An HMM Based Singing Transcription System

W.A. Krige and T.R. Niesler

Department of Electric and Electronic Engineering
Stellenbosch University
{wkrige,trn}@dsp.sun.ac.za

Abstract

A singing transcription system transforming acoustic input into midi note sequences is presented. Notes are individually modeled by hidden Markov models (HMM's) using untuned pitch, delta and voicing coefficients as feature vectors. Efficient use of a limited amount of training data is achieved by means of state tying. Explicit transition models are introduced to better identify boundaries between notes that are otherwise poorly modeled, and a non-repetitive grammar introduced to reduce insertions. The system is found to be able to transcribe sung passages with 88.5% accuracy.

1. Introduction

Transcription can be described as the act of translating from one medium to another. Transcription of a musical performance into a text representation is accomplished by means of a set of well defined symbols, designed to capture various characteristics and components of the performance. This translation into *standard music notation* is referred to as a *musical score*. Currently this process requires a skilled music professional and is done by hand.

The integration of computers and music, in terms of education, can be divided into four disciplines: teaching of music fundamentals, music performance evaluation, music analysis and music composition. An overview of these fields can be found in [1]. Although not educational in nature itself, automatic transcription of music can be used as a first stage to a number of educational applications. When applied to monophonic singing, automatic transcription creates opportunities for applications like melody database retrieval of music also referred to as query-by-humming (QBH) systems, sight-singing tutors, structured audio [2] and various singing analysis systems.

Although the monophonic transcription problem for specific instruments was largely solved approximately 20 years ago [3], the overall flexibility of the human voice as an instrument expands the problem sufficiently to sustain current research interest and contributions. Especially the variance in timbre during phonetically unrestricted singing requires that both the time and frequency domain be used for note onset/offset cues. As noted by *Viitaniemi et al* [4] and *Clarisse et al* [5], segmentation and quantization of the continuous pitch track into a sequence of notes is still an unsolved area of research. Although the currently larger QBH research field has provided much insight into the processing of singing signals, the need for a note level respresentation is of greater importance in the transcription domain, since it corresponds exactly to the output level of representation. The observation therefore made by *Shih et al* [6] regarding the neglect of notes as individual musicological units in QBH systems, is of even greater significance to the singing transcription community.

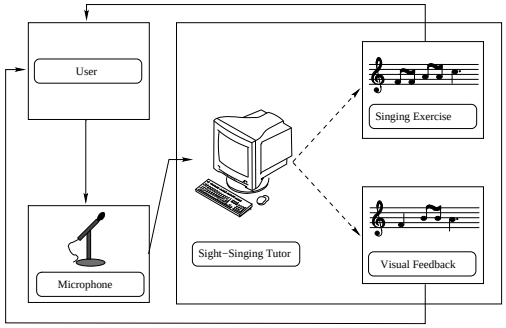

Figure 1: *Schematic diagram of a sight-singing tutor.*

The work presented in this paper, is closely related to the system proposed in [7], whereby notes are individually modeled within a statistical framework. Our system makes use of HTK [8], an HMM toolkit designed for speech recognition applications. Furthermore, our system is intended eventually to be used as a sight-singing tutor platform, and is therefore not refined to be a state-of-the-art transcription system. We have therefore not applied pitch tuning and extensive duration modelling. A schematic representation of a sight-singing tutor system can be seen in Figure 1. With a sight-singing tutor system the user is asked to sing a selected vocal exercise. This exercise is then used by the system as a transcription reference and is compared with the users audio input transcription to determine how accurately the user has sung. The user is then given visual feedback of the singing peformance.

The structure of the paper is as follows: Section 2 gives a general overview of the proposed system and its various components, followed by details on how the dataset was constructed in Section 3. Section 4 describes the acoustic modelling of notes and explains the choice of feature vectors, HMM model topology and grammar. The evaluation of our system is presented in Section 5. The conclusions reached and further recommendations regarding our system are given in Section 6 and conclude the paper.

