

Modelling of Note Events for Singing Transcription

Matti P. Ryyänänen, Anssi P. Klapuri

Institute of Signal Processing, Tampere University of Technology,
P.O.Box 553, FI-33101 Tampere, Finland
{matti.ryynanen, anssi.klapuri}@tut.fi

Abstract

This paper concerns the automatic transcription of music and proposes a method for transcribing sung melodies. The method produces symbolic notations (i.e., MIDI files) from acoustic inputs based on two probabilistic models: a note event model and a musicological model. Note events are described with a hidden Markov model (HMM) using four musical features: pitch, voicing, accent, and metrical accent. The model uses these features to calculate the likelihoods of different notes and performs note segmentation. The musicological model applies key estimation and the likelihoods of two-note and three-note sequences to determine transition likelihoods between different note events. These two models form a melody transcription system with a modular architecture which can be extended with desired front-end feature extractors and musicological rules. The system transcribes correctly over 90 % of notes, thus halving the amount of errors compared to a simple rounding of pitch estimates to the nearest MIDI note.

1. Introduction

Transcription of music refers to the process of generating symbolic notations, i.e., *musical transcriptions*, for musical performances. Conventionally, musical transcriptions have been written by hand, requiring both time and musical education. If the transcription could be accomplished computationally, it would significantly benefit music professionals and, more importantly, enable voice-input functionalities in consumer applications. *Melodies* are consecutive note sequences with organised and recognisable shape, and they are important in characterising music content.

The conventional approach to transcribe melodies is to extract pitch estimates from an acoustic input and to convert these into a symbolic notation. There are several solutions for pitch estimation. For a review of different methods, see [1], [2], and [3]. However, a reliable conversion of pitch estimates into a symbolic notation has proven to be a challenging problem, especially for singing. This is because a typical performance contains inaccuracies in both pitch and timing which are difficult to correct without prior knowledge about musical conventions. If pitch estimates are simply rounded to the nearest MIDI note numbers and considered as notes, the result both sounds coarse and provides no note boundaries required to produce a symbolic notation.

In recent years, several methods have been proposed for transcribing monophonic melodies (i.e., melodies with a single note sounding at a time) particularly in the context of query-by-humming systems. Clarisse *et al.* proposed a method which first determines note segments from a humming input and then assigns a note value for each segmented note region [4]. Another system, proposed by Viitaniemi *et al.* in [5], introduced a musical-key estimation model and used a probabilistic model to infer note values from raw pitch estimates. In addition, they estimated transition likelihoods between pitched frames to enhance the transcription accuracy. However,

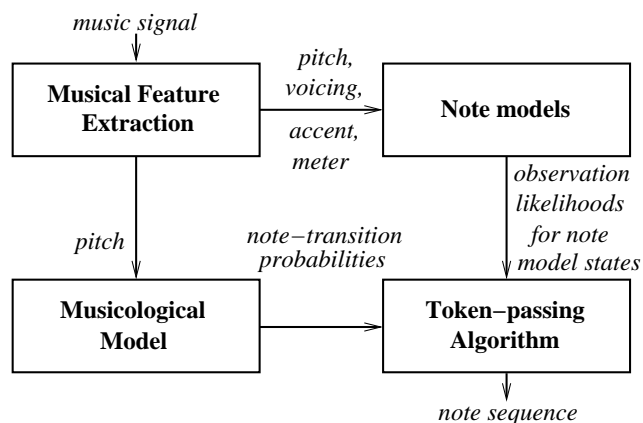


Figure 1: The block diagram of our melody transcription system.

both of the mentioned systems ignore note events as musicological units having dynamic nature. This idea of notes was considered by Shih *et al.* who modelled hummed notes with a three-state left-to-right HMM [6]. Their note model focused on modelling phonemes by using mel-frequency cepstral coefficients, energy measures, and the derivatives of these. However, their approach was to model the timbre of the hummed phonemes instead of the musical features of note events.

