IT'S TIME FOR A SONG – TRANSCRIBING RECORDINGS OF BELL-PLAYING CLOCKS

Matija Marolt
University of Ljubljana
Faculty of Computer and Information Science
matija.marolt@fri.uni-lj.si

Marieke Lefeber
Meertens Instituut and Museum Speelklok
marieke.lefeber@meertens.knaw.nl

ABSTRACT

The paper presents an algorithm for automatic transcription of recordings of bell-playing clocks. Bell-playing clocks are clocks containing a hidden bell-playing mechanism that is periodically activated to play a melody. Clocks from the eighteenth century give us unique insight into the musical taste of their owners, so we are interested in studying their repertoire and performances - thus the need for automatic transcription. In the paper, we first present an analysis of acoustical properties of bells found in bell-playing clocks. We propose a model that describes positions of bell partials and an algorithm that discovers the number of bells and positions of their partials in a given recording. To transcribe a recording, we developed a probabilistic method that maximizes the joint probability of a note sequence given the recording and positions of bell partials. Finally, we evaluate our algorithms on a set of recordings of bell-playing clocks.

1. INTRODUCTION

Bell-playing clocks are clocks containing a hidden bell-playing mechanism, which is activated every hour, every half an hour or even every quarter of an hour to play a melody (see Figure 1). To make this happen, the going train activates the musical train, which starts the rotation of the musical cylinder. The cylinder contains a pattern of pins which 'play' a series of keys as the cylinder rotates. Through threads these keys are connected to hammers, which strike the bells. Such bell-playing mechanisms are usually part of longcase- or bracket clocks.

Bell-playing clocks probably originate from carillons which played their melodies already in the thirteenth century in the towns of the Low Countries. From the end of the fifteenth century these instruments were also made for domestic use, but they were unique pieces, only affordable for the very rich. From the end of the seventeenth century, bell-playing clocks became more and more popular, although they still remained a status symbol, only affordable for the rich elite. Many eighteenth-century bell-playing clocks have been preserved. Clock restorer Melgert Spaander from Zutphen (Netherlands) restored and recorded over 150 of these clocks and made these recordings available for our researches. The collection consists of approximately 1500 melodies, which offer us, in a way, recordings from the eighteenth century. We are studying the repertoire of these clocks and also the performances of melodies with the aim of gaining more insight into the musical taste of the eighteenth-century elite. In order to study the repertoire of clocks, we need to transcribe all of the recorded melodies, so that they can be analyzed. Transcribing these melodies by hand requires a lot of practice and is made even more difficult by the inharmonicity and long decay times of bell sounds.

Figure 1. Melodies of a bell-playing clock.

In this paper, we present an algorithm for automatic transcription of recordings of bell-playing clocks. Automatic music transcription is a difficult problem to solve, although methods are improving constantly; Klapuri and Davy provide an extensive overview of the current state of the art [1]. Because we know little of the acoustical properties of clock bells, we could use unsupervised learning techniques for transcription. Such techniques have been used previously by several authors: Abdallah and Plumbley [2] used sparse coding for transcription of synthesized harpsichord music, while Virtanen used it to transcribe drums [3]. A number of authors use variants of non-negative matrix factorization to transcribe polyphonic music [4-7]. Their methods, however, were devised for music composed of harmonic sounds and are thus difficult to apply to our domain. Recently, Marolt [8] proposed to use non-negative matrix factorization with selective sparsity constraints to transcribe recordings of church bells.

While we initially experimented with unsupervised learning techniques, we obtained better results with the method proposed in this paper. We propose a two step approach to transcription: first, we present an analysis of acoustical properties of bell sounds and derive an algorithm that discovers the number of bells and positions of their partials in a given recording. A probabilistic method that relies on analysis of the recorded signal, the found bell partials, and on some higher-level musical knowledge is used to perform the transcription. We evaluate our approach on a collection of recordings of bell-playing clocks.