2. System Overview

Most statistical singing transcription systems are built using the modules shown in Figure 2. Audio input is low-pass filtered to reduce high-frequency noise and harmonics. The filtered signal is transformed into an intermediate-level representation which captures the most essential characteristics and is referred to as signal *features*. These vectors are used to adjust the statistical models, HMM's in our case, to represent the events being

modeled. The one-to-one correspondence between the statistical models and the identity of the notes makes the transcription process conceptually simple. The recognition and segmentation process proceeds by finding the most likely event sequence given a network of models, that would account for the features being observed. Lastly, the note sequence can be adjusted by evaluating each note in terms of the overall sequence within which it occurs, its musicological context and by means of note transition probabilities based on a major-minor scale key pair. Note duration restrictions can also be applied during this phase to absorb clear insertions. In some systems such sequential constraints are integrated into the preceding recognition stage. The final translation of a note sequence to sheet music requires duration quantization and interpretation of the sequence in terms of accepted music notation.

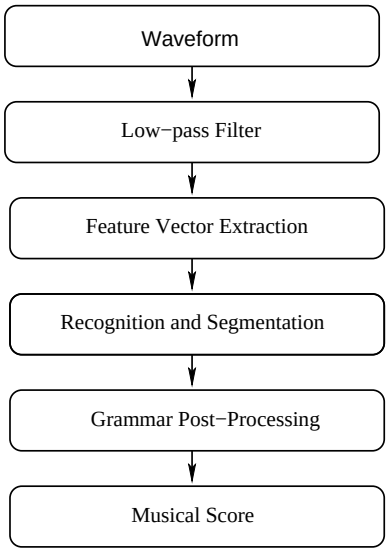

Figure 2: *Schematic representation of a singing transcription system.*

3. Corpus

Due to the lack of recordings of suitable monophonic singing with which to train statistical models, a new dataset was compiled. In order to maximize the limited amount of data the corpus range has been limited to female sopranos only. The dataset contains 10671 notes from 15 female soprano voices spanning 32 semitones (from G3 to D6#). The UNISA grade III, IV and V prescribed list of technical singing exercises was used as a basis for our corpus. Each of the 15 subjects was required to sing an average of 50 such exercises during a recording session. Figure 3 shows a typical example.

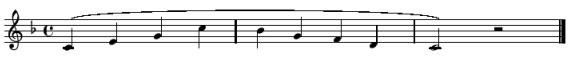

Figure 3: *Unisa vocal exercise example.*

Table 1 shows how the dataset was divided into training and testing data. The *ProTools LE 7.1* recording software and a *Rhode NT2000 Studio Condenser Microphone* were used. All

Table 1: *Dataset partition information*

Descriptor	Training Set	Testing Set
Number of singers	12	3
Number of exercises	624	176

recordings were stored using 16-bit linear encoding at a sampling rate of 44.1kHz. Figure 4 displays the distribution of all the notes in the dataset.