This paper presents a probabilistic note model that considers the temporal behaviour of musical features during note events and enables a more appropriate examination of the musical relationships between consecutive notes. Figure 1 shows the block diagram of our transcription system. First, the system extracts musical features from a music signal. The pitch estimates are processed by the musicological model which estimates the musical key and produces a matrix of transition likelihoods between notes. The musical features are used by the note event models to calculate likelihoods of different note candidates. The most probable note sequence is found by using the Token-passing algorithm [7], producing the transcribed melody and note boundaries.

The main focus of this paper is on the modelling of note events and on using the model in singing transcription. The paper is organised as follows. Section 2 explains the extraction of musical features. Section 3 introduces the note model and the musicological model. Section 4 explains the evaluation of the system and reports the transcription results, and Section 5 summarises the contents of the paper.

2. Musical feature extraction

The front-end of the transcription system extracts four musical features in successive frames of an input signal: *pitch* x_t , *voicing* v_t ,

accent a_t , and meter m_t where t denotes the starting time of a frame. The features are extracted in 25 ms intervals. Pitch and voicing represent the fundamental frequency and the degree of periodicity within a frame, respectively. Accent and meter features indicate the degree of *phenomenal accent* and *metrical accent* as a function of time [8]. Phenomenal accents refer to the moments having perceptual emphasis in music signals and in practise indicate performed note beginnings. Metrical accent, on the other hand, corresponds to the underlying pulse of a music performance and it is used to indicate predicted note beginnings, i.e., should there exist a note beginning according to the timing of the performance. Figure 2 shows the four musical features extracted from a singing performance.

2.1. Pitch and voicing extraction

Pitch and voicing are extracted using the YIN algorithm, as originally proposed by de Cheveigné and Kawahara in [3]. Given that y_n is a discrete time-domain signal with sampling rate f_s (Hz), κ is a constant absolute threshold value, and W is the summing interval of 50 ms, the YIN algorithm produces a pitch estimate x_t and a voicing value ν_t at time t as follows:

1. Calculate the squared difference function $d_t(\tau)$ where τ is the lag.

$$d_t(\tau) = \sum_{j=t}^{t+W-1} (y_j - y_{j+\tau})^2 \quad (1)$$

The lag gets values $0 \leq \tau < W$.

2. Evaluate the cumulative-mean-normalised difference function $d'_t(\tau)$:

$$d'_t(\tau) = \begin{cases} 1, & \tau = 0 \\ d_t(\tau) / [(1/\tau) \sum_{j=1}^{\tau} d_t(j)], & \text{otherwise.} \end{cases} \quad (2)$$

3. Find the smallest value of τ for which a local minimum of $d'_t(\tau)$ is smaller than a given absolute threshold value κ . If no such value is found, find the global minimum of $d'_t(\tau)$ instead. Denote this lag value with τ' .
4. Interpolate the $d'_t(\tau)$ function values at abscissas $\{\tau' - 1, \tau', \tau' + 1\}$ with a second order polynomial. Search the minimum of the polynomial in the range $(\tau' - 1, \tau' + 1)$ and denote the corresponding lag value with $\hat{\tau}$.

In our transcription system, the pitch x_t and the voicing ν_t features are obtained by

$$x_t = 69 + 12 \log_2 \left(\frac{f_s / \hat{\tau}}{440 \text{ Hz}} \right), \quad \text{and} \quad (3)$$

$$\nu_t = d'_t(\hat{\tau}). \quad (4)$$

The values of the voicing feature range between zero and one. A value below the absolute threshold ($\nu_t < \kappa$) denotes a *voiced pitch estimate*, meaning that small voicing values express a high degree of periodicity. We use the absolute threshold value $\kappa = 0.15$.

If there is no reference tuning available for a singing performance, pitch estimates can be tuned with an algorithm proposed in [9] to minimise the distance between pitch estimates and MIDI notes. In addition, the algorithm compensates for a possible base-line drift of tuning.

