SINGER MELODY EXTRACTION IN POLYPHONIC SIGNALS USING SOURCE SEPARATION METHODS

Jean-Louis Durrieu, Gaël Richard and Bertrand David

TELECOM ParisTech / CNRS LTCI
46, rue Barrault - 75634 Paris Cedex 13 - France
durrieu@enst.fr

ABSTRACT

We propose a new approach for singer melody extraction, based on blind source separation techniques. The short time Fourier transform (STFT) of the singer signal is modelled by a Gaussian mixture model (GMM) explicitly coupled with a generative source/filter model. We then introduce a simplification of this general GMM and approximate the STFT of the music signal using Non-negative Matrix Factorization (NMF) techniques. The melody line is extracted from the explicit source component of the model thanks to a Viterbi algorithm. The results are very promising and comparable or better than those of state-of-the-art systems.

Index Terms— Music, Source/Filter Model, Blind Source Separation, Spectral Analysis, Non-Negative Matrix Factorization

1. INTRODUCTION

When listening to a song, a human listener can easily focus on the melody sung and separate it from the background music. There is a growing effort in the community to provide a machine with the capabilities to perform singer melody extraction from polyphonic music, i.e. to transcribe the sequence of notes sung by a human performer on accompanying background music. In fact it is especially useful in applications such as Query-By-Humming, where it could automatically generate the melody database out of the original song database, without the need of having them encoded as MIDI files. It could also provide a natural way to compare songs and identify cover-versions. Furthermore, the separation of the lead vocal part opens the path for Demix/Remix and Karaoké applications.

Traditional methods for melody extraction from polyphonic mixtures are based upon a multipitch estimation followed by a melody tracker that finds the most probable melodic line (e.g. [4] or [7] for an overview of ISMIR evaluations).

An alternative approach would rely on a prior source separation step. For example in [6], such a separation is performed by means of an adaptive Wiener filter. More precisely, the authors model the short term Fourier transform (STFT) $X(f,t)$ of the signal $x(t)$ as the sum of two spectra: $V(f,t)$ for the singer's voice and $M(f,t)$ for the background music where each of them is characterized by a Gaussian Mixture Model (GMM). The estimate of the desired voice (singer or music) is then computed thanks to an adaptive Wiener filter applied to the original STFT. Although this approach obtains satisfying results, it suffers from two major limitations: firstly, the GMM does not permit to easily take into account the specific characteristics of music (multiple sources, multiple notes) at a reasonable complexity; secondly, the model of singing voice is relatively crude and does not include any dependence of its timbre with the fundamental frequency.

Considering these limitations, we propose a new approach which includes a source/filter model of the singing voice and a model for the background music that is derived from [1] that can be efficiently estimated by means of Non-negative Matrix Factorization (NMF) techniques. In this framework, we estimate the fundamental frequency of the singing melody.

The paper is organised as follows: in section 2, we introduce the signal model. The parameter estimation is then described in section 3. We give some results, and compare the performance of our system to state-of-the art results obtained during an international evaluation (MIREX06's audio melody extraction task). Finally, we suggest some conclusions.

2. SIGNAL MODEL

The signal $x(t)$ is considered as the mixture of the singer voice signal $v(t)$ and the background music $m(t)$: $x(t) = v(t) + m(t)$. We also assume that these signals are independent. This does not really hold in the case of songs, where the singer is in accordance with the background music, creating a certain coupling between the signals. This is a simplifying assumption. Since we consider signals that are supposed to have rather characteristic spectra, especially with specific spectral envelopes, this hypothesis is not unreasonable.

As stated in [1], we will assume signal spectra to be zero-mean Gaussian, with a diagonal covariance matrix whose diagonal elements are equal to the signal power spectrum density (PSD). This is a reasonable assumption since the signal can be considered quasi-stationary on small analysis windows .

We denote matrices with capital letters, the size of STFT matrices generally is $N \times T$ where N is the number of frequency bins and T the number of analysis frames.

2.1. Singer Voice Model

We propose to include a pitch dependency in the model introduced in [6], by means of a source/filter model. We only consider the pitched part of the vocal signal. Therefore the source will be only characterized by spectral combs. On those pitched parts, the filters usually are related to the pronounced vowels and the corresponding spectral envelopes exhibit the formants, i.e. the resonances of the

This research was partly supported by the European Commission under contract *FP6-027026-K-SPACE* and the French GIP ANR under contract ANR-06-JCJC-0027-01 *DESAM*.
The authors would like to thank Roland Badeau, Associate Professor at Télécom-Paristech, for his helpful comments on this work.

vocal tract. We introduce first the assumed GMM and then the simplified model which we propose.