2. IDENTIFYING BELLS IN A RECORDING

2.1 Modeling Positions of Bell Partials

The shape or profile of a bell determines the relative frequencies of its vibrations. Bells have distinct but inharmonic partials – a partial being a frequency of vibration present in the sound of a bell. Little is known of the acoustical properties of bells used in clocks, so we first conducted a study to determine whether we can model the positions of bell partials. To estimate the properties of clock bells, we analyzed a set of 10 recordings of different clocks, containing a total of 88 bell sounds and manually annotated positions of their partials. Bells were found to have six strong partials; their positions relative to the perceived pitch (in cents) are listed in Table 1.

Mean and st. dev. of partial - pitch freq. (cents)	Mean magnitude relative to the strongest partial (dB)
0 ± 0	-21.8
1485 ± 81	-11.4
2433 ± 111	-4.8
3145 ± 124	-8.6
3719 ± 123	-13.0
4236 ± 91	-17.9

Table 1. Means and std. deviations of relative partial positions (in cents) for the analyzed bells. Mean magnitude of each partial relative to the strongest partial (in dB) is also shown

We can observe that partials are centered at approximately 2.4, 4, 6, 8.5 and 11.5 times the fundamental frequency (in Hz), the 3th and 4th partial being the loudest. The fundamental frequency corresponds to the perceived pitch. When studying relationships between these partials, we discovered a regularity, not unlike what Hibbert [9] discovered for church bells, namely that relationships between relative positions of partials (i.e. logarithmic frequency ratios) are linear. Figure 2 shows scatter plots of relative positions of partials 2, 4, 5 and 6 versus the relative position of the third partial - the linearity is obvious. This enables us to fit a linear regression model (also shown in Figure 2) that can be used to predict the positions of all six bell partials, if we know the positions of two of them, with an average error below 20 cents.

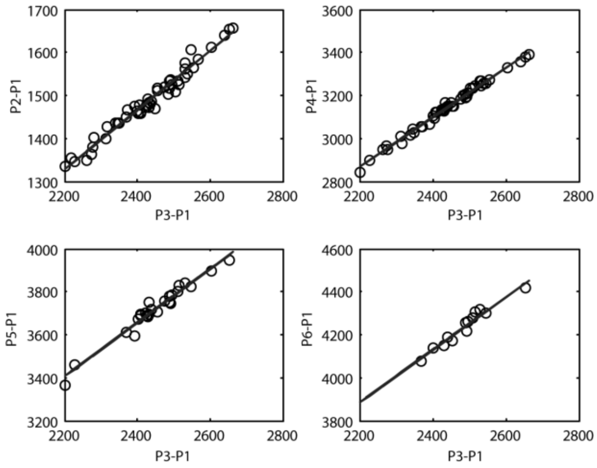

Figure 2. Linearity of relative partial positions.

Formally, a model that defines positions of bell partials given positions of two partials is expressed as a sum of Gaussians:

$$M_{f1,f2}(x) = \sum_{k=1}^{6} \exp\left(-\frac{(x - r_{f1,f2}^{k})^2}{2\sigma^2}\right), \quad (1)$$

where f_1 and f_2 are positions of any two bell partials, $r_{f1,f2}^{k}$ the k-th partial position as calculated by the regression fit and σ the allowed deviation from the regression fit.

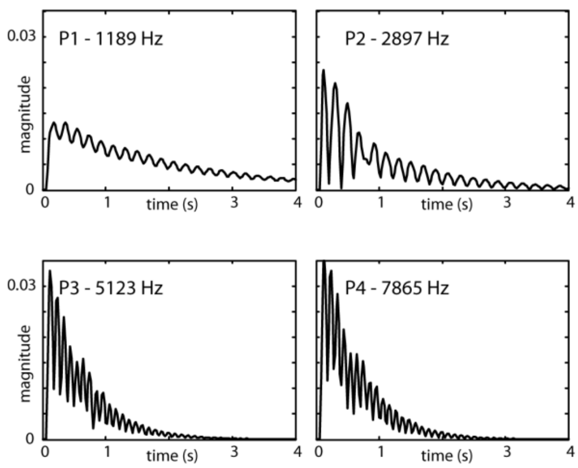

Figure 3. Time evolution of the first four partials of a clock bell.