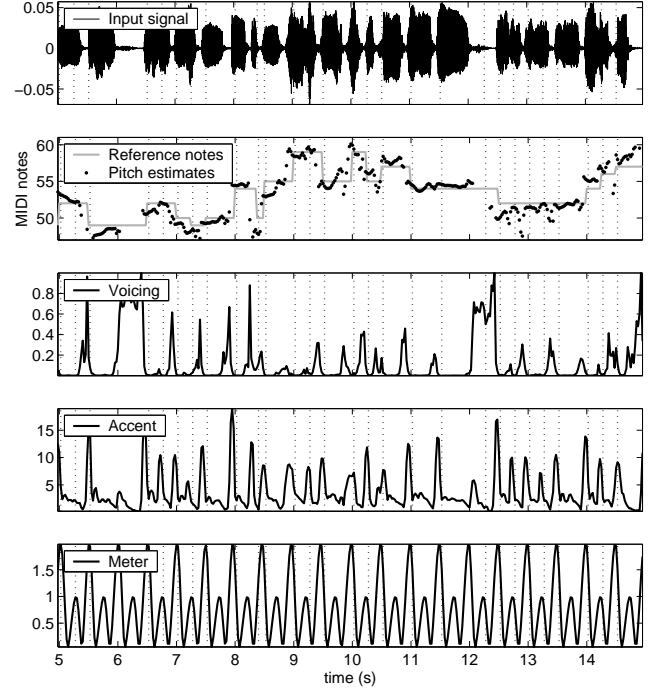


Figure 2: Pitch, voicing, accent, and meter extracted from a singing performance. The dotted vertical lines denote the reference note beginnings. Notice that small values of voicing indicate clear periodicity.

2.2. Accent and meter estimation

The accent feature indicates the degree of phenomenal accent in each frame, and it is here used to indicate the potential time instants of note beginnings. The accent estimation method was proposed in [10] and it produces *registral accent signals* at four frequency channels of an audio input. We apply the method as such, except that the outputs are summed across the channels. Since the sampling rate of registral accent signals by the original author is greater than our frame rate, the maximum of the summary signal in frame t is used as the accent feature a_t .

The metrical accent estimation is performed with the method proposed in [10] which is also capable of following changes in tempo. The method uses the registral accent signals to produce estimates of the metrical pulses at three levels: measures, tactus, and tatum levels. Measures divide a music performance into musical segments, tactus corresponds to the tempo, and tatum expresses the pulsing of the smallest temporal unit in the performance. The meter feature m_t is produced by generating sinusoidal impulses at the time instants where a tatum beat or a tactus beat has occurred. First, tatum beats are used to generate a signal m_t^{tat} by

$$m_t^{\text{tat}} = \cos \left(2\pi \frac{t - \theta_i}{\theta_{i+1} - \theta_i} \right), \quad \theta_i \leq t < \theta_{i+1}, \quad (5)$$

where θ_i is the occurrence time of the i :th tatum beat for $i = 2, 3, \dots, T-2$, and θ_T is the last tatum beat time. If $i = 1$, the equation is applied for $\theta_1 - (\theta_2 + \theta_1)/2 \leq t < \theta_2$ to prevent m_t^{tat} from abruptly jumping to one at $t = \theta_1$. Correspondingly for $i = T-1$, the equation holds at $\theta_{T-1} \leq t < \theta_T + (\theta_T + \theta_{T-1})/2$. Otherwise, m_t^{tat} is zero. Second, tactus beats are used to generate a signal m_t^{tac} in a similar fashion. However, tactus beats occur less frequently than tatum beats, and generating m_t^{tac} in the same

manner as in (5) would produce too wide sinusoidal pulses. Therefore, the pulse widths are determined by the tatum beats. Given the k :th tactus beat time β_k , we search the tatum beat nearest to β_k and denote it with θ_i^k . The m_t^{tac} is then defined as

$$m_t^{\text{tac}} = \begin{cases} \cos\left(2\pi\frac{t - \beta_k}{2\beta_k - \theta_i^k + \theta_{i-1}^k}\right), & \frac{\theta_i^k + \theta_{i-1}^k}{2} \leq t < \beta_k \\ \cos\left(2\pi\frac{t - \beta_k}{\theta_{i+1}^k + \theta_i^k - 2\beta_k}\right), & \beta_k \leq t < \frac{\theta_i^k + \theta_{i+1}^k}{2} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Finally, the meter feature is obtained by

$$m_t = \frac{1}{2} (m_t^{\text{tat}} + m_t^{\text{tac}}) + 1. \quad (7)$$

3. Probabilistic models

The transcription system applies two probabilistic models to melody transcription. A note event model uses the musical features to calculate likelihoods of different notes and a musicological model determines the probabilities of transitions between notes.