In the GMM framework, the state space is discretized, one state is defined by the couple (k, f_0) where k is the index of the filter spectral envelope $\{\sigma_k^2(f); f \in [1, N]\}$ and f_0 the index of the source spectral comb $\{\sigma_{f_0}^2(f); f \in [1, N]\}$. Let K be the maximum number of possible vowels, N_{notes} be the number of fundamental frequencies for the source part. These frequencies lie in a range covering the human voice frequencies, here from 100 Hz to 800 Hz. They are discretized such that two successive frequencies are distant of $\frac{1}{8}$ tone. We generate these spectra according to a glottal source model: KLGLOTT88 [5]. For each frame t, the singer voice signal $V_t = [V(1,t) \dots V(N,t)]^T$ knowing the states $k(t) \in [1, K]$ and $f_0(t) \in [1, N_{notes}]$ follows a normal distribution such that:

$$V(f,t)|k(t), f_0(t) \sim \mathcal{N}_c(0, \sigma^2_{k(t),f_0(t)}(f)) \tag{1}$$

where $\sigma^2_{k(t),f_0(t)}(f) = a^2_{k(t)}(t)\sigma^2_{k(t)}(f) \times a^2_{f_0(t)}(t)\sigma^2_{f_0(t)}(f)$ and $\mathcal{N}_c$ is the Gaussian circular distribution [1] . $a^2_k(t)$ and $a^2_{f_0}(t)$ are amplitude factors, so that we can deal with different dynamics without having one state per dynamic.

Finally, let ω_k (resp. ν_{f_0}) be the a priori probabilities of state k (resp. f_0) . The likelihood of the GMM for the singer voice can be written as:

$$p(V_t) = \sum_{k,f_0} \omega_k \nu_{f_0} p(V_t|k, f_0)$$

However, due to the heavy computational load of the parameter estimation using for example an Expectation-Maximization (EM) algorithm [6], we propose below a simpler model which reduces the number of states to one global state. It is then possible to avoid the EM algorithm and find a faster way of estimating the parameters, as shown in section 3. First, let us notice that it is more convenient to rewrite $\sigma^2_{k(t),f_0(t)}(f)$ in equation (1) as:

$$\begin{aligned} \sigma^2_{k(t),f_0(t)}(f) &= \sum_{k,f_0} a^2_k(t)\sigma^2_k(f) a^2_{f_0}(t)\sigma^2_{f_0}(f) 1_{\{k=k(t), f_0=f_0(t)\}} \\ &= \sum_k \tilde{a}^2_k(t)\sigma^2_k(f) \times \sum_{f_0} \tilde{a}^2_{f_0}(t)\sigma^2_{f_0}(f) \end{aligned} \tag{2}$$

where $1_{\{k=k(t),f_0=f_0(t)\}} = 1$ if $k = k(t)$ and $f_0 = f_0(t)$, and 0 otherwise, $\tilde{a}_k(t) = a_k(t)1_{\{k=k(t)\}}$ and $\tilde{a}_{f_0}(t) = a_{f_0}(t)1_{\{f_0=f_0(t)\}}$. Let Σ_K the $N \times K$ matrix such that $\Sigma_K(f,k) = \sigma^2_k(f)$, and similarly $\Sigma_{F_0}(f, f_0) = \sigma^2_{f_0}(f)$, $\tilde{A}_K$ the $K \times T$ matrix such that $\tilde{A}_K(k,t) = \tilde{a}^2_k(t)$ and similarly $\tilde{A}_{F_0}(f_0, t) = \tilde{a}^2_{f_0}(t)$, then:

$$\sigma^2_{k(t),f_0(t)}(f) = \left[(\Sigma_K \tilde{A}_K) .* (\Sigma_{F_0} \tilde{A}_{F_0})\right]_{f,t} \tag{3}$$

where $.*$ denotes the Hadamard product and for a given matrix M, $[M]_{f,t} = M(f,t)$. Equation (3) suggests that we define a new likelihood for $V(f,t)$, which correspond to a one state GMM:

$$V(f,t) \sim \mathcal{N}_c\left(0, [(\Sigma_K A_K) .* (\Sigma_{F_0} A_{F_0})]_{f,t}\right) \tag{4}$$

