A Graphical Model for Recognizing Sung Melodies

Christopher Raphael
School of Informatics
Indiana Univ.
Bloomington, IN 47408
craphael@indiana.edu

ABSTRACT

A method is presented for automatic transcription of sung melodic fragments to score-like representation, including metric values and pitch. A joint model for pitch, rhythm, segmentation, and tempo is defined for a sung fragment. We then discuss the identification of the globally optimal musical transcription, given the observed audio data. A post process estimates the location of the tonic, so the transcription can be presented into they key of C. Experimental results are presented for a small test collection.

Keywords: monophonic music recognition, graphical models

1 INTRODUCTION

The problem of automatic transcription of sung melodic fragments needs little justification or motivation within the music information retrieval community, since some form of this problem is the first step in any query-by-humming-type system. This community contains quite a few efforts that describe this recognition problem in various levels of detail including McNab et al. (1996), Haus and Pollastri (2001), Meek and Birhmingham (2002), Pauws (2002), Song et al. (2002), Clarisse et al. (2002), Pardo et al. (2002). Singing recognition has other applications such as the preservation of unnotated vocal music traditions and for speech-recognition-like interfaces to music notation software. We also find significant intellectual merit in this problem, independent of any applications, with its deep ties to human cognition and the associated modeling and computational challenges.

Music is an unusually organized and rule-bound domain when compared to other recognition domains such as speech or vision. In such a domain we are particularly inclined to use "Ockham's razor" as a guiding principle — given two hypotheses that explain the data equally well, we believe the simpler one to be more likely. We feel this criterion is particularly appropriate for music since, it seems to be consistent with human perception of music, while it is often straightforward to formalize the notion of simplicity for musical hypotheses. The idea of Ockham's razor is thoroughly embedded in much literature on recognition, including that in the music information retrieval community, and is often implemented through explicit penalty terms in optimization formulations or through the use prior distributions in probabilistic models. Examples of explicit penalties within the MIR community are Dixon (2001), Scheirer (1998) and Goto (2004) while examples of model-based penalties are Raphael and Stoddard (2003), Cemgil and Kappen (2003), and Abdallah and Plubley (2004).

Some notions of simplicity can be described without any knowledge of the deeper structure of music. For instance, a sung fragment is presumably composed of "notes" having fundamental frequencies that, given a tuning reference, are pitches in the chromatic scale. We expect comparatively few notes in a sung fragment, so a hypothesis that explains each frame of audio as the closest chromatic pitch is apt to explain the observed audio data well, but produce an unrealistically complex hypothesis. On the other hand, a hypothesis that groups contiguous regions of similar frames into notes will produce simpler hypotheses and is justifiable, even if the notes are somewhat "further" from the actual audio data.

All practitioners of machine recognition are likely to agree with this analysis so far, but the art of modeling lies, in large measure, in deciding how far to extend the idea. Continuing with the same example, the segmentation of the data into notes can be accomplished more accurately when the reference tuning is given, since then the possible note frequencies are no longer a continuum, but rather a small number of distinct and well-separated possibilities. So, clearly we are much better off if the tuning is known, but does this justify simultaneously estimating the tuning as well as the partitioning into notes?

This same question appears over and over in the recognition of melodic segments. For instance, if we know the key of the fragment, the likelihoods of various pitches changes dramatically, strongly favoring notes in the scale of that key. Does this justify simultaneously estimating the key?

The human's partitioning of audio data into notes usually occurs within a rhythmic framework in which inter-onset times are simple proportions of one another. While it is possible to partition audio data into notes just using pitch information, understanding the average length of the basic time unit, say beat or measure, allows us to capitalize on the basic rhythmic structure of music. Does this added knowledge justify the simultaneous estimation of beat length or tempo? As with pitch, there is considerably more rhythmic structure to music than the notion of simple proportions. Typically, music exists within a meter implying rather strong assumptions about how measures divide into notes. Should we incur the computational burden of simultaneous estimation of meter to increase the discriminating power of the model? Certainly there are other examples of this basic question.