Time evolution of partials follows an exponential decay curve, and is faster for higher partials, which on the other hand, are initially louder. Individual partials frequently exhibit beating, also evident in Figure 3. Beating is caused by the so-called doublets, which arise when bells are not symmetrical about a vertical axis through their centers. This asymmetry causes most of the vibrational modes in bells to split into two distinct modes with slightly different frequencies that beat against each other. Beating is problematic when we try to estimate the onsets and time evolution of partials, so we try to remove its effect, as described in section 3.

2.2 Bells? What Bells?

When analyzing a recording of a bell-playing clock, we initially have no information on the number of bells involved, their tuning or positions of their partials. In this section, we introduce an algorithm that uses the bell partial model presented in section 2.1 to estimate the number of bells in a recording and positions of their partials. The algorithm is based on the observation that bells are rarely struck at the same time, which is mainly due to the inharmonic nature of bell sounds and their imperfect tuning. Therefore, we can use the "common fate" auditory grouping principle, especially onset synchrony to find groups of partials that belong to individual bells.

We first calculate a magnitude spectrogram **F** of a recording. To reduce variance in partial magnitudes in different frequency regions, we multiply the spectrogram with a perceptual weighting model, as introduced by Vincent [10]. Weights are calculated on an average spectrum and applied globally to yield a flattened time-frequency representation $\mathbf{F_w}$. Such flattening "amplifies" partials with small magnitudes, which makes it easier for the algorithm to consider those partials in the process of finding partial groups. This is especially important, because magnitude of the fundamental frequency of a bell lies over 20 dB below its loudest partial (see Table 1). As the fundamental corresponds to pitch, we need to accurately estimate it, otherwise the pitch of a bell can only be approximated.

Bells have sharp onsets and long decay times, so the next step of the algorithm accentuates the fast positive changes (sharp onsets) in the magnitude spectrogram. The dynamics of changes within frequency bins of $\mathbf{F_w}$ is estimated by calculating first order delta coefficients **D** of the bins with a sliding window of length N_d. Delta coefficients provide estimates of the gross shape of short time segments of the frequency bins. They emphasize fast and big changes, such as onsets, and deemphasize slower and smaller changes, such as beating. This is illustrated in Figure 4 that displays delta coefficients of a bell partial calculated on a recording of a bell-playing clock.

To discover groups of partials with synchronous onsets, we calculate covariances of their delta coefficients:

$$c_{ij} = \max\left(\frac{1}{n-1} \sum_{k=1}^{n} (d_{ik} - \mu_i)(d_{jk} - \mu_j), 0 \right) \quad , \quad (2)$$

where d_{ij} represents an element of the delta spectrogram **D** and μ_i the average of the i-th row of **D**. Because delta coefficients emphasize onsets, the value c_{ij} represents a measure of onset synchrony of partials with frequencies corresponding to bins i and j. Bells do not share many partials and are rarely struck at the same time, so a bell's partial will have high synchrony with other partials of the same bell, but not with partials of other bells.

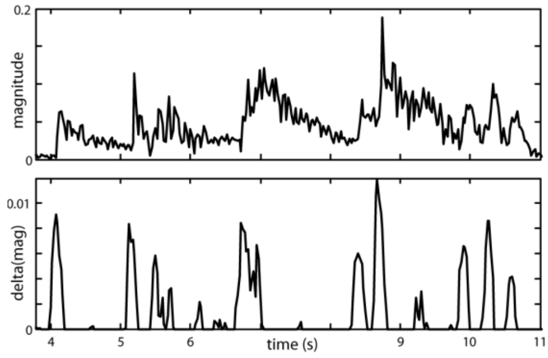

Figure 4. Amplitude envelope and delta coefficients of a bell partial from a bell-playing clock recording