3.1. Note event model

Note events are described with a left-to-right hidden Markov model (HMM). The note HMM is a state machine where the states approximate the typical values of musical features in the corresponding temporal segments of note events. The model represents different notes so that there exists one note HMM per each MIDI note number $n = 36, \dots, 79$. Given the musical features at time t , we determine the *observation likelihoods* that a certain state of note n has emitted the features. For note n , the musical features comprise an *observation vector* \mathbf{o} where we use the *pitch difference* $\Delta x = x - n$ instead of the absolute pitch estimate value x . The HMM parameters (and also the other features) are thus independent of the represented note, and only one note-HMM parameters need to be trained in order to represent all the different notes. The model is defined by the following HMM parameters.

1. The set of states $S = \{s_1, s_2, \dots, s_K\}$ within the model where K is the number of states. The i :th state of the model represents the i :th temporal segment of a note event.
2. The state-transition probabilities, i.e., the conditional probabilities $P(s_j|s_i)$ that state s_i is followed by state s_j where $s_i, s_j \in S$. The HMM topology is left-to-right without skips, meaning that $P(s_j|s_i) \neq 0$ only when $j = i$ or $j = i + 1$.
3. The observation likelihood distributions, i.e., the likelihoods $P(\mathbf{o}|s_j)$ that an observation vector \mathbf{o} is emitted by state $s_j \in S$.
4. The initial and the final state probabilities. States s_1 and s_K must be the first and the last state within a note event.

The state-transition probabilities and the observation likelihood distributions were estimated from an acoustic database containing audio material performed by five male and six female non-professional singers. The singers were accompanied by MIDI representations of the melodies which they heard through headphones while performing. Only the performed melody was recorded and, later, the reference accompaniments were synchronised with the performances. The performances of three male and four female singers were used to train the note model, including approximately 3100 note events in total. The reference notes were used to determine note boundaries in the training material, and pitches x_t^{ref}

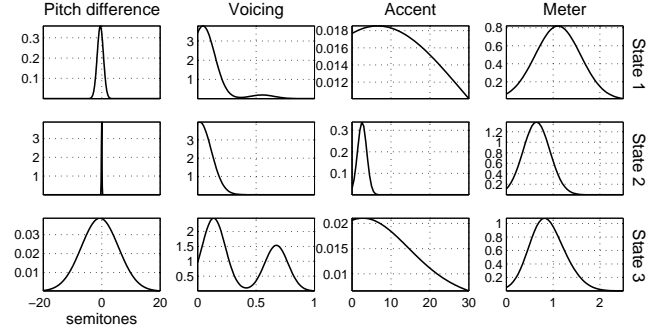


Figure 3: The observation likelihood distributions in three states modelled with two GMM components for the feature set $\{\Delta x, \nu, a, m\}$.

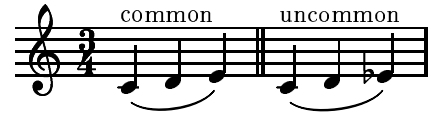


Figure 4: Examples of a common and an uncommon three-note sequences in the C-major-key context.

of the reference notes were used to determine the pitch difference $\Delta x_t = x_t - x_t^{\text{ref}}$ in frame t . The musical features during note events formed observation sequences from which the maximum likelihood estimates for the HMM parameters were obtained using the Baum-Welch algorithm (explained e.g. in [11]).

The model was trained for four different *feature sets*: $\{\Delta x, \nu\}$, $\{\Delta x, \nu, a\}$, $\{\Delta x, \nu, m\}$, and $\{\Delta x, \nu, a, m\}$. The number of features in a set is equal to the length of the observation vector \mathbf{o} . The features are here assumed to be statistically independent of each other. The number of states was varied from one to five and the number of Gaussian mixture model (GMM) components η in the observation likelihood distributions from one to six.

