A PROBABILISTIC MODEL FOR THE TRANSCRIPTION OF SINGLE-VOICE MELODIES

Timo Viitaniemi, Anssi Klapuri, Antti Eronen

Institute of Signal Processing, Tampere University of Technology,
P.O.Box 553, FIN-33101 Tampere, FINLAND,
{viitanit, klap, eronen}@ cs.tut.fi

ABSTRACT

A method is proposed for the automatic transcription of single-voice melodies from an acoustic waveform into a symbolic musical notation (a MIDI file). The system consists of a signal processing front-end which calculates a continuous pitch track and of a probabilistic model which converts the pitch track into a discrete musical notation. Our proposed probabilistic model consists of three parts operating in parallel: a pitch trajectory model, a musicological model, and a duration model. The first handles imperfections in the performed/estimated pitch values using a hidden Markov model, the second estimates musical key signature to improve the transcription accuracy, and the last models the duration of the notes.

1. INTRODUCTION

Transcription of music refers to the act of listening to a piece of music and of writing down a symbolic musical notation for it. Automatic transcription of sung, hummed, or whistled melodies has several applications. Retrieval of music based on a query by humming has become a topic of interest in last few years [1, 2, 3, 4]. Writing down musical scores using a notation software would be greatly facilitated when acoustic input (singing) would be allowed. Also, automatic transcription allows the development of programs for the self-study of music.

Transcription of single-voice melodies comprises two main subproblems. First, a track of pitch estimates is extracted from an acoustic waveform (see Figure1(a)). This part is largely a solved problem since several algorithms are available that are accurate and operate in real time. However, reliable conversion from a continuous pitch track to a symbolic musical notation has turned out to be very difficult (see Figure 1(b)). This is due to, e.g., imperfections in the sung pitch values and durations, transitions between short notes, and vibrato.

On a computer, a musical notation is typically represented as a MIDI stream or a MIDI file (a symbolic music representation format consisting of the notes, their onset and offset times, the instrument, and so on). In principle, the simplest possible conversion from a pitch track to a MIDI representation could be done by simply rounding a pitch estimate to a closest MIDI note n_t and interpreting all pitch value changes as note boundaries. The pitch of the MIDI note is represented as an integer number, obtained from a fundamental frequency value x_t (Hz) by rounding

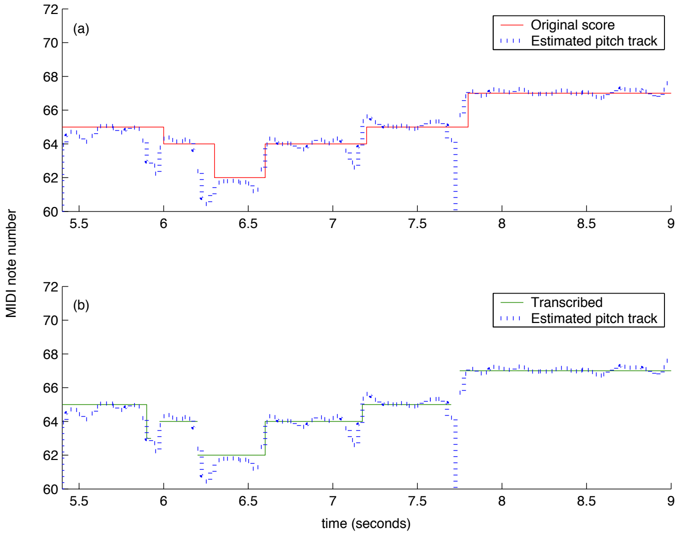

Figure 1. Examples of (a) the notes of the reference melody in a MIDI-representation, and the estimated pitch-track of the recorded singing; (b) the quantized pitch curve of the estimated pitch-track.

$$n_t = 69 + 12\frac{log(\frac{x_t}{440})}{log(2)}. \qquad (1)$$

However, simply rounding from the above equation produces very poor results. Statistical methods have turned out to be more successful for transcribing human voice [5, 6].

In this work, we are using a statistical approach and estimation of higher-level musical concepts, i.e., key signature and tempo, to achieve more reliable notation results. In the following, the notation problem is approached using three parallel probabilistic models (see Figure 2). First of these, a *pitch-trajectory model*, is a hidden Markov model (HMM) which extracts the countour of the melody-line. States of this model correspond to discrete note values of the original melody with time-units of discrete pitch-

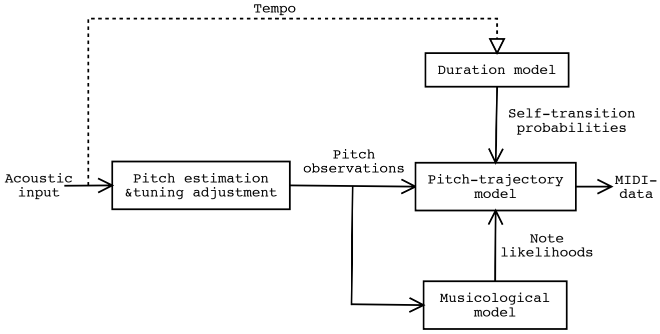

Figure 2. Description of the overall system.

estimation frames. The estimated pitch track is interpreted as an output of the HMM (observations).