A global covariance matrix **C** could be calculated on the entire delta spectrogram **D**, but we found that this puts too much emphasis on bells that occur frequently in a recording and fails to find partials groups of other less frequent bells. We therefore calculate local covariance matrices on all segments that are obtained by sliding a window of length n over the delta spectrogram **D** with a step size of $n/2$. This results in a set of local covariance matrices $\mathbf{C}^{(t)}$. The overall measure of onset synchrony of a partial i is then calculated by weighting the contributions of local covariance matrices with the overall energy of the partial in each segment, as approximated by $c_{ii}^{(t)}$:

$$s_{ij} = \frac{1}{\sum_{t=1}^{m} c_{ii}^{(t)}} \sum_{t=1}^{m} c_{ii}^{(t)} c_{ij}^{(t)} \quad . \quad (3)$$

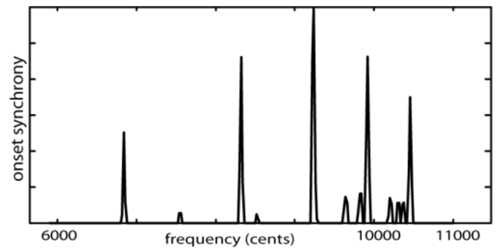

Figure 5. Onset synchrony of a partial at 6840 cents. Four other partials from the same bell (8320, 9240, 9920 and 10460 cents) are clearly visible.

Figure 5 displays one row of the resulting matrix **S**, representing onset synchrony of a partial in a bell clock recording. A group of five partials belonging to the same bell sound clearly stands out.

To discover groups of synchronous partials, we analyze each row $\mathbf{s}_i$ of the matrix **S**, and search for parameters of the bell model presented in section 2.1 that best describe $\mathbf{s}_i$. Specifically, for each row $\mathbf{s}_i$ we find model parameters f_1 and f_2 that maximize:

$$\arg\max_{f1,f2} M_{f1,f2} \cdot \mathbf{s}_i \quad , \qquad (4)$$

where · denotes the dot product operator. Due to the sparseness of **S** (see Figure 5), an exhaustive search for optimal parameters can be performed efficiently. If the dot product in eq. (4) exceeds a preset threshold T, the model $M_{f1,f2}$ is considered to represent one of the bells in the analyzed recording. The actual positions of bell's partials may deviate from the model, so we estimate them from $\mathbf{s}_i$ by simple component-wise multiplication:

$$\mathbf{b} = (M_{f1,f2} \bullet \mathbf{s}_i)^{\frac{1}{K}} \ , \qquad (5)$$

where K is used to compress partial magnitudes.

The final outcome of the algorithm is a set of vectors $\mathbf{b} \in \mathfrak{B}$ describing the sounds of bells in a recording. We present an evaluation of the algorithm in section 4.

3. TRANSCRIPTION

To transcribe a recording of a bell-playing clock, we need to find which notes (bells) were played and when they were played - their onset times. Although this may seem to be an easy task given positions of partials of bell sounds, the task is complicated due to several factors. First, partials interact; they get amplified, cancelled or beat against each other, which makes it difficult to follow their amplitude envelopes and find their onsets. Decay times of bell sounds are long and although bells are usually not played at the same time, the number of concurrently sounding bells (polyphony) is always high. Partials decay at different rates, so the spectrum of bells changes with time. Recordings contain fast passages with inter-onset times of less than 100ms, as well as embellishments such as grace notes and arpeggios that further complicate transcription. And last, these are not synthetic recordings, nor are they very professionally made; they contain many noisy artefacts, such as background noise, noises coming from the clock mechanism or similar.

We chose to take a probabilistic approach to transcription and search for the most probable sequence of notes in a recording. The transcription process starts by calculating the onset times and onset probabilities of bells. We use the complex domain onset detection function and peak picking algorithm [11], which performs well with bell sounds, because of their sharp percussive onsets. Onset probabilities are calculated from the value of the onset detection function at each onset.

