PITCH SPELLING WITH CONDITIONALLY INDEPENDENT VOICES

Gabi Teodoru
School of Informatics
Indiana University
ateodoru@indiana.edu

Christopher Raphael
School of Informatics
Indiana University
craphael@indiana.edu

ABSTRACT

We introduce a new approach for pitch spelling from MIDI data based on a probabilistic model. The model uses a hidden sequence of variables, one for each measure, describing the local key of the music. The spellings in the voices evolve as conditionally independent Markov chains, given the hidden keys. The model represents both vertical relations through the shared key and horizontal voice-leading relations through the explicit Markov models for the voices. This conditionally independent voice model leads to an efficient dynamic programming algorithm for finding the most likely configuration of hidden variables — spellings and harmonic sequence. The model is also straightforward to train from unlabeled data, though we have not been able to demonstrate any improvement in performance due to training. Our results compare favorably with others when tested on Meredith's corpus, designed specifically for this problem.

1 INTRODUCTION

We consider here the problem of pitch spelling from MIDI, as has been addressed by several others, including Meredith, [1], [2], Cambouropoulos [3], [4], Chew and Chen, [5], Longuet-Higgins [6], and our previous work [7]. The goal here is to provide the pitch spellings (is it F♯ or G♭?) necessary to notate common practice music using a data source, such as MIDI, that does not distinguish between alternate spellings. The most immediate use for such an algorithm is to produce more readable music scores from MIDI data — at present, the pitch spellings from the commercial music notation programs we know of leave much to be desired. But pitch spelling will also be a part of the inevitable migration from MIDI, which, at present, constitutes the lion's share of symbolically represented music, to more expressive symbolic representations. While, perhaps, not as deep a problem as harmonic analysis, pitch spelling is the most obvious observable attribute of harmony — thus pitch spelling provides a means to quantify the accuracy of a harmonic analysis in objective terms.

We introduce a model that uses the notion of *conditionally independent voices*. That is, we model the musical voices as conditionally independent sequences, while depending on a common collection of key variables. More explicitly, we model the influence of harmony by a hidden Markov chain of local keys, one for each measure. Given the keys, the evolution of each voice in each measure occurs *independently* of the others, but is *dependent* on the key sequence. The voices are also modeled as Markov Chains. Thus we allow for interaction between the voices while clearly articulating the way in which this interaction occurs.

Our model assumes that there are two primary issues that explain pitch spellings: voice leading and local harmony. These two sources of information are articulated in the music theory text [8]. One principle therein suggests using accidentals to show the direction of chromatic passing tones, thus capturing the "yearning" ascribed to accidentals in informal discussions by musicians. (This same notion is also discussed by the Russian composer Nikolai Rimsky-Korsakov in [9].) Another principle from [8] advises avoiding "remote" accidentals; thus B♭ is preferred to A♯ in C major, since the former is in the scale of the F major, which is near to C. This principle is captured by the *key conditional* nature of our model, with its implicit notion of the likelihood of various pitches in different keys. Of course, there are times when these two principles come into conflict with each other, as in the spelling of chromatic scales. While notational conventions prescribe solutions here, and in other cases, our model can only explain the spelling in terms of local key and voice leading.

The most obvious distinction between our approach and the others mentioned is our formulation in terms of a generative probabilistic model. Within this context, we believe that the merits of various pitch spellings can best be weighed within the context of a hidden key, so we explicitly model key and simultaneously estimate this key sequence along with spelling. Furthermore, we clearly articulate our objective as the *globally* most likely configuration. All of the algorithms cited use some notion of "pitch closeness" in choosing spellings, as in Temperley's line of fifths and Chew's spiral array, though, in our case, this notion of closeness is in terms of the hidden key variable. These algorithms differ in their incorporation of voice leading. Temperley and Sleater, Meredith, and, to some extent Longuet-Higgins use voice leading, while Cambouropoulos, and Chew and Chen do not.