The second probabilistic model, a *musicological model*, estimates the key signature from a measured pitch track. Each note has a key signature dependent probability of occurence which are weighted dynamically according to the estimated key signature.

The last probabilistic model, a *duration model*, adjusts the state self-transition probabilities of the pitch-trajectory model according to the current tempo. This model relies on the fact that the note durations in a real music, are discrete rather than continuous, e.g., in relations of, $\frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, \frac{1}{1}$. In this work, the tempo is assumed to be known a priori, although it is possible to estimate it from the time-domain signal [7].

2. SIGNAL PROCESSING FRONT-END

A number of good algorithms are available for extracting the pitch track of single-voice speech and singing signals. The YIN-algorithm of de Cheveigne and Kawahara proved to be suitable for the present task and was thus employed. The method has been originally proposed in [8] and has been extensively evaluated and compared to earlier methods by the original authors.

The original discrete time-domain signal is denoted by y_m. Fundamental frequency (F_0) estimates are denoted as x_t, where t is the time frame index and values of x_t are restricted between 60 Hz and 2.1 kHz. F_0 estimates are produced in successive 50 ms frames (W) with 25 ms overlap as follows:

1. Calculate the difference function of a signal y_m as

$$d_m(\tau) = \sum_{j=m}^{m+W-1} (y_j - y_{j+\tau})^2. \tag{2}$$

   If y_m is perfectly period, $d_m(\tau)$ equals zero at wavelengths τ which correspond to multiples of the true period.

2. The difference function (2) is mean-normalized as

$$d'_m(\tau) = \begin{cases} 1 & ,\tau = 0 \\ d_m(\tau) / \left[(1/\tau) \sum_{j=1}^{\tau} d_m(j) \right] & ,\tau \neq 0 \end{cases} \tag{3}$$

   resulting d'_m to be independent of the power of the signal y_m.

3. Real-world signals of interest are not perfectly periodic, i.e., d'_m is just approaching zero. For this reason, an absolute threshold is applied which selects the smallest value of τ that gives a local minimum of d'_m deeper than the absolute threshold. This threshold determines the level of periodicity that is required. The value used here was 0.15 and it was trained using an acoustic database (see Section 5.1).

4. The selected local minimum of $d'_m(\tau)$ and its immediate neighbours are fit by a parabola, and the ordinate of the interpolated minimum is used to obtain a more accurate (real valued estimate of τ.

Only very few singers can tune their voice according to an absolute tuning (e.g., $A_4 = 440$ Hz) without hearing a reference melody. This means that there exist a difference δ between an absolute tuning and the tuning of the singer, which has to be estimated and compensated. This estimation and compensation can be done (in MIDI note numbers) as follows:

$$\delta = \frac{1}{T} \sum_{t=1}^{T} \left[mod(x'_t + 0.5, 1) - 0.5 \right] \tag{4}$$

$$x_t = x'_t + 0.5 - mod(\delta + 0.5, 1), \tag{5}$$

where mod is the modulus-after-division operator and x'_t is the pitch estimate before tuning adjustment.

3. PROBABILISTIC MODELS

3.1. Observation Probabilities

There is no one-to-one relation between a measured pitch curve and the notes of the original melody, since both the singer and the pitch estimator produce "errors". For this reason, the note values "behind" a pitch track cannot be directly observed. This motivates the usage of an HMM, where the internal states correspond to discrete note values (one state per note in this paper) and F_0 estimates to the symbols in the output of the model.

The relationship between discrete (note) states and pitch estimates is modelled with an observation probability distribution $P(o = x | n = i)$, i.e., the probability of a pitch estimate x given a (note) state i. This distribution can be estimated using an acoustic database, where each signal is associated with a reference transcription of discrete note values. Such a database was collected and based on that the observation distribution is estimated as

$$b_i(o) = P(o = x | n = i) \approx \frac{count(o = x, n = i)}{count(n = i)}, \tag{6}$$

where $o = x$ denotes the occurence of fundamental frequency x (Hz), and $n = i$ denotes the occurence of the i:th note. We assume that the observation density has a similar shape across all states. Thus the observation densities of different states differ only with respect to the location of the zero offset (actual F_0 value). The shape of the observation density is shown in Figure 3 which represents the estimated observation distribution modelled with a *Gaussian mixture model* (GMM). The abscissa of the figure indicates the pitch estimate's distance from the pitch of a state i in semitones. Distribution shows that states have some amount of probability mass for erroneous pitch estimates about an octave below and an octave above the true value. These are due to the errors made by the YIN-algorithm.