Given N onset times and the fact that bells are seldom struck at the same time, transcription can be viewed as a problem of finding a sequence of notes and rests $s_1,s_2,s_3...s_N$ that best describes the analyzed signal; s_1 starts at the first found onset, s_2 at the second and so on. s may represent a note (all notes n_i, i=1..M are described by their corresponding bell models from the set $\mathfrak{B}$); or may be a rest (r). Specifically, we wish to find a sequence of notes and rests that maximizes the joint probability:

$$P(s_1)P(s_2 \mid s_1)P(s_3 \mid s_2) \cdot \ldots \cdot P(s_N \mid s_{N-1}) \ . \qquad (6)$$

To estimate probability of a note $P(s_i=n_j)$, we take two factors into consideration: the probability that note n_j described by the corresponding bell model $\mathbf{b}_j$ actually occurred in the signal at onset i, and the probability of that onset. Note probability is calculated by multiplying the bell model with the time-frequency representation **D** (as defined in section 2.1). Onset probability is proportional to the value of the onset detection function at the onset. We can thus write the probability of a note n_j occurring at onset i as:

$$P(s_i = n_j) = P(o_i)\tfrac{1}{C_i}\mathbf{b}_j \cdot \mathbf{d}_i \ . \qquad (7)$$

where · denotes dot product, $\mathbf{d}_i$ represents the time-frequency representation **D** at time i and P(o_i) the probability of an onset at that time. C_i is a scaling factor used to normalize the dot product to a [0-1] range.

Probability of a rest is defined as:

$$P(s_i = r) = (1 - P(o_i))\prod_{k=1}^{M}\left(1 - P(s_i = n_k)\right), \qquad (8)$$

thus if no notes are likely to occur and the onset is also not likely, a rest will be likely.

To define conditional probabilities $P(s_i \mid s_{i-1})$, we introduced two changes to the above expressions. First, if note n_k occurred at time i-1, we subtract the note from the time-frequency representation **D**, thus eliminating its effect at time i:

$$\mathbf{d}_i(s_{i-1}{=}n_k) = \max(\mathbf{d}_i - N(i-1,\sigma)\,\mathbf{b}_k \bullet \mathbf{d}_{i-1},\ 0) \ . \qquad (9)$$

Operator • denotes component-wise multiplication and N the unscaled normal distribution, which models the time evolution of delta coefficients. As we can observe in Figure 4, delta coefficients are roughly bell-shaped at onsets, so we approximate them with a normal distribution. If s_{i-1} is a rest, nothing is subtracted, so $\mathbf{d}_i(s_{i-1}=r)=\mathbf{d}_i$.

As intervals between adjacent notes in a melody are usually small (a phenomenon also known as pitch proximity), we include an additional factor into $P(s_i \mid s_{i-1})$. Pitch proximity is modeled by a proximity profile $\mathbf{R}(n)$, which as in [12], is represented by a normal distribution centered around a given pitch n, indicating pitch probabilities of the following note. The obtained conditional probability of consecutive notes can thus be written as:

$$P(s_i{=}n_j \mid s_{i-1}{=}n_k) = P(o_i)R_i(n_k)\tfrac{1}{C_i}\mathbf{b}_j \cdot \mathbf{d}_i(s_{i-1}{=}n_k) \ . \qquad (10)$$

If s_i is a rest, we can calculate the conditional probability with the expression given in eq. (8), whereby we replace note probabilities with conditional probabilities and multiply the expression with a constant representing the proximity profile.

The most likely sequence of notes and rests can be efficiently estimated with dynamic programming; the resulting set of onset times and notes represents the transcription of a recording.

4. EVALUATION

In order to evaluate our algorithm, we manually transcribed and annotated positions of partials in a set of 10 recordings of different bell-playing clocks. Results and discussion are given in the following sections.