In some contexts, the goal of recognition might be to learn these higher level constructs such as key, tempo, and meter. In these cases, it seems we have no choice other than including the constructs into the model. However, even if we only desire a segmentation into notes, we believe there is significant benefit to modeling these "nuisance parameters." People tend to be quite categorical in their perceptions of music: Intervals are heard distinctly as major thirds, octaves, etc. , even when the frequencies are not completely consistent. Similarly we tend hear rhythmic relations with definiteness even when not completely supported by the literal data. For instance, this note lies on the downbeat and this other is twice as long as the first. We believe it is the simultaneous existence of tempo, meter, key, harmony, phrase structure, motivic structure, and their interrelations, such as harmonic rhythm, that brings about this categorical perception. That is, within the context of these higher level constructs, the human will believe no other data interpretation "makes sense." For this reason, we believe that models including deeper levels of structure such as key, meter, and harmonic analysis, (even in monophonic fragments) have much greater power to discriminate accurately, even when the higher level constructs are not of interest.

We have suggested above that *simultaneous* estimation of these higher level constructs is the only alternative to simply forgetting about them, and, of course, this is not the case. Our bias for simultaneous estimation is that it circumvents the "chicken and egg" problem. For instance, one can't really estimate note value (quarter, eighth, etc.) accurately without having a notion of tempo and vice-versa. In general, simultaneous estimation is preferable when the joint knowledge of parameters leads to a much more definite data model than either parameter in isolation. For instance, scale degree and tuning standard combine to give a definite expectation of observed frequency that can't be realized without both parameters. In some cases it might be possible to "bootstrap" one's way up, adding more sophisticated structure to our interpretation with a series of successive recognition passes. When there is no chicken-and-egg problem, we are in favor of this approach, in spite of its messiness, and give an example in this paper.

This work should be viewed, in part, as an exploration of these ideas. We try to formulate the maximum amount of musically relevant information into our model that can be handled in simultaneous estimation. After the fact, we try to disambiguate further by estimating more structure. We are *not* trying to build a front end for any particular Query-by-Humming system. While we view the experimental results as promising, we believe that even deeper structure will lead to still better recognition as discussed later. Our approach differs significantly from the work cited above in its attempt to model the music at a significantly deeper level. We believe the informal results, while far from perfect, support this general line of research.

Specifically, the problem we address is as follows. We treat sung musical fragments with known time signature and mode: 3/4 time and major mode with a defined list of possible measure positions in our experiments. We simultaneously estimate the partition of audio data into notes, and the labeling of the notes with pitches and rhythmic values that make sense within the metric context. We also simultaneously estimate a (potentially) time-varying tempo process. The scheme we propose is capable of identifying the *globally* most likely configuration of these parameters, given the audio data. In a post-processing phase we further estimate the frequency of the tonic and relabel the recognized pitches within this context. This fixes some pitch errors and allows us to present all of the recognized results *automatically* transposed to the key of C major.

2 THE MODEL

We assume that the audio fragment to be recognized has a known time signature. While this assumption is certainly unrealistic for some examples, the audio recognition problem is difficult enough to warrant some simplifying assumptions. We further assume the possible rhythm positions are enumerated in a set $\mathcal{R}$ and model the sequence of note onset positions as a Markov chain.