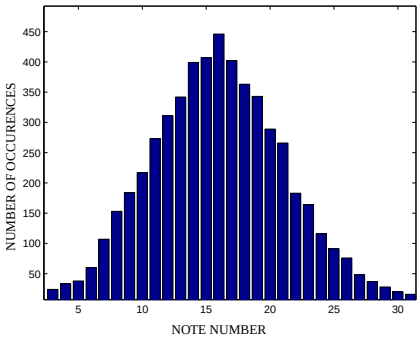

Figure 4: *Distribution of notes in the dataset.*

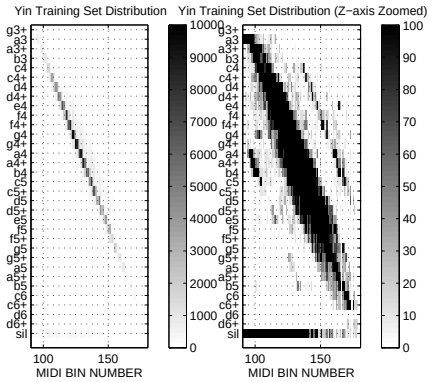

Figure 5: *Histograms of automatically estimated pitch values for each note in the training set. The plots show the number of pitch estimates calculated for each note model. In the graph on the right the histogram bin maximum has been clipped to better show the number of octave and fifth error intervals. Each bin represents 37.5 cents* $\approx \frac{1}{3}$ *semitones. The bin range spans the midi note numbers 21 - 96.*

4. Acoustic Modelling

In order to process the dynamic nature of the human voice, a flexible approach is needed. As can be seen in Figure 5, the various notes in the training dataset display a notable amount of variance in terms of estimated pitch frequency (pitch estimation will be descirbed in Section 4.1.1). The variability can be attributed to singing errors, pitch estimation errors and transition instability regions, but also the inherent stochastic element

of music. *Ryynänen et al* [7] describes note events as being *"musicological units having dynamic nature"*. Hidden Markov models are well suited to this type of problem and are often used for time series modelling. In particular, HMM's can be used to find the optimal corresponding 'hidden' (note) event sequence, given some observed characteristics regarding the melody (i.e. pitch) [4].

The various initialization, training and testing steps associated with most HMM based recognition systems, including the proposed system, are outlined in Figure 6. Firstly, every note sequence within the dataset has to be manually transcribed. The labelling process can be very time consuming, as some note sequences may be incorrect or not suitable for training. A hidden Markov model λ is defined in terms of the matrix of transition probabilities A, the observation probability distributions B and the initial state distribution π: $\lambda = (A, B, \pi)$. The model is initialized and then trained by locally optimizing $P(O|\lambda)$, the probability of observing the sequence of feature vectors O, given a certain model λ. This iterative model training process is known as Baum Welch re-estimation [9, 10]. Once the HMM parameters have been trained, the recognition of notes sequences can be determined, using the Viterbi algorithm [11, 12] which seeks to find the single state sequence Q, that maximizes the probability $P(Q, O|\lambda)$, given an observation sequence O. The most likely note model sequence can then easily be found from the best state sequence. To evaluate the system, each generated note sequence is compared to its reference transcription and an accuracy is calculated.

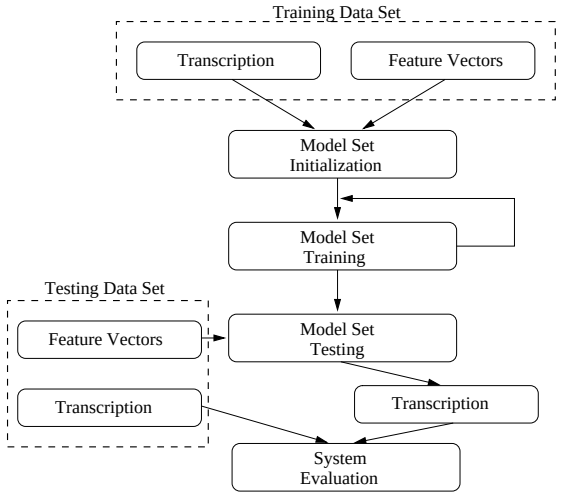

Figure 6: *Acoustic modelling and evaluation steps.*

4.1. Feature Vectors

Unlike speech recognition features that are focused mainly on pronunciation and are largely pitch independent, singing transcription must focus on the pitch and be pronunciation independent. Our system uses pitch, with pitch-delta and voicing coefficients added to assist in note boundary detection. Given that the technical exercises of the dataset consist mainly of single *legato* phrases, the energy envelope itself is not helpful for the extraction of note event features. Many systems use adaptive pitch tuning [4, 7, 13], but since the system will be expanded in the future to accommodate user feedback, absolute pitch frequency is used instead.