Figure 3 shows the trained observation likelihood distributions for a three-state note model using all the features. The model states 1, 2, and 3 could be interpreted as *transient*, *sustain*, and *silence* segments of a note event, respectively. The transient segment of note events exhibits a wide-spread accent-value distribution, typical meter values near one expressing the occurrence of tatum beats, and some variance in the values of Δx . During the sustain stage, the variance of Δx is small, the frames are mostly voiced, and the accent and meter values are smaller. Eventually in the silence state, Δx values spread, and voicing becomes bimodal. This shows that some of note events include silence or noise at the end of the events, usually as a consequence of breathing or consonants between notes. These distributions express the typical behaviour of the musical features during note events.

3.2. Musicological model

The musicological model controls transitions between notes in a probabilistic manner based on the fact that some note sequences are more common than others in a certain *musical key*. Figure 4 shows two three-note sequences in C major key. Despite only a semitone difference in the last notes, the first sequence is significantly more common than the second one in this musical context. Given the key of C major and an ambiguous note estimate between the last notes, the musicological model would prefer the first sequence, thus producing a musically more meaningful note sequence.

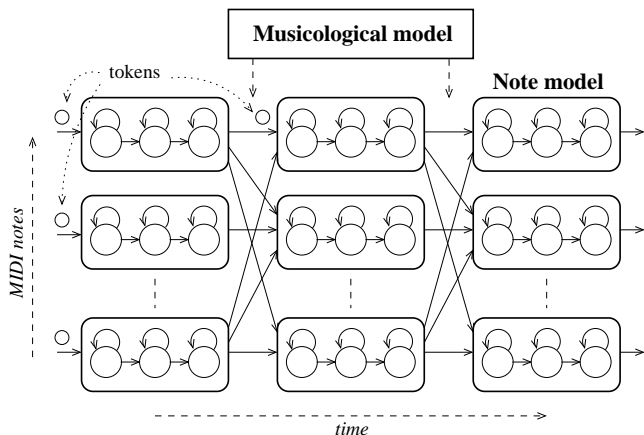


Figure 5: The combination of the note models and the musicological model.

Musical key is roughly defined by the basic note scale used in a song. The first note of the scale is called a *tonic note*. Here tonic notes $k_{\text{maj}}, k_{\text{min}} \in \{0, 1, \dots, 11\}$ where the values $0, 1, \dots, 11$ correspond to the major and minor keys with tonic notes C, C#, D, ..., B. If major and minor scales consist of the same notes, the scales are considered to be relative, thus defining a *relative-key pair* for which the tonic notes obey

$$k_{\text{maj}} = \text{mod}(k_{\text{min}} + 3, 12) \Leftrightarrow k_{\text{min}} = \text{mod}(k_{\text{maj}} + 9, 12), \quad (8)$$

where mod is the modulus-after-division operator. The musicological model first finds the most probable relative-key pair for the entire song and then uses this key pair to determine transition likelihoods for note sequences.

The most probable key pair is determined by using a musical key estimator proposed and evaluated in [5]. The method produces likelihoods for keys $k_{\text{maj}}, k_{\text{min}}$ from the voiced pitch estimates here rounded to the nearest MIDI note numbers in a performance. The likelihoods of relative major and minor keys are summed together and the key pair with the highest likelihood is chosen.

Note N -gram probabilities for $N \in \{2, 3\}$ were estimated by counting the occurrences of different note sequences in the EsAC melody database [12] of which over half a million note sequences were used. A note sequence is specified by N intervals (i.e., differences in the pitches of two notes) one of which is the interval between the first note of the sequence and the tonic note of the song, and the remaining $N - 1$ intervals are the intervals between successive notes in the sequence. The counted occurrence probabilities were smoothed with the Witten-Bell discounting algorithm [13], producing the probabilities for note bigrams and note trigrams given the key k of the song, i.e., $P(n_t = j | n_{t-1} = i, k)$ and $P(n_t = j | n_{t-2} = h, n_{t-1} = i, k)$. Given the previous note(s) and the relative-key pair, the likelihood to move to note j is the sum of likelihoods $P(n_t = j | \cdot, k_{\text{maj}})$ and $P(n_t = j | \cdot, k_{\text{min}})$ divided by two. If key estimation is disabled, we use the sum of note-sequence likelihoods over all keys divided by 24, i.e., the number of keys. If the musicological model is completely disabled, we use equal transition likelihoods for all note transitions.