To be more specific, suppose the fragment is in 3/4 time and that only note onsets beginning at eighth-note positions are deemed possible. Then the possible onset positions are described by the set

$$\mathcal{R} = \{\text{start}, \frac{0}{6}, \frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}, \text{tie}, \text{end}\}$$

We model the sequence of measure positions by a Markov Chain, $R_0, R_1, \ldots, R_K$ where $R_k \in \mathcal{R}$ that must begin in the start state and end in the end state. Thus we assume an initial distribution $P(R_0 = \text{start}) = 1$ and transition probability matrix

$$P(R_{k+1} = r_{k+1} | R_k = r_k) = Q(r_k, r_{k+1})$$

The tie element corresponds to a note that is tied over from the current measure to the beginning of the next measure and can thus be considered another "version" of the bar line position. Adding this element to our set of possible "states" allows us to model arbitrarily long notes without significantly increasing the size of the state space. We constrain the transitions so that $Q(\text{start}, \text{start}) = Q(\text{start}, \text{tie}) = 0$ and $Q(r_k, r_{k+1}) = 0$ when $J(r_{k+1}) < I(r_k)$ where

$$J(r) = \begin{cases} 1 & \text{if r = tie} \\ r & \text{otherwise} \end{cases}$$

$$I(r) = \begin{cases} 0 & \text{if r = tie} \\ r & \text{otherwise} \end{cases}$$

and $r_k, r_{k+1} \notin \{\text{start}, \text{end}\}$. This simply states that a note cannot cross the bar line without using the tie state, as in usual musical notation. The chain generates measure positions R_k until we reach the end state; we write $R_K = \text{end}$ so that K is the index of the final state. The modeling allows the rhythm to be unambiguously reconstructed from the sequence of states. For instance, the sequence start, $\frac{2}{6}$, tie, $\frac{3}{6}$, $\frac{0}{1}$, end corresponds to a rhythm beginning on the 2nd quarter of the measure which is tied over to a dotted quarter in the next measure followed by another dotted quarter and ending with a note on the downbeat of the following measure. We will write $R = (R_0, \ldots, R_K)$ and $r = (r_0, \ldots, r_K)$, and similarly for other vectors, for the collection of all rhythm variables and their actual values. Due to the Markov assumption, $P(R = r)$ factors as

$$P(R = r) = \prod_{k=1}^{K} Q(r_k, r_{k+1})$$

for sequences r starting in the start position and ending in the end position. Each transition, not including the start and end states has an unambiguous amount of musical time, in measures, it traverses, which we denote $l(r_k, r_{k+1}) = J(r_{k+1}) - I(r_k)$.

Associated with each measure position R_k is a pitch variable $P_k \in \mathcal{P} = \{\text{rest}, p_{\text{lo}} \ldots, p_{\text{hi}}\}$ giving either a rest or the MIDI pitch of the note that is sung during R_k to R_{k+1}. Without a key as reference it is difficult to give a probability distribution for the pitches. However, if we knew the tonic, we could design a reasonably informative distribution on pitches. In our first stage of recognition we assume a uniform distribution on pitches. In a later refinement we will estimate the location of the tonic and use a more refined pitch model. In both cases we use a "bag of notes" model, meaning the pitches are independent draws from some fixed pitch distribution. We write $B(p_k)$ for the pitch distribution.

Unlike the model for the measure positions and pitches, which are discrete, we model the sequence of onset times for the notes as a Gaussian process. For simplicity of notation, we prefer to measure time in terms of analysis *frames*, which are Δ-second-long sequences of audio samples on which we compute the FFT. Let $S_1, \ldots, S_{K-1}$ be the local tempo variables, given in frames per measure, and define $T_1, \ldots, T_{K-1}$ to be the sequence of actual note onset times, in frames. We model these variables jointly by

$$\begin{aligned} S_k &= S_{k-1} + \sigma_k & (1) \\ T_k &= T_{k-1} + l(R_{k-1}, R_k)S_k + \tau_k & (2) \end{aligned}$$

for $k = 2, \ldots, K - 1$. The $\{\sigma_k, \tau_k\}$ variables are 0-mean and Gaussian so the S process can be seen to be a random walk. This model has been used in Raphael (2004) and Cemgil (2004). If the $\{\tau_k\}$ variables were 0 then the note onset times would evolve exactly as predicted by the note lengths and tempo. The addition of the τ variables adds robustness to the model by allowing small deviations from what is predicted by the tempo and note length. The rhythm-conditional density for the tempo and onset variables is then

