should be useful in many other audio-processing scenarios. Overall, the combination of prior knowledge available in controlled recording conditions with our proposed signal model seems to provide a considerable boost in transcription performance. For the future it could be interesting to post-process the output of our method using discriminative methods – this could combine the best of two worlds, as our method provides a straightforward integration of prior knowledge in the form of single notes and thus a straightforward adaptability to new acoustic environments, while discriminative methods such as RNN-based language models [56] capture higher level semantics and musical expectation.

Acknowledgement

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) programme EP/L019981/1.

References

[1] J. A. Moorer, "On the transcription of musical sound by computer," *Computer Music Journal*, pp. 32–38, 1977.

[2] A. P. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. New York: Springer, 2006.

[3] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[4] H. Kirchhoff, S. Dixon, and A. Klapuri, "Multi-template shift-variant non-negative matrix deconvolution for semi-automatic music transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 415–420.

[5] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. Springer-Verlag, 1991.

[6] A. Ozerov, C. Févotte, and M. Charbit, "Factorial scaled hidden Markov model for polyphonic audio representation and source separation," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2009, pp. 121–124.

[7] G. Mysore and M. Sahani, "Variational inference in non-negative factorial hidden markov models for efficient audio source separation," in *Proceedings of the International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, 2012, pp. 1887–1894.

[8] M. Nakano, J. Le Roux, H. Kameoka, T. Nakamura, N. Ono, and S. Sagayama, "Bayesian nonparametric spectrogram modeling based on infinite factorial infinite hidden Markov model," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011, pp. 325–328.

[9] Z. Ghahramani and M. I. Jordan, "Factorial hidden markov models," *Machine Learning*, vol. 29, no. 2-3, pp. 245–273, 1997.

[10] S. Ewert, M. D. Plumbley, and M. Sandler, "A dynamic programming variant of non-negative matrix deconvolution for the transcription of struck string instruments," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 569–573.

[11] I. Barbancho, A. Barbancho, A. Jurado, and L. Tardón, "Transcription of piano recordings," *Applied Acoustics*, vol. 65, no. 12, pp. 1261–1287, 2004.

[12] A. T. Cemgil, "Bayesian music transcription," Ph.D. dissertation, Radboud University Nijmegen, 2004.

[13] S. Dixon, "On the computer recognition of solo piano music," in *Proceedings of Australasian Computer Music Conference*, 2000, pp. 31–37.

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

[15] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2121–2133, 2010.

[16] C. Yeh, A. Roebel, and X. Rodet, "Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1116–1126, 2010.

[17] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[18] G. Reis, F. F. De Vega, and A. Ferreira, "Automatic transcription of polyphonic piano music using genetic algorithms, adaptive spectral envelope modeling, and dynamic noise level estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2313–2328, 2012.

[19] C. Raphael, "Automatic transcription of piano music," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2002.

[20] A. P. Klapuri, "Multipitch analysis of polyphonic music and speech signals using an auditory model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 255–266, 2008.

[21] G. E. Poliner and D. P. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, 2007.

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

[23] N. Bertin, R. Badeau, and E. Vincent, "Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 538–549, 2010.

[24] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 2012, pp. 121–124.