3.3. Note model and musicological model combined

The note model and the musicological model constitute a probabilistic note network illustrated in Figure 5. The network consists of the note models and transitions between them controlled by the mu-

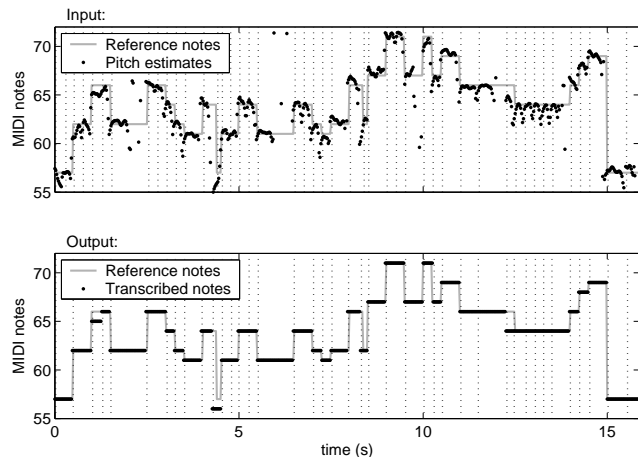


Figure 6: A melody transcribed with the system. The dotted vertical lines denote the reference note beginnings in the input panel and the transcribed note beginnings in the output panel.

sicological model. Notice that the figure represents the note models at every time instant even though there actually exists only one note model for each MIDI note (not for every time instant). The network is used to transcribe melodies by finding the most probable path through the network according to the likelihoods given by the note models and the musicological model. In particular, the between-note transitions are weighted by a scalar constant to match the dynamic ranges of the models.

The optimal path is found using the *Token-passing algorithm* [7]. The algorithm propagates *tokens* through the network. Each note model state increments the weight of the tokens by the observation likelihoods and the transition probabilities between the states. When a token is emitted out of a note model, we have a note boundary which is appended to a list of boundaries for the sake of backtracking later on. The musicological model increments the token weights at transitions between note models, considering the previous note models that tokens have visited. Eventually, the optimal note path is defined by the lightest token propagated through the network, and the corresponding note sequence is found by backtracking the list of note boundaries.

Figure 6 shows a transcription example. The upper panel shows pitch estimates extracted from a singing performance and the lower panel compares the transcribed notes with the reference notes. The system performs a reliable segmentation of notes as a consequence of using the without-skips note HMM. As an example, consider the time interval 11–13 seconds in the figure where a sequence of notes with identical pitches is correctly segmented.

4. Simulation results

The proposed melody-transcription system is evaluated using 57 melodies performed by two male and two female singers from the same acoustic database that was used to train the note models. The test signals were not included in the training set. For a more appropriate evaluation of the system, the test signals should be performed spontaneously without accompaniment and to use manual transcriptions as references. However, the current evaluation scheme punishes also for sung notes that clearly differ from the accompaniment reference or miss entirely which would be taken into account in manually transcribed references.

Different simulation setups are defined by varying the follow-

Feature set	E_f	E_n
$\{\Delta x, \nu\}$	10.3	10.3
$\{\Delta x, \nu, a\}$	9.2	9.6
$\{\Delta x, \nu, m\}$	10.3	9.4
$\{\Delta x, \nu, a, m\}$	9.1	9.9

Table 1: *The best results of each note-model feature set.*

ing parameters: (i) the number of note-model states K , (ii) the number of GMM components η for the note-model observation distributions, (iii) the used musical-feature set for the note model, (iv) the length of note N -grams, (v) enabling of the key estimation, and (vi) enabling the use of the musicological model.

4.1. Evaluation criteria

The transcription system is quantitatively evaluated by measuring the difference between the reference melodies and the transcribed melodies. Two evaluation criteria are used: a frame-based criterion and a note-based criterion.