In fact, the states k and f_0 are not explicit anymore in the underlying model of equation (4). However, since the spectra in Σ_{F_0} are given and fixed, we are able to estimate the matrices A_K and A_{F_0} such that they approximate $\tilde{A}_K$ and $\tilde{A}_{F_0}$. To avoid any ambiguity between the coefficients of A_K and A_{F_0}, the column vectors of A_K are normalized: $a^2_k(t)$ and $a^2_{f_0}(t)$ thus contain information about the presence of the filter k and the pitch f_0 in frame t. In addition, $a^2_{f_0}(t)$ also includes the intensity of the singer signal for frame t. The matrices Σ_K, A_K and A_{F_0} are estimated from the mixture signal thanks to the algorithm developed in section 3.

[1]The Gaussian circular distribution of a complex random variable $z \sim \mathcal{N}_c(0, \sigma^2)$, with $z = \rho e^{i\theta}$, is defined such that: $p(\rho, \theta) = \frac{\rho}{\pi\sigma^2} \exp\left(-\frac{\rho^2}{\sigma^2}\right)$.

2.2. Background Music Model

We consider the background music signal $m(t)$ as being, for each analyzed frame, the instantaneous mixture of R sources $m_r(t)$. Due to the linearity of the Fourier transform (FT), $M(f,t)$, the STFT of m, is also the instantaneous mixture of the R spectra $M_r(f,t)$ of the sources: $M(f,t) = \sum_{r=1}^{R} a_r(t) M_r(f,t)$. In addition, we assume that $M_r(f,t) \sim \mathcal{N}_c(0, \sigma^2_r(f))$. Therefore, our model is equivalent to the following equation:

$$M(f,t) \sim \mathcal{N}_c\left(0, \sum_{r=1}^{R} a^2_r(t)\sigma^2_r(f)\right) \tag{5}$$

Let Σ_R and A_R be the matrices such that $\Sigma_R(f,r) = \sigma^2_r(f)$ and $A_R(r,t) = a^2_r(t)$, the covariance in (5) is then equal to the matrix product $[\Sigma_R A_R]_{f,t}$. In [1], the authors propose a NMF approach to estimate this product. In section 3, we propose to solve this problem in a Maximum Likelihood (ML) framework.

2.3. Mixture Signal

The signal we wish to analyze is the sum of the singer voice signal V and the music signal M. Thanks to the Gaussian hypothesis and from the equations 4 and 5, we can deduce the likelihood for X:

$$X(f,t) \sim \mathcal{N}_c\left(0, D(f,t)\right) \tag{6}$$

where $D = (\Sigma_K A_K) .* (\Sigma_{F_0} A_{F_0}) + \Sigma_R A_R$. All the matrices except Σ_{F_0} must be estimated. Depending on the approach, one can either learn Σ_R from non-vocal recordings or estimate them directly from the mixture. We found that the latter gave better results in our tests.

3. ESTIMATION OF THE PARAMETERS

In this section, we propose a method to estimate Σ_K, A_K, A_{F_0}, Σ_R and A_R. Let $\theta = \{\Sigma_K, A_K, A_{F_0}, \Sigma_R, A_R\}$ be the set of these parameters. Σ_{F_0} is given as explained in section 2.1.

3.1. Maximum Likelihood Criterion

We consider the ML estimation of the parameters, i.e. finding $\hat{\theta}$ such that:

$$\mathrm{p}_{\hat{\theta}}(X) = \max_{\theta} \mathrm{p}_{\theta}(X)$$

where $\mathrm{p}_{\theta}(X)$ is the likelihood of the observations X knowing the parameters θ of the model. By taking the opposite of the log-likelihood of X knowing the parameters θ, and removing the elements that do not depend on the parameters to estimate, we obtain the following cost function we want to minimize:

$$C(\theta) = \sum_{f,t} \log\left(D(f,t)\right) + \frac{|X(f,t)|^2}{D(f,t)} \tag{7}$$