$$\begin{aligned} p(s, t|r) &= N(s_1; \mu_{S_1}, \sigma^2_{S_1})N(t_1|0, \sigma^2_{T_1}) & (3) \\ &\times \prod_{k=2}^{K} N(s_k; s_{k-1}, \sigma^2_{S_k}) & (4) \\ &\times \prod_{k=2}^{K} N(t_k; t_{k-1} + l(r_{k-1}, r_k)s_k, \sigma^2_{T_k}) & (5) \end{aligned}$$

where $N(\cdot; \mu, \sigma^2)$ is the normal density function with mean μ and variance σ^2. The variances $\{\sigma^2_{S_k}, \sigma^2_{T_k}\}$ can be allowed to depend on the amount of musical time traversed by the transitions, since, presumably, longer notes allow for larger increments in tempo and greater deviations from the expected length.

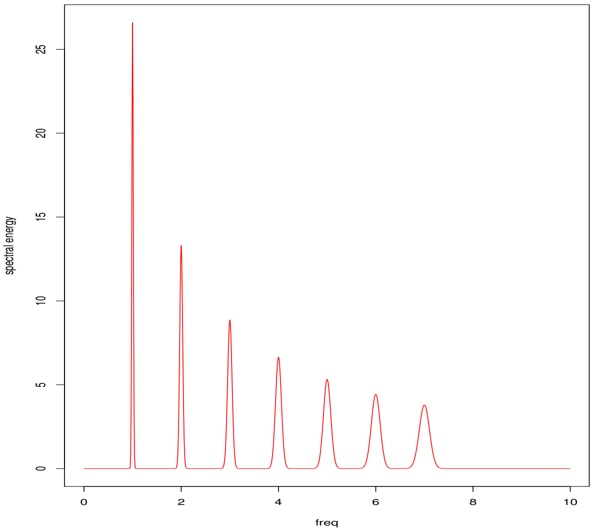

Figure 1: The distribution which generates the "spectral bits."

Finally, let $Y_1, \ldots, Y_N$ denote the frames of audio data each accounting for Δ seconds. If the note onsets are fixed ($T = t$) then these frames are partitioned into contiguous segments corresponding to the notes of the fragment. In particular, each frame, n, lies in segment $k(n)$ where $t_{k(n)} \leq n < t_{k(n)+1}$. We connect our hidden variables to the data by assuming that the $Y_1, \ldots, Y_N$ are conditionally independent given $T = t$ and $P = p$ so that

$$p(y|t, p) = \prod_{n=1}^{N} p(y_n|\pi(t, p, n))$$

where $\pi(t, p, n)$ is the pitch being sung at frame n. That is $\pi(t, p, n) = p_{k(n)}$.

To be specific, if $\pi = \pi(t, p, n)$ is the pitch being sung, we define the idealized power spectrum, f_π, as a superposition of peaks centered at the harmonics of pitch π as in Figure 1. f_π is assumed to be normalized to sum to unity. In defining our data model we treat the observed power spectrum in frame n, y_n as a histogram of a sample from the probability distribution f_π. That is

$$p(y_n|\pi) = c \prod_\omega f_\pi(\omega)^{y_n(\omega)}.$$

In the case in which the "pitch" is a rest, we take $f_{\pi=\text{rest}}$ to be a uniform model

Putting this all together gives a factorization of our model as

$$\begin{aligned}
p(r,p,t,s,y) &= p(r)p(p)p(s|r)p(t|r,s)p(y|p,t) && (6)\\
&= \prod_{k=1}^{K} Q(r_k, r_{k+1})B(r_k) && (7)\\
&\times N(s_1;\mu_{S_1},\sigma^2_{S_1})N(t_1|0,\sigma^2_{T_1})\\
&\times \prod_{k=2}^{K} N(t_k; t_{k-1} + l(r_{k-1},r_k)s_k, \sigma^2_{T_k})\\
&\times \prod_{k=2}^{K} N(s_k; s_{k-1}, \sigma^2_{S_k})\\
&\times \prod_{n=1}^{N} p(y_n|\pi(t,p,n))
\end{aligned}$$