The frame-based evaluation criterion is defined by the number of correctly transcribed frames c_{cor} and the number of voiced frames c_{ref} in the reference melody. A frame is considered to be correctly transcribed, if the transcribed note equals to the reference note in that frame. The evaluation database contains performances that are slightly unsynchronised in time compared to the reference melodies. This is compensated by considering two note values at the reference-note boundaries to be correct within a ± 50 ms distance from the note boundary (i.e., ± 2 frames). The *frame error* E_f for a transcribed melody is defined as

$$E_f = \frac{c_{\text{ref}} - c_{\text{cor}}}{c_{\text{ref}}} \cdot 100\%. \quad (9)$$

In contrast to the frame-based evaluation, the note-based evaluation criterion uses notes rather than frames as the evaluation units. The note-based evaluation is symmetrically approached from both the reference and the transcribed melodies' point of view. First, we count the number of reference notes that are *hit* by the transcribed melody and denote this number with \check{c}_R . A reference note is hit, if a note in the transcribed melody overlaps with the reference note both in time and in pitch. Second, the same scheme is applied so that the reference and transcribed melody exchange roles, i.e., we count the number of transcribed notes that are hit by the reference melody and denote the count with \check{c}_T . The *note error* E_n for a transcribed melody is then defined as

$$E_n = \frac{1}{2} \left(\frac{c_R - \check{c}_R}{c_R} + \frac{c_T - \check{c}_T}{c_T} \right) \cdot 100\%, \quad (10)$$

where c_R is the number of reference notes, and c_T is the number of transcribed notes. The frame and note errors are calculated for each individual melody in the evaluation database and the average of these is reported.

4.2. Results

The melody-transcription system achieved error rates below 10 % with both evaluation criteria. The best results for different feature sets are presented in Table 1. By using a simple rounding of pitch estimates to the nearest MIDI notes, the corresponding frame error in the database was 20.3 %, whereas the feature set with pitch difference and voicing $\{\Delta x, \nu\}$ achieved error percentages slightly over 10 %. When the accent feature a was included in the note

$K \backslash \eta$	1	2	3	4	5	6	crit
1	90.4	15.2	16.2	15.9	15.9	15.6	E_f
	64	19.5	18.4	18.6	18.3	18.2	E_n
2	72.8	12	13.9	13.2	13.1	13.2	E_f
	61.2	15.2	16.9	16	16.1	16.1	E_n
3	17.9	9.2	9.7	9.9	9.8	10	E_f
	18.1	10.2	10.6	10.4	10	10.2	E_n
4	18.8	9.4	9.8	10.1	9.8	10.2	E_f
	18.8	10.3	9.8	10	9.6	9.9	E_n
5	16.6	10.3	10.5	10.1	10.1	10.4	E_f
	15.3	11.2	11.2	10.4	10.4	10.4	E_n

Table 2: *Error rates for the feature set $\{\Delta x, \nu, a\}$ using key estimation and note bigrams.*

model, error percentages decreased by approximately one percentage unit for both the frame and the note error criterion. Further, replacing the accent feature with the meter feature m reduced note errors and achieved the best performance according to the note error criterion; however, frame errors were increased. On the other hand, the fourth feature set including all the proposed features reached the best performance according to the frame-error criterion. All the best-performance setups used key estimation and note bigrams.

Table 2 shows the error percentages when using the feature set $\{\Delta x, \nu, a\}$, key estimation, and note bigrams. The table shows clearly the influence of the number of note-model states K and the number of GMM components η . The first column of the table shows the cases where only one GMM component ($\eta = 1$) is used and the number of states is varied; the error percentages decrease when the number of states in the note model is increased. Similar trend can be observed by fixing the number of states to 1 and varying the number of components. However, increasing the number of states or GMM components does not significantly improve transcription results after using three or four states in the note model and two GMM components. The increase of components reduces the generality of the note model, and the increase of model states does not improve the temporal-separation accuracy of note events. The best results were achieved by using three or four states and two to five GMM components in the observation likelihood distributions.

Figure 7 shows the best results for each feature set when the musicological model was not used, key estimation was either enabled (on) or not enabled (off), and note N -gram length was either 2 or 3. Surprisingly, by disabling the musicological model, the system performed approximately as well as using note bigrams without key estimation, meaning that the good performance of the system is mostly a consequence of using the note model. Using key estimation in the musicological model clearly improves the system performance, but it was unexpected that the use of note bigrams with key estimation produces a slightly better performance than using note trigrams. An explanation for this could be that the trained trigrams are too specific to the training material (which differs from the test material) when they are used with key estimation. However, if key estimation was disabled, note trigrams performed better.