4.1 The Bell-finding Algorithm

We used the following parameters to test the bell-finding algoritm: the magnitude spectrogram was calculated with the Constant-Q transform [13], using a maximum window size of 100ms, a step size of 25 ms and 20 cent spacing between adjacent frequency bins. The deltas were calculated with a sliding window of N_d=9 frames, the covariance matrices on n=100 frames long segments. Finally, the threshold T that determines whether a bell model should be included in the final results was set to $1/20^{th}$ of the maximum value of all models and the compression coefficient K to 10.

For comparison, we also developed an alternative approach for estimating partials in bell sounds. We used non-negative matrix factorization (NMF) to factorize the delta magnitude spectrogram **D** into matrices **W** and **H**, where the basis vectors in **W** would ideally correspond to bell spectra and **H** would explain how bell magnitudes change over time. Several efficient implementations of NMF exist in the literature; in our experiments we used the SNMF algorithm introduced by Kim and Park [14]. The algorithm is based on the alternating non-negativity constrained least squares and active set method and allows to impose sparsity constraints on **H** or **W**.

With NFM learning, the number of basis vectors is fixed and we need to set it in advance, prior to the actual learning. Because in our case basis vectors correspond to spectra of individual bells, we need to know the number of bells in a recording prior to learning. This is not usually the case, but to perform the comparison of both approaches, we give the NMF algorithm a small "advantage" by setting the number of basis vectors to the actual number of bells as was manually annotated for each recording.

	precision	recall
proposed algorithm	0.94	0.98
SNMF	0.87	0.87

Table 2. Comparison of two bell finding algorithms

Table 2 shows average precision and recall scores of the two algorithms on all recordings. Although both perform well, the proposed approach outperforms non-negative matrix factorization. We contribute the difference to two main reasons. To find partials of bell sounds, the proposed algorithm uses a local approach; namely covariance matrices are calculated on short segments of the entire recording and then combined based on magnitudes of the analyzed partials in these segments. On the other hand, NMF works globally by iteratively minimizing the factorization error. The difference is important when searching for bells that are not frequently played. NMF will tend to ignore them and rather focus on minimizing the overall error which may lie in varying decay times of bell partials or noise. The local nature of our approach will not fail for such cases, as the bells will stand out in individual local segments and will consequently also show in the global matrix **S**. Another advantage of the proposed approach is its use of the knowledge provided by the model of bell partial positions, as presented in section 2.1. Namely, the search for bell partials is limited by the model, so only regions of the signal that correspond to predicted partial frequencies are considered. Therefore, noise, either background or made by the clock mechanism or other external factors, can largely be ignored. NMF uses no such high-level knowledge, so it is affected by noise on all levels, as it tries to accommodate it and include it into the basis vectors.

All of the false negative errors (missed bells) made by our bell finding algorithm were bells a semitone apart from another more dominant bell, with most of their partials overlapping. Such bells are mainly used as embellishments and were ignored because their onsets were masked by the more dominant bell. However, since these bells are not very frequently played, these errors do not have a large influence on overall transcription accuracy, as we show in the following section.

4.2 The Transcription Algorithm

To evaluate how various choices made when designing our transcription algorithm influence its performance, we tested several variants of the algorithm: A – the described algorithm, B – excluding the pitch proximity profile, thus making all note transitions equally probable, C – excluding note subtraction, thus avoiding conditional note probabilities and D – using annotated onsets instead of the calculated ones and E – using annotated bell partials instead of the calculated ones. Average precision and recall scores of transcriptions of all recordings are shown in Table 3.

As we can observe, differences between these variants are not very big. This is due to the fact that the differences mostly affect "problematic" parts of recordings that include fast passages and embellishments, while elsewhere the combination of the delta magnitude spectrogram, accurately estimated positions of bell partials, and correctly found onsets makes them irrelevant.