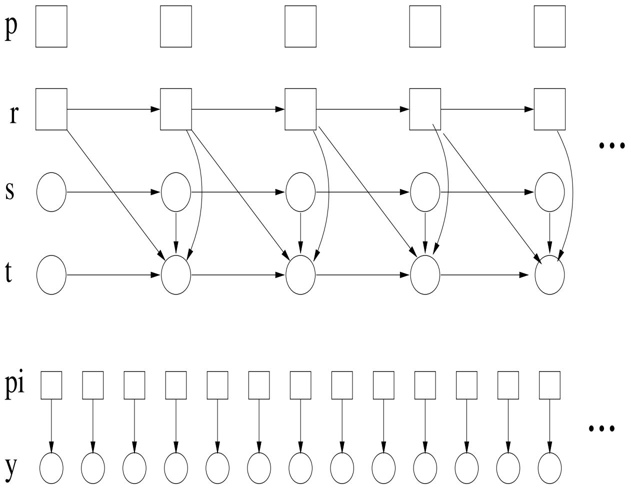

Figure 2: Description of the model as a directed acyclic graph. The top section of the model represents, from top to bottom, pitch (p) , rhythm (r) , tempo (s) , and onset times (t). The bottom section of the model gives the conditional distribution of the audio data (y) , given the labeling of the frames (π). Since the labeling π can be deterministically derived from t, p, we define a model $p(r, p, t, s, y)$

A graphical depiction of the model is given in Figure 2.

3 FINDING THE GLOBAL MAP CONFIGURATION

A rather surprising fact is that, given our spectrogram data y, the *globally* optimal configuration of the r, p, t, s, (rhythm, pitch, onset times, tempo) sequences can be computed using a variant of dynamic programming, under reasonable assumptions. We discuss here the computation of this global optimum

$$\begin{aligned}
(\hat{r},\hat{p},\hat{t},\hat{s}) &= \arg\max_{r,p,t,s} p(r,p,t,s|y)\\
&= \arg\max_{r,p,t,s} p(r,p,t,s,y)
\end{aligned}$$

Our approach is to construct a tree that, in principle, accounts for all possible configurations of the r, p, t, s sequences. In constructing this tree the continuously-valued note onset times, t, are only considered to only take integral values $t_k \in \{0, 1, \ldots, (N-1)\}$. A more fastidious description of the model of the previous section would have noted that the onset variables of Eqns. 4 and 5 are not actually normal, but rather a discrete approximation of normal evaluated only at the integers and further constrained so that $0 \leq t_1 < t_2 < \ldots, < t_K \leq N-1$.

A first observation is that, since there is no dependence among our pitch variables, $p_1, \ldots, p_{K-1}$, then given a configuration of onset times $t_1, \ldots, t_{K-1}$, the most likely configuration of pitches is simple to compute. For instance, the values t_1, t_2 specify that there is a note that begins at frame t_1 and ends at t_2 (as long as $r_1 \neq$ tie). Thus the optimal pitch associated with this region must be

$$\hat{p}_1 = \arg\max_{\pi\in\mathcal{P}} \prod_{n=t_1}^{t_2-1} p(y_n|\pi)$$

Thus fixing note boundaries automatically fixes the optimal choice of pitches, so we will leave the pitch variables out of the construction of our tree since they can be inferred from the onset frames. The computation that associates every possible sequence of frames with an optimal pitch can be performed before we begin the construction of our tree.