[25] T. Virtanen, J. F. Gemmeke, B. Raj, and P. Smaragdis, "Compositional models for audio processing: Uncovering the structure of sound mixtures," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 125–144, 2015.
[26] A. Ozerov, E. Vincent, and F. Bimbot, "A general flexible framework for the handling of prior information in audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1118–1133, 2012.
[27] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003, pp. 177–180.
[28] B. Fuentes, R. Badeau, and G. Richard, "Adaptive harmonic time-frequency decomposition of audio using shift-invariant PLCA," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 401–404.
[29] M. Goto, "A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication (ISCA Journal)*, vol. 43, no. 4, pp. 311–329, 2004.
[30] H. Kameoka, T. Nishimoto, and S. Sagayama, "A multipitch analyzer based on harmonic temporal structured clustering," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 982–994, 2007.
[31] G. Grindlay and D. P. Ellis, "Multi-voice polyphonic music transcription using eigeninstruments," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WAASPA)*, 2009, pp. 53–56.
[32] E. Benetos, S. Ewert, and T. Weyde, "Automatic transcription of pitched and unpitched sounds from polyphonic music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3107–3111.
[33] S. A. Abdallah and M. D. Plumbley, "Unsupervised analysis of polyphonic music by sparse coding," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 179–196, 2006.
[34] K. O'Hanlon, H. Nagano, and M. D. Plumbley, "Structured sparsity for automatic music transcription," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012, pp. 441–444.
[35] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, Grenada, Spain, 2004, pp. 494–499.
[36] E. Benetos and S. Dixon, "Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model," *Journal of the Acoustical Society of America*, vol. 133, no. 3, pp. 1727–1741, 2013.
[37] W. Jesteadt, C. C. Wier, and D. M. Green, "Intensity discrimination as a function of frequency and sensation level," *Journal of the Acoustical Society of America*, vol. 61, no. 1, pp. 169–177, 1977.
[38] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.
[39] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1066–1074, 2007.
[40] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Recent developments in total variation image restoration," *Mathematical Models of Computer Vision*, vol. 17, 2005.
[41] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[42] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
[43] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.
[44] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.
[45] J. Eckstein and D. P. Bertsekas, "On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1-3, pp. 293–318, 1992.
[46] A. Nedic and A. Ozdaglar, "Cooperative distributed multi-agent optimization," *Convex Optimization in Signal Processing and Communications*, vol. 340, 2010.
[47] X. Zhang, M. Burger, and S. Osher, "A unified primal-dual algorithm framework based on bregman iteration," *Journal of Scientific Computing*, vol. 46, no. 1, pp. 20–46, 2011.
[48] S. Ewert, "Technical annex and extensions: Alternating directions framework for piano transcription," Tech. Rep., 2016.
[49] R. T. Rockafellar, "Convex analysis," *Princeton Mathematical Series*, vol. 46, 1970.
[50] D. FitzGerald, M. Cranitch, and E. Coyle, "Extended nonnegative tensor factorisation models for musical sound source separation (article id 872425)," *Computational Intelligence and Neuroscience*, vol. 2008, 2008.
[51] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 708–716, 2000.
[52] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010.
[53] J. J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Cañadas-Quesada, "Musical instrument sound multi-excitation model for non-negative spectrogram factorization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1144–1158, 2011.
[54] A. P. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2006, pp. 216–221.
[55] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010.
[56] S. Sigtia, E. Benetos, S. Cherla, T. Weyde, A. S. d. Garcez, and S. Dixon, "An RNN-based music language model for improving automatic music transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 53–58.

Sebastian Ewert received the M.Sc./Diplom and Ph.D. degrees (summa cum laude) in computer science from the University of Bonn (svd. Max-Planck-Institute for Informatics), Germany, in 2007 and 2012, respectively. After a postdoc at the Centre for Digital Music, Queen Mary University of London (United Kingdom), he became lecturer for signal processing in the centre in 2015. Currently, he is additionally holding a research position in the EPSRC programme Fusing Audio and Semantic Technologies (FAST) and is one of the leaders of the Machine Listening Lab.

Mark Sandler (PhD FAES FIET FIEEE CEng) is Founding Director of the Centre for Digital Music, a world-leading research group in audio and music technology with over 80 members. The Centre is in the School of Electronic Engineering & Computer Science, where he holds the chair in Signal Processing. He is also Director of the Centre for Doctoral Training in Media and Arts Technology, a UK government funded special PhD training programme. He is a recipient of the Royal Society Wolfson Research Merit Award (2015-19). He has published over 400 papers in conferences and journals.