3.2. An Iterative Algorithm

The proposed method is inspired by the work of [3]. We followed a similar process to obtain multiplicative updating rules for the parameters of θ, i.e. at iteration n, from a certain set $\theta^{(n-1)}$, we compute $\theta^{(n)}$ such that each element of the latter is derived from the element of the former. More precisely, $\sigma_k^2(f)^{(n)}$ for instance is obtained from the previously estimated $\sigma_k^2(f)^{(n-1)}$ by an operation of the following type: $\sigma_k^2(f)^{(n)} = \alpha\sigma_k^2(f)^{(n-1)}$, where α depends on the partial derivative of C with respect to the variable $\sigma_k^2(f)$ at the point $\theta^{(n-1)}$. In fact the partial derivatives of C can be expressed under the following form, e.g. with respect to $\sigma_k^2(f)$: $\frac{\partial C}{\partial \sigma_k^2(f)} = P_+ - P_-$, where P_+ and P_- are positive. In this case, with $\alpha = \frac{P_-}{P_+}$, the updated parameter evolves towards the direction of descent for C. We do not give a formal proof of convergence for the proposed algorithm, but a general proof for this kind of methods is given in [3].

We also choose this multiplicative gradient method because the obtained partial derivatives do not allow to analytically separate the desired parameter from the others. Following this scheme, we obtain the following updating rules for the different parameters of the parameter set θ, where $S = |X|.^2$, and using Matlab notations:

$$
\begin{aligned}
P_{F_0} &= S. * (\Sigma_K A_K)./D.^2 \\
Q_{F_0} &= (\Sigma_K A_K)./D \\
A_{F_0} &\leftarrow A_{F_0}. * (\Sigma_{F_0}^T P_{F_0})./(\Sigma_{F_0}^T Q_{F_0}) \\
P_K &= S. * (\Sigma_{F_0} A_{F_0})./D.^2 \\
Q_K &= (\Sigma_{F_0} A_{F_0})./D \\
A_K &\leftarrow A_K. * (\Sigma_K^T P_K)./(\Sigma_K^T Q_K) \\
\Sigma_K &\leftarrow \Sigma_K. * (P_K A_K^T)./(Q_K A_K^T) \\
A_R &\leftarrow A_R. * [\Sigma_R^T(S./D.^2)]./[\Sigma_R^T(1./D)] \\
\Sigma_R &\leftarrow \Sigma_R. * [(S./D.^2)A_R^T]./[(1./D)A_R^T]
\end{aligned}
$$

Only one of the five parameter matrices A_{F_0}, A_K, A_R, Σ_K and Σ_R is updated at each iteration in this order. It is an abritrary order and may not be optimal in all cases.

We noticed that in our simulations, the criterion first decreases rather fast and stabilizes after about 50 iterations.

3.3. Extracting the Desired Pitch Sequence

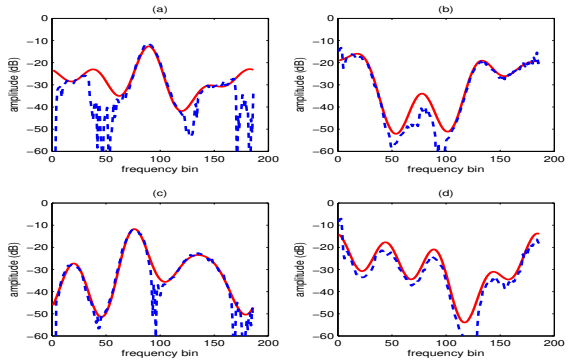
In the GMM framework, the fundamental frequency sequence $\{F_0(t), t \in [1,T]\}$ can be directly obtained by a maximum a posteriori decision rule. However, we do not obtain these probabilities in our framework . That is why we need to define a post-processing step in order to extract the desired melody.

If the signal exactly follows the GMM model, then all the elements in A_{F_0} are equal to zero except at most one per frame t: $a^2_{\tilde{f}_0}(t)$, which means that $\tilde{f}_0 = F_0(t)$. However, we do not obtain such ideal matrices. We could approximate the desired pitch for frame t by the pitch $\tilde{f}_0$ that maximizes $a^2_{\tilde{f}_0}(t)$. This choice is motivated by the fact that the amplitude factors A_{F_0} are sparse and concentrate around the desired pitches.