4.1.1. The Yin pitch estimator

We use the *Yin* algorithm as proposed in [14], as our primary pitch estimator. This algorithm has been found to be effective in other music transcription systems [4, 7].

For a given discrete time-domain signal x, sampled at a frequency f_s, the Yin algorithm outputs the fundamental frequency f_o at time t together with a voicing parameter v_t. The algorithm is based on a squared difference function $d_t(\tau)$ which is calculated over a window of W samples and is similar to the AMDF function [15]:

$$d_t(\tau) = \sum_{j=t}^{t+W} (x_j - x_{j+\tau})^2$$

Here τ is an integer lag variable such that $\tau \in [0, W)$. The difference function is normalized by dividing by the cumulative mean of the function over shorter lag periods:

$$d_t^{'}(\tau) = \begin{cases} 1 & \tau = 0 \\ d_t(\tau)/[(\frac{1}{\tau})\sum_{j=1}^{\tau} d_t(j)] & \text{otherwise} \end{cases}$$

This eliminates the need to define a lower limit for τ within $d_t^{'}(\tau)$, since the cumulative mean function seeks to maximize the difference function for small lag periods below the pitch period range of interest. The Yin algorithm finds the local minimum with the smallest lag period $\tau^{'}$, and then interpolates over the interval $\{\tau^{'} - 1, \tau^{'} + 1\}$. The minimum of the interpolation polynomial is chosen as τ_p. The pitch period can then be converted to an absolute frequency using $f_{o(t)} = f_s/\tau_p$. The voicing parameter v_t is given by $d_t^{'}(\tau_p)$, which is the magnitude of the Yin function at τ_p. This parameter is a function of the strength of the correlation at τ_p, which is related to the overall degree of periodicity in the signal within the current frame. To enhance pitch continuity and reject clear spurious peaks, only pitch values within the range of 27.5 – 2093.0 Hz (A0 – C7) are accepted as valid, with invalid values set to the previous valid pitch value. The pitch track is smoothed with a 10th order median filter.

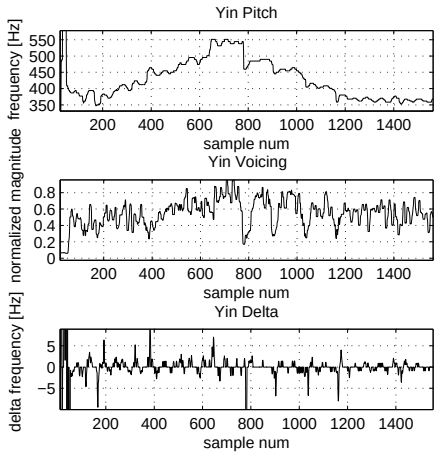

Figure 7: *Typical pitch,voicing and delta features.*

4.1.2. Delta coefficients

The time differentials of the pitch values, referred to as delta coefficients, are calculated at time t using the regression formula [8, p.63] given by:

$$df_{o(t)} = \frac{\sum_{\theta=1}^{\Theta} \theta(f_{o_{t+\theta}} - f_{o_{t-\theta}})}{2\sum_{\theta=1}^{\Theta} \theta^2}$$

The window width parameter, Θ, is set to 2 in our experiments. Figure 7 illustrates a typical pitch track and its associated voicing and delta-pitch values.

4.2. Note event modelling

Due to the similarities between singing transcription and speech recognition, it seems reasonable to incorporate some of the well researched methods and tools that the latter field has to offer. For this reason the HTK speech recognition toolkit [8] has been used for most of our training and recognition procedures. The subjective and unpredictable nature of the human voice calls for a flexible approach, whereby the inherent variability in singing can be modeled and weighted according to probabilistic measures.