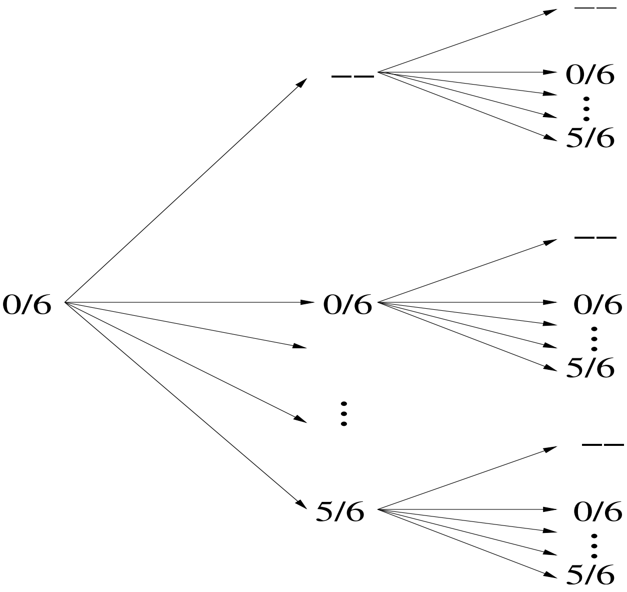

Figure 3: The tree describing the possible evolution of all rhythm sequences and partitions of the audio data.

The tree is constructed by specifying the rhythm variable from $\mathcal{R}$ for each frame of audio data. The first frame is labeled with the value start At each lower level in the tree we can either remain in the current note, the upper branch in Figure 3, or we can move on to a new note and choose a new value from $\mathcal{R}$, the lower branches in the tree. It is important to observe that, while the tree only specifies the possible sequences of R, other information is implicitly specified. First of all, a partial path in this tree fixes the frames at which rhythm transitions take place, therefore fixing the first several values of T. Furthermore, as noted above, fixing the note transition frames implies fixing the optimal choices of the pitch variables P. Thus, given our audio data Y, the only variables that are not fixed by choosing a tree branch are the local tempo variables, S.

Suppose we consider a branch of the tree at depth n, therefore a possible explanation of the first n frames of data $y_1^n = y_1, \ldots, y_n$. Suppose that in this branch the kth note begins on the nth frame. Thus the audio data y_1^n accounts for the variables $r_1^k, p_1^{k-1}, t_1^k, s_1^k$. Examination of Eqn. 7 shows that $p(r_1^k, p_1^{k-1}, t_1^k, s_1^k, y_1^n)$ is a product of constants and Gaussian density functions. Thus this probability can be expressed as the exponential of some quadratic function of the t_1^k and s_1^k variables. It is well-known that if one maximizes a quadratic function over several of the variables, the result is quadratic in the remaining variables. Thus

$$\begin{aligned} \max_{t_1^k, s_1^{k-1}} p(r_1^k, p_1^k, t_1^k, s_1^k, y_1^n) &= h e^{-\frac{1}{2}(s_k - m)^2/v} \\ &\stackrel{\mathrm{def}}{=} K(s_k; h, m, v) \end{aligned}$$

The details of how this maximization are performed are somewhat involved and can potentially distract one from the simple observation that the computation can be performed in closed form. Details are discussed in Raphael (2002) for a similar problem and model.

The above maximization gives the optimal probability of the branch as a function of the current tempo. We will store this function at every branch. In fact, it is relatively easy to compute the function recursively from the parent branch. In particular if $\hat{p}_b(s)$ is the optimal probability of the current branch b as a function of the current tempo s, Then for a child branch b', we have

$$\hat{p}_{b'}(s) = \hat{p}_b(s)$$

when no note transition takes place between b and b'. Otherwise, if a note transition takes place at level n of the tree, we move from rhythm position r to r', from the last note onset time t to the current time $t' = n$, and from the last (unknown) tempo, s to the current tempo s' by

$$\begin{aligned} \hat{p}_{b'}(s) &= \max_s \hat{p}_b(s) Q(r, r') B(\hat{\pi}) \\ &\times \; p(s'|s) \\ &\times \; p(t'|s', t) \\ &\times \; \prod_{\nu=t}^{t'} p(y_\nu | \hat{\pi}) \end{aligned}$$

where $\hat{\pi}$ is the optimal pitch for the interval (t, t').