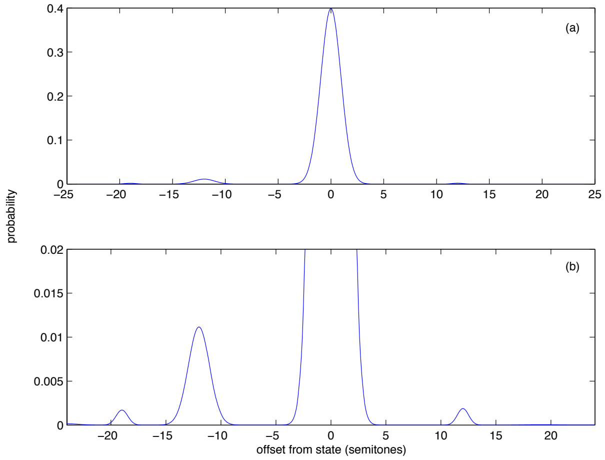

Figure 3. Observation distribution $b_i(o)$ modelled with a GMM: (a) in full scale, and (b) in a zoom plot to emphasize octave peaks at 12 semitones above and below the state i, and a fifth peak at 19 semitones below the state i.

3.2. Key Signature Probabilities

Key signature is a note-system, where some notes have a greater probability of occurrence than others. Most common key signatures in the western music are the twelve major keys (C major, D major etc.) and the twelve minor keys (C minor, D minor etc.). In this paper, we calculate probabilities for these 24 key signatures based on estimated pitch values.

Occurrence probabilities of different note values given the key, i.e., $P(n = i|k = j)$, have been estimated from large amouts of classical music by several authors [9]. Using the Bayes formula, the probability of different key signatures given the pitch measurements can be calculated as

- Probability of the j:th key given one note n

$$P(k = j|n = i) = \frac{P(n = i|k = j)P(k = j)}{P(n = i)} \qquad (7)$$

- Probability of the j:th key given one pitch-estimate x.

$$\begin{aligned} P(k = j|x) &= \frac{P(x|k = j)P(k = j)}{P(x)} \\ &= \sum_{\text{all } i} P(x|n = i)P(n = i|k = j)\frac{P(k = j)}{P(x)}, \qquad (8) \\ i &= 1 \ldots N, \end{aligned}$$

where the observation probabilities are assumed to be independent of the key after a note value $n = i$ has been given, i.e., $P(x|k = j, n = i) = P(x|n = i)$.

- The probability of the j:th key given pitch estimates $O = (x_1 \ldots x_T)$.

$$P(k = j|O) = \prod_{t=1}^{T} P(k = j|x_t) \qquad (9)$$
$$i = 1 \ldots N$$

The key probabilities act as a musicological model and can be applied to compute the probability of a note state i at time instant t given a series of observations up to t as

$$P_k(n_t = i|O) = \sum_{j=1}^{24} P(n_t = i|k = j)P(k = j|O), \qquad (10)$$
$$i = 1 \ldots N,\ O = o_1, o_2, \ldots, o_t$$

3.3. Bigram Probabilities

Bigram probabilities $P(n_{t+1} = i|n_t = j)$, i.e., the probabilities for a transition from one note (state j) to another (state i), were estimated from the EsAC database [10]. The EsAC-database, in its public form, contains 5983 folksongs which are mostly from Germany.

The bigram probabilities were estimated independently of the musical key, i.e., all key signatures were normalized to be equally probable. The result of the estimation for transitions from a state can be seen in Figure 4. The probability of self-transition (offset 0) is missing since it is defined by the duration model. A fixed self-transition probability was used in simulations in which the duration model was not used. This self-transition probability was trained with an acoustic database, and the value 0.4 was used.

3.4. Duration Model

In this paper we assume that durations of the HMM states are independent of the pitch, i.e., all notes can share the same duration distribution. The duration model of a conventional HMM is an exponential distribution [11, page 259]. This models poorly the duration of musical notes since real note durations approximate discrete time values ($\ldots \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, \frac{1}{1} \ldots$) and are tempo dependent (duration of $\frac{1}{4}$ equals to $1/\textit{tempo}$ sec). This motivates the use of explicit duration modelling.