Each semitone, corresponding to midi numbers *n = 55...86* is modeled with a single-Gaussian non-skipping left-to-right HMM with the number of states ranging between 1 and 4. As pointed out by *Ryynänen et al* [7], the various states in the HMM models can be seen to represent the different stages in a note event. Initialization of models is achieved by computing and assigning a single global training set mean and variance to all models. Apart from the note onset uncertainty, the transition regions between notes tend to degrade the overall modelling accuracy of notes since the transient pitch is context dependent and can vary greatly depending on the note interval and pronunciation. To decrease this initial note model variance, separate transition models have been inserted between all notes. Two different approaches to transition models have been tested. The first uses two transition models, for ascending and descending transitions respectively. The second uses four transition models with ascending and descending transitions classified as either large (intervals larger than 3 semitones) or small (intervals of 3 semitones or less). The transition models rely heavily on the pitch delta and voicing coefficients to detect note onsets and endings.

The lack of sufficient training data often leads to some undertrained HMM states. To address this we have employed state tying, commonly used to deal with undertraining in speech recognition applications [8, p.148-150]. For HMM's with between 3 and 6 states, we have tied all but the first 2 states. The independent states are left to model the initial instability during the note onset.

4.3. Note recognition grammar

Our first protoype systems exhibited a high rate of insertions since HMM's cannot adequately model durations. A single sustained note was often interpreted as a series of repetitions of the same note. In an effort to avoid this, a simple non-repetitive grammar model has been implemented. A 3 note system example is shown in Figure 8.

This grammar allows transitions from each note to all other notes, but does not allow repetitions of the same note without a separating silence.

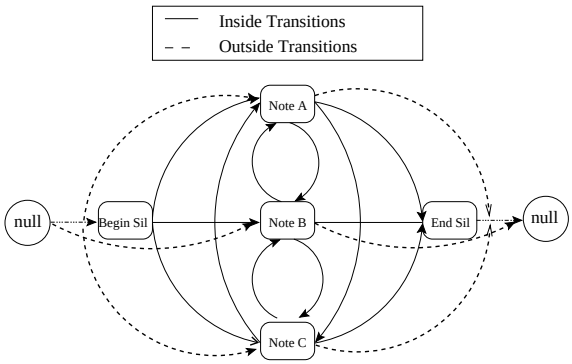

Figure 8: *Non-repeating note grammar network.*

5. Results

We have tested the performance of various system configurations when applied to the transcription of the test-set. Transcriptions were obtained by Viterbi decoding using the acoustic models and the recognition grammar described in the previous section. The systems are evaluated in terms of recognition accuracy as defined in Equation 1. The percentage transcription accuracy, A, is defined as follows:

$$A = \frac{Num - Del - Sub - Ins}{Num} \times 100\% \qquad (1)$$

Where Num is the total number of notes in the transcription reference. Del is the number of deletion errors, Ins the number of insertion errors and Sub the number of substitution errors. The default HTK error weights [8, p.183-184] have been used to evaluate our system. The language model likelihood scaling factor [8, p.183], referred to as the *inter model transition penalty*, used to balance the number of insertion and deletion errors have been kept at -20 for all experiments.

Table 2: *Basic system performance for feature vectors including pitch(P), delta pitch(D) and voicing(V)*

HMM States	P	P+D	P+D+V
1	70.48	51.78	55.47
2	88.31	84.75	75.28
3	85.67	85.42	81.18
4	84.44	87.52	81.49

The first system in Table 2, employs a single HMM for each note, with no transition models. The number of states in each HMM was varied between 1 and 4, and the feature consisted of pitch only(P), pitch and delta pitch(P+D), or pitch, delta pitch and voicing(P+D+V).

The system in Table 3 uses the components of the first system, but also includes transition models between notes. For single-dimensional feature vectors consisting only of the pitch estimate, the introduction of transition model leads generally to a deterioration in performance. Since the transition models are designed to model the change of pitch during the transition from one note to the next, they cannot be sufficiently characterized by pitch alone.