At this point we seem to be faced with an exponentially growing tree, making the above process impossible to continue for more than a few levels of the tree. The surprising fact is that the tree can be pruned to a tiny fraction of its original size with no loss of optimality, using dynamic programming.

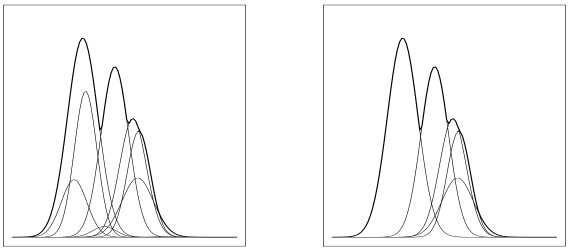

Figure 4: **Left:** The functions $\{\hat{p}_\beta(s)\}$ before **Right:** The reduced collection of functions after thinning.

Suppose we denote the collection of branches that begin a new note $\rho \in \mathcal{R}$ at level n of the tree by $B(\rho, n)$. If for one of these paths, $b \in B(\rho, n)$,

$$\hat{p}_b(s) \le \max_{\beta \in B(\rho, n)} \hat{p}_\beta(s)$$

for all s, there is no hope of b being a prefix to the optimal path, since for all values of the current state (ρ, n, s) some other path has a higher optimal probability. Thus we can prune b with no loss of optimality. We refer to this operation as the thinning operation, graphically depicted in Figure 4 and write $\mathrm{Thin}(B(\rho, n))$ for the surviving branches. It is easy to show that thinning can be performed with a computational complexity that is quadratic in the number of original branches.

We continue the construction of this tree with thinning until we reach level N. at this point it is easy to find the surviving branch with the best optimal probability ending with $\rho = \mathrm{end}$. This will be the globally optimal path and we can trace its history back to the root.

4 EXPERIMENTS

We now describe experiments with the analysis method described above. Our goal in conducting this research is to examine the problem of monophonic recognition from a deeper structural level than has previously done. In particular, we wish to see if the imposition of basic musical knowledge can be an aid to the recognition process, rather than to develop the best "front end" to a Query-by-Humming system. Thus, the experiments serve as a "course check," rather than a formal evaluation, and are well-suited to the exploratory nature of this work.

We collected a small test set of simple melodies in 3/4 time, all in major mode, sung by male voices. The melodies were sung by a non-random subset of the author's network of acquaintances. Several of the examples are "choruses" of male voices. The test set contained a total of 15 sound files. Our intention was to restrict our attention to the cases in which the musical content is unambiguous to the human listener. We believe these "cleaner" examples constitute the most interesting subset since the human is relatively certain of the correct hypothesis, while the examples still pose considerable problems for recognition. Thus these examples are well-suited for a study of the relation between knowledge representation and recognition results.

One improvement over the basic model we pursued concerns the role of the key of the excerpt. In the first pass of our algorithm we use a pitch model that gives equal probability to all chromatic pitches assuming an arbitrary choice of tuning. Not knowing the key leaves really no other reasonable choice. Even with what must be an occasionally inaccurate choice of tuning, our algorithm often does a reasonable job of segmenting the data into notes and ascribing rhythm. In a final phase, we "correct" the pitches by the following method.

We begin with a model for pitch distribution assuming the key of C major. This model is not estimated from data, but simply assumes that the notes in the tonic triad are the most likely, the notes in the scale are the 2nd most likely, and the remaining "black" notes are the least likely. We consider the data likelihood, assuming the given note segmentation, using 24 quarter-steps candidates for the tonic. For each tonic location we label each pitch with the choice that maximizes the pitch likelihood times the data likelihood. This has the effect of "nudging" ambiguous pitches toward plausible notes in the key. We choose the tonic location that maximizes this likelihood over all of the data, and call the tonic C. Thus all examples are automatically transposed to C major, no matter where they are sung. This method proves quite effective and identifies the correct key in all cases but the 1st example of "It Came upon a Midnight Clear." As it happens, the first phrase of the carol does not contain the 4th scale degree, thus making F a reasonable (or at least reasonably scoring) choice for the tonic. In addition to supplying useful information, the estimation of the tonic helps to correct notes whose actual frequencies are ambiguously placed. This is an example of how modeling of deeper musical structure can improve recognition results.