5. Conclusions

In this paper, a method was described for transcribing monophonic melodies. The method was based on a probabilistic note event model and a musicological model. The resulting system produced transcribed note sequences from acoustic inputs quite accurately.

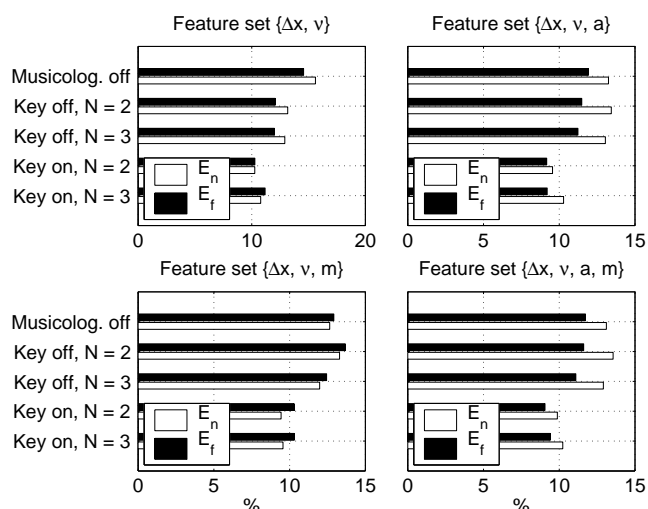


Figure 7: The influence of the musicological model and its parameters to the best results of each feature set.

The transcription system halved the amount of errors compared to a simple rounding of pitch estimates that was mostly due to the use of the note model. In addition, the note model segmented notes, enabling the possible quantisation of note lengths. Key estimation was important in reducing the amount of errors.

The transcription system can be easily extended in the future. New kinds of musical feature extractors can be used as a front-end to the note event model. The note model can be straightforwardly trained for other instruments than human voice, too, and the musicological model can be extended with new musicological rules. The system could transcribe polyphonic music by allowing several melodic lines to be handled simultaneously and by providing an appropriate front-end for multipitch estimation.

6. Acknowledgements

Timo Viitaniemi provided the acoustic melody database and collaborated at the early stages of developing the note model.

7. References

- [1] W. J. Hess, "Pitch and voicing determination," in *Advances in speech signal processing* (S. Furui, M. M. Sondhi, eds.), pp. 3–48, Marcel Dekker, Inc., New York, 1991.
- [2] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis* (W. B. Kleijn and K. K. Paliwal, eds.), ch. 14, pp. 495–518, Elsevier Science, 1995.
- [3] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111, pp. 1917–1930, April 2002.
- [4] L. P. Clarisse, J. P. Martens, M. Lesaffre, B. De Baets, H. De Meyer, and M. Leman, "An auditory model based transcriber of singing sequences," in *Proceedings of 3rd International Conference on Music Information Retrieval, ISMIR '02*, 2002.
- [5] 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, FINSIG '03*, pp. 59–63, May 2003.
- [6] H. Shih, S. S. Narayanan, and C.-C. J. Kuo, "A statistical multidimensional humming transcription using phone level hidden Markov models for query by humming systems," in *Proceedings of IEEE 2003 International Conference on Multimedia and Expo*, vol. 1, pp. 61–64, 2003.
- [7] S. J. Young, N. H. Russell, and J. H. S. Thornton, "Token passing: a simple conceptual model for connected speech recognition systems," tech. rep., Cambridge University Engineering Department, July 1989.
- [8] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. The MIT Press, 1983.
- [9] M. Ryyänänen, "Probabilistic Modelling of Note Events in the Transcription of Monophonic Melodies," Master's thesis, Tampere University of Technology, March 2004.
- [10] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the Meter of Acoustic Musical Signals," *IEEE Transactions on Speech and Audio Processing*, to appear.
- [11] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, pp. 257–289, February 1989.
- [12] E. Dahlig. EsAC database: Essen associative code and folk-song database, available at www.esac-data.org, 1994.
- [13] I. H. Witten and T. C. Bell, "The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression," in *IEEE Transactions on Information Theory*, vol. 37, pp. 1085–1094, July 1991.