Let ϵ(d) be an arbitraty duration distribution of the duration model, and let $P(d)$ be the probability for state

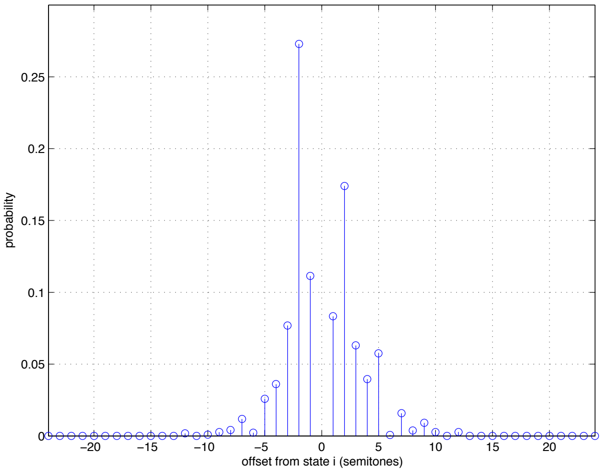

Figure 4. Bigram probabilities estimated from the EsAC-database.

exchange after being in the same state for d consecutive time-instants. The following procedure converts an arbitrary duration distribution $\epsilon(d)$ into state-exchange probabilities $P(d)$, which can be directly used in a modified *Viterbi*-algorithm.

Initialize $P(1) = \epsilon(1)$

Step 2 : $\epsilon(2) = (1 - P(1))P(2)$

$\Leftrightarrow P(2) = \epsilon(2)/(1 - \epsilon(1))$

Step d : $\epsilon(d) = (1 - P(1))(1 - P(2)) \ldots (1 - P(d-1))P(d)$

$$\Leftrightarrow P(d) = \frac{\epsilon(d)}{(1 - P(1))(1 - P(2)) \ldots (1 - P(d-1))}$$

The duration histogram for real music was estimated from the EsAC-database. A smooth approximation of the duration density is obtained using the GMM shown in Figure 5, whose component weights have been taken from the histogram.

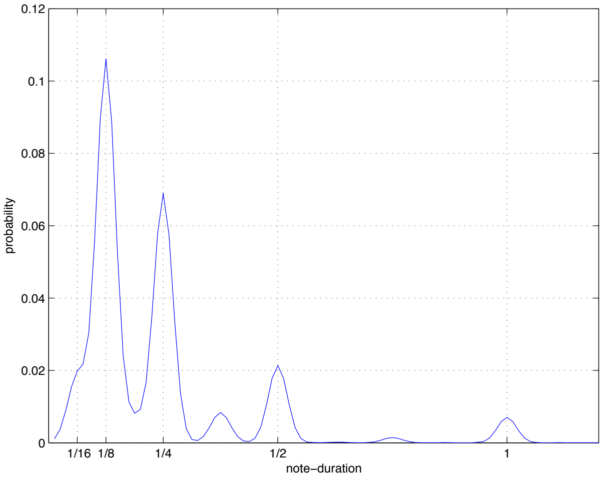

Figure 5. A smooth duration density modelled with a GMM.

4. COMPUTATIONS

The most likely sequence of notes using the described model is computed using a modified Viterbi-algorithm. Let $d_i^{(t)}$ denote the duration in the state i at time instant t.

- **Observation probabilities** $b_i(x_t)$: see Eq. 6.
- **Transition probabilities** from state i to state j:

$$a_{ij}(d_i^{(t)}) = \begin{cases} P(d_i^{(t)})P(n_t = j|n_{t-1} = i)P_k(n_t = j|O) & , j \neq i \\ (1 - P(d_i^{(t)}))P_k(n_t = j|O) & , j = i \end{cases}$$

- **Updating duration information** after transition from state i to j:

$$d_j^{(t+1)} = \begin{cases} 1 & , j \neq i \\ d_j^{(t)} + 1 & , j = i \end{cases}$$

5. SIMULATIONS

5.1. Vox-database

An acoustic database referred as a *Vox* was used in validating the proposed models. It contains single-voice samples from 11 non-professional singers. The total size of the database is 120 minutes, and all recordings were stored as PCM waveforms with 44.1 kHz sampling rate and 16 bit resolution. Each of the 11 test subjects performed four folk songs and two scales, which they sung, hummed, and whistled. All recordings from seven persons (excluding scales) were used to train the current model, and recordings from four persons (excluding scales) were used to test it.