When the delta-pitch is added to the feature vector, the inclusion of the transitional models does lead to a performance

Table 3: *Performance of the transcription system when transition models are included*

HMM States	P	P+D	P+D+V
1	87.27	71.53	68.27
2	87.76	81.43	82.41
3	84.56	86.16	86.84
4	82.84	88.01	87.39

Table 4: *Performance of the transcription system when tied state modeling is included*

HMM States	P	P+D	P+D+V
3	84.19	85.85	80.38
4	87.45	85.67	78.66
5	86.59	85.79	80.69
6	85.73	84.93	79.77

improvement. Only the four transition model system results are shown, although both systems perform with very similar accuracy. The system in Table 4 uses the components of the first system, but ties all but the first 2 HMM states in an attempt to improve model robustness while allowing the inclusion of a larger number of states.

Table 5: *Performance of the transcription system when tied state modeling and transition models are included*

HMM States	P	P+D	P+D+V
3	84.13	85.67	87.15
4	87.70	88.31	85.49
5	88.36	86.95	84.50
6	88.50	87.70	83.46

Lastly, the system in Table 5 combines all the refinements of the previous three systems. Although some of the simpler systems perform equally well in terms of recognition accuracy, the tied state transitional model system seems to be more robust, since it leads to more consistent improvements, as shown in Figure 9.

The dataset contains a large number of technical exercises featuring uninterrupted vowel sequences. This improves pitch track continuity, but tends to degrade the usefulness of the voicing feature whose addition is seen to generally degrade performance. This phenomenon is also reflected in the feature comparison in Figure 10.

Figure 11 provides a recognition accuracy comparison of the different systems implemented and also of the different number of HMM states used. The top graph highlights the consistent gains achieved by introducing transition models and by tying HMM states. The bottom graph shows the effect of increasing the number of states. The necessity of modeling not only the stable pitch region of a note, but also the note onset and transition regions between notes can be seen in the significant recognition accuracy improvement from single state HMM's to HMM's with a multiple number of states.

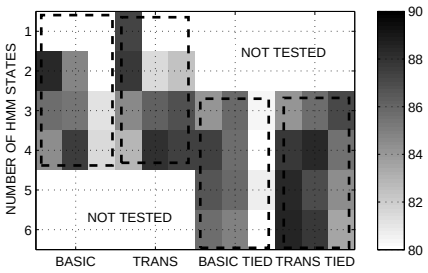

Figure 9: *System performance comparison in terms of recognition accuracy. Darker colours indicate higher recognition percentages as shown by the legend to the right. The columns of each system represent the different feature vector compositions: Pitch, Pitch and Deltas, Pitch,Deltas and Voicing respectively.*

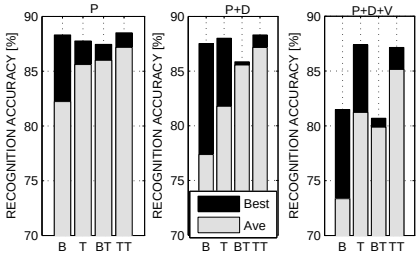

Figure 10: *Feature comparison. The average and best system performance over all number of states: The horizontal axis labels represent the 4 different systems, (B)asic system, (T)ransition models added, (B)asic system with (T)ied states and (T)ransition models with (T)ied states. The abbreviations directly above the figure denote the type of feature vectors used: (P)itch, (P)itch(+D)eltas, (P)itch(+D)eltas(+V)oicing.*

6. Conclusions and future work

A basic transcription system using HMM's, is proposed. The system introduces the concept of transition models, a non-repetitive grammar and tied-state modelling. The system incorporating both transient models and tied-state modelling, is shown to have the best average performance, although the various systems perform very similar in terms of best performance. The best achieved recognition accuracy was 88.50 %. However, the incorporation of delta-pitch and voicing features was less successful. These may still however be useful for alignment purposes in the sight-singing tutor.