We designed an algorithm that smoothens the obtained melody line. It is comparable to the Viterbi algorithm in that it finds the pitch sequence that maximizes a given criterion, built as a trade-off between the amplitude factor $a^2_{f_0}(t)$ obtained for a candidate f_0, and the continuity of the pitch sequence. We use the amplitudes as weights for each pitch: the higher this weight for a given node (f_0, t), the more likely the sequence will go through it. To take into account the continuity of a sung melody, we penalize the transitions between the pitches. We define the following function, weighting the transition from pitch f_1 to pitch f_2 (in Hertz): $q(f_2|f_1) = \exp(-\beta|n_2 - n_1|)$ with $\beta = 10$ and where $n_i = 12(\log_2(f_i) - \log_2(440)) + 69$ is the MIDI note number corresponding to f_i [2] . Thanks to the use of the MIDI note number, the transition weight for a half-tone is always the same, independently from the frequency itself.

For each $t \in [1,T]$, we compute the score $S(f_0, t)$ of the best melody line arriving to each node (f_0, t). As in the Viterbi algorithm, we set $S(f_0, 1) = \log(a^2_{f_0}(1))$ for every f_0. For the node (f_0, t), we define the antecedent $(f_A, t-1)$ such that f_A maximizes the function of f: $S(f, t-1) + q(f_0|f)$. We then define $S(f_0, t) = S(f_A, t-1) + \log(q(f_0|f_A)) + \log(a^2_{f_0}(t))$. Once we reach $t = T$, we find the pitch $F_0(T)$for which $S(f, T)$ is maximum. We then track back all the antecedents and form the melody F_0 such that: $F_0(t) = \text{Antecedent}(F_0(t+1))$.

Fig. 1. Synthetic Data: Original spectral envelopes $\sigma_k^{(0)}$ (solid lines) and corresponding envelopes estimated by the proposed algorithm (dashed lines).



4. RESULTS

We first analyze the performance of the proposed algorithm on synthetic data, in order to evaluate how well the algorithm can estimate the desired parameters, especially the spectral envelopes σ_k and the melody line $\{F_0(t), t \in [1,T]\}$. We also tested the algorithm on real data from the melody extraction contest of ISMIR 2004 and compared the results with those of MIREX06's participants on the same database.

4.1. Synthetic Data

A synthetic $N \times T$ STFT matrix $D^{(0)}$ is generated according to the equation of $D(f,t)$ given in section 2.3. The sampling rate is assumed to be 11025 Hz, the number of bins for the FT is $N_{\text{fft}} = 512$, which corresponds to 46.4 ms temporal windows. We only consider the low frequency part of the STFT, under $f_c = 4000$ Hz, which leads to $N = 186$. We choose $K = 4$. Σ_{F_0} is given by the source model KLGLOTT88. For pitches from 100 to 800, regularly spaced

[2] 440 Hz and 69 are the frequency and MIDI note number of the reference note A4.

Opera Songs	Raw Pitch Acc.	Overall Acc.
Proposed Method	**81.2%**	**70.1%**
Participant 1	39.0%	35.6%
Participant 2	63.0%	64.1%
Participant 3	42.6%	47.3%
Participant 4	64.2%	61.9%
Participant 5	45.9%	46.5%
Vocal Songs	Raw Pitch Acc.	Overall Acc.
Proposed Method	**82.6%**	70.5%
Participant 1	56.9%	48.9%
Participant 2	80.4%	**80.6%**
Participant 3	70.7%	70.1%
Participant 4	81.3%	78.6%
Participant 5	69.7%	65.0%

Table 1. Melody Extraction Results on ISMIR 2004 Database Vocal Songs, Comparison with the Results of MIREX 2006 Participants

every eighth tones, $N_{notes} = 146$. We randomly generate the spectral envelopes $\sigma_k^2(f)^{(0)}$. In order to be consistent with real filters, they are cepstrally smoothed. $A_{F_0}^{(0)}$ is such that there is only one active f_0 per frame t, and such that it corresponds to a chirp, i.e. every element is 0 except on the "diagonal" of the matrix. $A_K^{(0)}$ is also such that only one filter is active per frame, and each filter is active for several successive frames. Although the algorithm does not need and does not integrate temporal constraints, this settinghelps us to analyze the results. We set $R = 0$ for the moment. All the parameters to be estimated are randomly initialized.