A number of the recognized examples incorrectly estimated the tempo by a factor or two or three. The former case amounts to representing the music in 6/8 rather than 3/4 with a 6/8 measure account for two 3/4 measures. This error is nearly inevitable at our current stage, since the distinction between these two meters requires a very deep musical understanding which goes beyond that represented in our current model. The one example, "Daisy," whose tempo was off by a factor of three is more puzzling. We suspect that early in the recognition process branches were mistakenly pruned that account for the correct tempo.

Several of the examples, "Happy Birthday," "God Save the Queen," and "Silver Bells" were recognized as "shifted" versions of the correct one. The distinction between these metrical shifts is also a subtle one, but it is demonstrably one that our model makes correctly most of the time.

The audio files as well as the transcriptions are available at http://xavier.informatics.indiana.edu/~craphael/ismir05.

References

S. Abdallah and M. Plubley. Polyphonic music transcription by non-negative sparse coding of power spectra. *Proceedings of the 5th International Conference in Music Informatics Retrieval*, 2004.

A. T. Cemgil. *Bayesian Music Transcription*. PhD thesis, Radboud University of Nijmegen, 2004.

A. T. Cemgil and H. J. Kappen. Monte carlo methods for tempo tracking and rhythm quantization. *Journal of Artificial Intelligence Research*, 18, 2003.

L.P. Clarisse, L.P. Martens, M. Lesaffre, B. DeBaets, H. DeMeyer, and M. Leman. An auditory model based transcriber of singing sequences. *Proceedings of the Third International Conference in Music Informatics Retrieval*, 2002.

S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1), 2001.

M. Goto. An audio-based real-time beat tracking sustem for music with or without drum sounds. *Journal of New Music Research*, 30(2):159–171, 2004.

G. Haus and E. Pollastri. An audio front end for query-by-humming systems. *Proceedings of Second Annual Symposium on Music Informatics Retrieval*, 2001.

J. McNab, I . H. Witten, C. L. Henderson, and S. J. Cunningham. Towards the digital music library: Tune retreival from acoustic input. *Digital Libraries*, 1996.

C. Meek and W. Birhmingham. Johnny can't sing: A comprehensive error model for sung music queries. *Proceedings of the Third International Conference in Music Informatics Retrieval*, 2002.

B. Pardo, W. Birmingham, and J. Shifrin. Name that tune: A pilot study in finding a melody from a sung query. *Jorunal of the American Socity for Information Science and Technology*, 55, 2002.

S. Pauws. Cubyhum: A fully operational query-by-humming system. *Proceedings of the Third International Conference in Music Informatics Retrieval*, 2002.

C. Raphael. A hybrid graphical model for rhythmic parsing. *Artificial Intelligence*, 137(1):217–238, 2002.

C. Raphael. A hybrid graphical model for aligning polyphonic audio with musical scores. *Proceedings of the 5th International Conference in Music Informatics Retrieval*, 2004.

C. Raphael and J. Stoddard. Harmonic analysis with probabilistic graphical models. *Proceedings of the Fourth International Conference in Music Informatics Retrieval*, 2003.

E. Scheirer. Tempo and beat analysis of acoustic musical signals. *J. Acoust. Soc. Am*, 103(1), 1998.

J. Song, S. Y. Bae, and K. Yoon. Mid-level melody representation of polyphonic audio for query-by-humming system. *Proceedings of the Third International Conference in Music Informatics Retrieval*, 2002.