For the problematic parts, the pitch proximity profile that favors smaller intervals (B) and especially note subtraction (C), which mostly prevents repetitions of predominant notes, do have a positive effect on performance. As (*D*) shows, approx. half of the missed notes are caused by missed onsets and recall is raised by approx. 0.05 if perfect onset detection is used. On the other hand, nothing is gained by using the manually annotated bell partials, so the bell finding algorithm seems to be working very well and the errors it made seem to be almost irrelevant.

	transcription	
	precision	recall
A: proposed algorithm	**0.95**	**0.89**
B: no proximity	0.94	0.87
C: no conditional prob.	0.91	0.88
D: perfect onsets	0.94	0.94
E: perfect bell models	0.95	0.89

Table 3. Comparison of variants of the transcription algorithm

Most of the errors, either missed notes (false negatives) or extraneous notes (false positives) are made in fast passages, where note repetitions are missed, notes are transcribed in an incorrect order or weak onsets ignored. Overall, the performance is good enough, so that transcriptions will be used for further analysis and included in a searchable database of melodies; in fact when analyzing the errors, we discovered that several errors were in the ground truth and not in the calculated transcriptions.

5. CONCLUSION

The proposed approach to transcription of bell-playing clock recordings is a good first step towards analysis of these recordings. The bell-finding and transcription algorithms perform well and will be used to transcribe the entire collection of recordings of bell-playing clocks. We will add the resulting transcriptions to a searchable database of melodies, thus making them available to interested researchers for further analysis. There is room for improvements of the algorithm; we plan to consider ways of allowing for correct treatment of simultaneous notes, as well as to test the algorithm on other recordings containing bell sounds.

Acknowledgments. This work was supported in part by the Slovenian Government-Founded R&D project EthnoCatalogue: creating semantic descriptions of Slovene folk song and music.

6. REFERENCES

[1] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. New York: Springer, 2006, p.^pp. Pages.

[2] S. A. Abdallah and M. D. Plumbley, "Unsupervised analysis of polyphonic music by sparse coding," *Neural Networks, IEEE Transactions on*, vol. 17, pp. 179-196, 2006.

[3] T. Virtanen, "Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, pp. 1066-1074, 2007.

[4] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, 2003, pp. 177-180.

[5] E. Vincent, *et al.*, "Harmonic and inharmonic Nonnegative Matrix Factorization for Polyphonic Pitch transcription," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 2008, pp. 109-112.

[6] S. A. Raczynski, *et al.*, "Multipitch Analysis with Harmonic Nonnegative Matrix Approximation," in *ISMIR 2007, 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007, pp. 381-386.

[7] B. Niedermayer, "Non-negative Matrix Division for the Automatic Transcription of Polyphonic Music," in *ISMIR 2008, 9th International Conference on Music Information Retrieval*, Philadelphia, USA, 2008, pp. 544-545.

[8] M. Marolt, "Non-negative matrix factorization with selective sparsity constraints for transcription of bell chiming recordings," in *Sound and Music Computing Conference*, Porto, Portugal, 2009, pp. 137-142.

[9] W. A. Hibbert, "The Quantification of Strike Pitch and Pitch Shifts in Church Bells," Ph.D., Faculty of Mathematics, Computing and Technology, The Open University, Milton Keynes, UK, 2008.

[10] E. Vincent and M. D. Plumbley, "Low Bit-Rate Object Coding of Musical Audio Using Bayesian Harmonic Models," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, pp. 1273-1282, 2007.

[11] J. P. Bello, *et al.*, "On the use of phase and energy for musical onset detection in the complex domain," *Signal Processing Letters, IEEE*, vol. 11, pp. 553-556, 2004.

[12] D. Temperley, *Music and Probability*: The MIT Press, 2007.

[13] J. C. Brown, "CALCULATION OF A CONSTANT-Q SPECTRAL TRANSFORM," *Journal of the Acoustical Society of America*, vol. 89, pp. 425-434, Jan 1991.

[14] H. Kim and H. Park, "Nonnegative Matrix Factorization Based on Alternating Nonnegativity Constrained Least Squares and Active Set Method," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, pp. 713-730, 2008.