Providing that the dataset can be expanded significantly, the system can also be improved by introducing vibrato modelling to reduce note model variance. Bigram and trigram event models could also be introduced as context dependant note models. Other components such as musicologically based transition probabilities and rhythm estimation, which have already been proposed and used with some success, could also be incorporated. By using an HMM based system, the current note segmentation is sufficient to allow note quantization, although a reliable rhythm estimation component should be implemented first.

The current system is ultimately to serve as front-end to a interactive feedback system, also referred to as a sight-singing

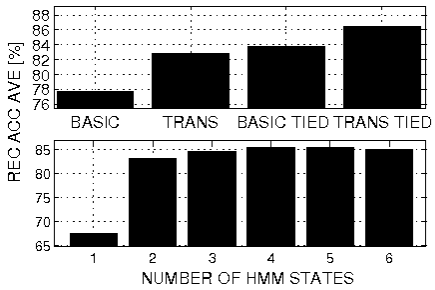

Figure 11: *Average system and state comparison.*

tutor. In order to score the singing performance, a prior reference of the performance is available to the recognition system. This allows for the use of forced time-alignment [8, p.186-186].

7. Acknowledgements

The authors gratefully acknowledge the assistance of M. Oosthuizen and M. Pearce-Du Toit of the Music Department of Stellenbosch University in the collection of the dataset. This work was supported by the South African National Research Foundation (NRF) under grant number FA2005022300010.

8. References

[1] M. Brandao, G. Wiggins, and H. Pain, "Computers in music education symposium on musical creativity," in *Proceedings of the AISB*, 1999.

[2] B. L. Vercoe, W. G. Gardner, and E. D. Scheirer, "Structured audio: creation, transmission, and rendering of parametric sound representations," in *Proceedings of the IEEE*, vol. 86, May 1998, pp. 922–940.

[3] M. Piszczalski, "A computational model for music transcription," Master's thesis, University of Stanford, 1986.

[4] T. Viitaniemi, A. Klapuri, and A. Eronen, "A probabilistic model for the transcription of single-voice melodies," in *Proceedings of the 2003 Finnish Signal Processing Symposium*, May 2003, pp. 59–63.

[5] L. P. Clarisse, J. P. Martens, M. Lesaffre, B. D. Baets, H. Meyer, and M. Leman, "An auditory model based transcriber of singing sequences," in *Proceedings of 3rd International Conference on Music Information Retrieval, ISMIR 02*, May 2002.

[6] C.-C. J. Kuo, H.-H. Shih, and S. S. Narayanan, "An HMM-based approach to humming transcription," in *Proceedings of IEEE International Conference on Multimedia and Expo*, vol. 1, 2002, pp. 337–340.

[7] M.Ryynänen and A. Klapuri, "Probabilistic modelling of note events in the transcription of monophonic melodies," in *Proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, Tampere, 2004. [Online]. Available: http://www.cs.tut.fi/

[8] "The HTK book, version 3.0," Cambridge University Engineering Department, 2000.

[9] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," in *Bell Syst. Tech. J.*, vol. 62, Apr. 1983, pp. 1035–1074.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," in *J. Roy. Stat. Soc.*, vol. 39, 1977, pp. 1–38.

[11] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," in *IEEE Trans. Informat. Theory*, vol. IT-13, Apr. 1967, pp. 260–269.

[12] G. D. Forney, "The Viterbi algorithm," in *Proceedings of the IEEE*, vol. 61, 1973, pp. 268–278.

[13] R. McNab, L. Smith, and I. Witten, "Signal processing for melody transcription," in *Proc. 19th Australasian Computer Science Conf.*, Melbourne, Jan. 1996, pp. 301–307.

[14] H. Kawahara and A. de Cheveigne, "Yin, a fundamental frequency estimator for speech and music," in *J. Acoust. Soc. Am.,111(4)*, 2002, pp. 1917–1930.

[15] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-time processing of speech signals*. New York: MacMillan Publishing Co., 1993.