Figure 1 shows $\sigma_k^{(0)}$ (solid lines) against the estimated σ_k that are most similar to them. We manually paired them, because the algorithm does not solve the inherent ambiguity on the indexes. The estimates are not exactly the same as the original. This comes from the fact that the information needed to retrieve them might be lacking in $D^{(0)}$, because of the chosen "melody". The spectral combs σ_{f_0} are very sparse in the frequency domain, where the energy essentially concentrates around the harmonics of the fundamental frequency. Therefore, every estimate of σ_k that rely too much on a frequency region of low energy is unreliable. Nevertheless, analyzing A_K shows that for some simulations, when the $\sigma_k^{(0)}$are similar, e.g. representing a common "formant", then the obtained $\sigma_k$can approximate the original envelopes by some linear combinations. The results are still satisfying, because for formant regions, the σ_k are generally well estimated.

Another synthetic STFT matrix is generated in order to check the performance of the pitch tracking step. We estimated the pitch from a solo vocal track thanks to the YIN algorithm [2] and generated a STFT matrix with this melody generating $A_{F_0}^{(0)}$. Several noisy conditions were tested with different values of R and the original melody line was successfully retrieved, except when the signal to noise ratio was becoming too low. This test suggested that under reasonable conditions, the proposed algorithm was able to achieve the intended task, as tests on real data will confirm.

4.2. Real Data

The algorithm is now tested on the ISMIR 2004 database for audio melody extraction [7]. We set $R = 64$. We consider only vocal songs in the database, i.e. 12 songs with 2 musical styles: opera (4 songs) and pop (8 songs). The sampling rate is 44.1 kHz.

We compare the results to those of the participants of the MIREX 2006 melody extraction task, which partly used the ISMIR 2004 database. The metrics considered are described in [7]. The results are summarized in table 1. The first column gives the accuracy of the results on frames that were identified as pitched in the reference files. The second column shows the overall accuracy, i.e. also taking into account the unpitched frames. On the 4 opera song files, our method performed very well. The performance significantly drops when unpitched frames are considered (overall accuracy). This may be explained by the fact that the presence of singing voice is not detected and silences in the melody are replaced by notes of other instruments.

Despite the reduced size of the database, our results are very promising. Since our approach is entirely unsupervised, it is reasonable to think that similar results could be obtained on larger databases.

5. CONCLUSION AND FUTURE WORK

We proposed a new method to estimate the pitch of the sung melody in polyphonic audio recordings. It is inspired by NMF blind source separation methods to which we added a source/filter model in order to fit the singer voice track. The results show that the proposed algorithm is at the state of the art, outperforming all methods proposed for MIREX 2006 in the case of opera style recordings.

We are investigating the potential of this method and how we can generalize it to signals other than human voice on background music. The source/filter model can also fit other instruments, especially wind instruments. In order to get closer to reality, we could also set constraints on the spectral envelopes σ_k such as fitting them to an auto-regressive moving average (ARMA) model. Another criterion to be minimized can also be tried, inspired by [6], where the adaptation of the basis σ_r and σ_k is done thanks to a vocal/non-vocal pre-processing step. The actual source separation performances of our algorithm are also to be evaluated against state-of-the-art methods, and our preliminary tests give promising results for the singer voice separation.

6. REFERENCES

[1] L. Benaroya, L. Donagh, F. Bimbot, and R. Gribonval. Non negative sparse representation for Wiener based source separation with a single sensor. In *Proc. of ICASSP*, volume 6, 2003.

[2] A. de Cheveigne and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *JASA*, 111:1917–1930, 2002.

[3] I. Dhillon and S. Sra. Generalized nonnegative matrix approximations with bregman divergences. In *Proc. of NIPS*, 2005.

[4] M. Goto. Robust predominant-f0 estimation method for real-time detection of melody and bass lines in cd recordings. In *Proc. of ICASSP*, volume 2, pages 757–760, 2000.

[5] D. Klatt and L. Klatt. Analysis, synthesis, and perception of voice quality variations among female and male talkers. *JASA*, 87:820–857, 1990.

[6] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval. Adaptation of Bayesian models for single-channel source separation and its application to voice/music separation in popular songs. *IEEE Trans. on ASLP*, 15:1564–1578, 2007.

[7] G. Poliner, D. Ellis, A. Ehmann, E. Gómez, S. Streich, and B. Ong. Melody transcription from music audio: approaches and evaluation. *IEEE Trans. on ASLP*, 14:1247–1256, 2007.