5.2. Key Estimation Evaluation

The accuracy of the key signature estimation (Equation 9) determines the efficiency of the musicological model. However, it is not necessary that the estimation finds exactly the correct key signature. This is due the fact that each major key has a *relative* minor consisting the same notes than the major key. One example of relative keys are C major and A minor which both consist of the notes *c, d, e, f, g, a*, and *b*. For this reason it is adequate to accept also the relative key as a correct.

This accuracy of key estimation was tested with the contents of Vox-database, and the results were classified according to criteria "*Exactly Correct*" and "Correct on Relative Key", and the results can be seen in Table 1.

Table 1. Key estimation results.

Input Data	**Result Criterion**	**Correct Rate (%)**
Acoustic	Exactly Correct Correct or Relative Key	68 86
Reference (MIDI)	Exactly Correct Correct or Relative Key	89 94

5.3. Overall System Evaluation

The performance of the overall system is evaluated with three error criteria: *Total Error*, *Fine Error*, and *Gross Error*.

- *Total Error* contains all errors

$$E_{tot} = \frac{\text{“Erroneously transcribed frames”}}{\text{“Amount of voiced frames in reference”}} \times 100\%$$

- *Fine Error* contains errors that are less than 20% in frequency domain (equals 3.16 semitones)

$$E_f = \frac{\text{“Amount of the Fine Errors”}}{\text{“Amount of voiced frames in reference”}} \times 100\%$$

- *Gross Error* contains errors that are greater than 20% in frequency domain

$$E_g = \frac{\text{“Amount of the gross errors in frames”}}{\text{“Amount of voiced frames in reference”}} \times 100\%$$

We noticed that recordings and the original melodies were seldom exactly synchronized in time causing unreasonable growth of the error percentages. This was compensated by allowing 50 ms mutual unsynchronization so that at note transition regions, both the preceding and the following note value are interpreted as correct.

The simulation results of the proposed model can be seen in Table 2. Label M_1 refers to note-transition (bigram) probabilities, M_2 to the application of key signature probabilites, and M_3 to using explicit duration modelling. Among these, duration modeling does not produce performance improvement. This may be due to the applied duration distribution or that the Viterbi-algorithm cannot find the optimal path without maximization over all durations. For these reasons it does not necessarily mean that the duration model would not improve results after further development.

Table 3 lists the results for different techniques with the method of best overall performance ($M_1 + M_2$). These results correlate well with the singers' statements concerning the difficulty of each singing technique.

Table 2. Simulation results with different model configurations.

Method used:	E_{tot} (%)	E_g (%)	E_f(%)
Rounding	20	1,9	18
M_1	15	1.5	13
M_2	17	1.9	16
$M_1 + M_2$	**13**	**1.4**	**12**
$M_1 + M_2 + M_3$	15	1.3	14

6. CONCLUSION

A probabilistic approach for the automatic transcription of monophonic music was proposed. The method consists of three parallel parts: the pitch-trajectory model, the musicological model, and the duration model. Each of these characterizes a certain aspect of music. In simulations, the first two models improved the transcription accuracy significantly. The duration model did not bring performance advantage with the applied error measures.

Table 3. Simulation results for the best model ($M_1 + M_2$) when given different types of acoustic input signals.

Results for best model M_1+M_2:			
Technique	E_{tot} (%)	E_g (%)	E_f(%)
Singing	12	1.6	11
Syllable	**9**	**0.9**	**8**
Humming	14	1.6	12
Whistling	19	1.6	18

7. REFERENCES

[1] D. Chamberlin A. Ghias, J. Logan. Query by humming: Musical information retrieval in an audio database. San Fransisco, California, November 1995. Cornell University, ACM Multimedia Conference'95.

[2] E. Pollastri G. Haus. An audio front end for query-by-humming systems. Music Information Retrieval, 2001.

[3] M. Sandler J. P. Bello, G. Monti. Techniques for automatic music transcription. Music Information Retrieval, 2000.

[4] E. Selfridge-Field. What motivates a musical query? Music Information Retrieval, 2000.

[5] X. Rodet B. Doval. Fundamental frequency estimation and tracking using maximum likehood harmonic matching and HMMs. 1993.

[6] C.-C. Jay Kuo H.-H. Shih, S. S. Narayanan. An HMM-based approach to humming transcription. volume 1, pages 337–340. Proc. IEEE Int. Conf. on Multimedia and Expo, 2002.

[7] A. P. Klapuri. Musical meter estimation and music transcription. Cambridge Music Processing Colloquium, 2003.

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

[9] C. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, first edition, 1990.

[10] E. Dahlig. Esac -database: Essen associative code and folksong database, 1994.

[11] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. volume 77, pages 257–289, February 1989.