Our approach is computationally efficient while capturing both notions of horizontal and, to some extent, vertical interaction between voices. Our model is automatically

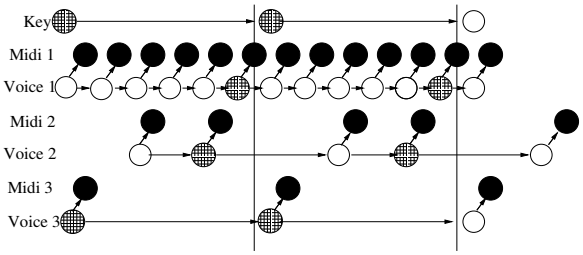

Figure 1. The hidden variables of the model, in open circles, are the Markov chain corresponding to the key sequence, along with conditionally independent Markov chains for each voice. The observable variable are the MIDI pitch classes denoted by solid circles. The key variables are directly related to each variable within the same measure, though we do not draw this in our figure for the sake of clarity.

trainable from unlabeled data, though it is unclear if EM-type training is appropriate for our situation, as discussed in a later section. We present results on the database collected by Dave Meredith as a testbed for several different pitch spelling algorithms, presented at ISMIR 05 [10]. Our overall error is lower than the best presented in [10], Meredith's ps1303 algorithm, by a factor of more than 3, though the results vary between composers.

2 MOTIVATION OF OUR MODEL

We assume we are given a partition of our piece into a collection of V *voices*. This partition will be obvious in the case of vocal music or monophonic instrumental music in which each part will be associated with a voice. However, the notion of voice is often meaningful in keyboard and other musical domains as well. Several efforts in the ISMIR community, [11], [12], [13], as well as our own, have addressed automatic voicing of music that does not contain explicit parts. Both [11], [13], have found this problem to be readily solvable by dynamic-programming-type algorithms seeking a partition that minimizes a cost function. We doubt that the results of the pitch-spelling algorithm discussed here are particularly sensitive to the way in which the data is voiced when the notion of voice is suspect.

We view our data as a collection of notes whose only pitch attributes we consider are the pitch classes, $\{0, \ldots, 11\}$, where the classes represent the *remainder* when the MIDI pitch is divided by 12. For instance C and B$\sharp$ would be represented by class 0, etc. We notate the pitch classes of the notes as $\{o_{mvi}\}$, where $m = 1, \ldots, M$ indexes the measures of the piece, $v = 1, \ldots, V$ indexes the voices, and $i = 1, \ldots, I = I(m, v)$ indexes the notes within the mth measure of the vth voice. While, of course, the number of notes in a particular voice and measure will be variable, for the sake of clarity we will suppress this dependence in our notation and simply write I for $I(m, v)$.

While the $\{o_{mvi}\}$ are the observable variables corresponding to our musical surface, we explain each variable as the result of two *hidden variables*, K_m and S_{mvi}. Here K_m is the local *key* of the piece — 12 possible tonics with a *mode* of either major or minor — and S_{mvi} is the *solfege* variable describing the scale degree with possible modifications. K_m and S_{mvi} are connected, since the domain of S_{mvi} depends on the key K_m. Explicitly, we have

$$S_{mvi} \in \{\text{rest}, \hat{1}, \ldots, \hat{7}, \sharp\hat{1}, \sharp\hat{2}, \sharp\hat{4}, \sharp\hat{5}, \sharp\hat{6}, \flat\hat{2}, \flat\hat{3}, \flat\hat{5}, \flat\hat{6}, \flat\hat{7}\} \quad (1)$$

for the major mode and

$$S_{mvi} \in \{\text{rest}, \hat{1}, \ldots, \hat{7}, \sharp\hat{6}, \sharp\hat{7}, \sharp\hat{1}, \sharp\hat{3}, \sharp\hat{4}, \flat\hat{2}, \flat\hat{4}, \flat\hat{5}\} \quad (2)$$

for the minor mode, in which the above notation of scale degrees is relative to natural minor — $\hat{6}$ and $\hat{7}$ refer to the sixth and seventh scale degrees of the natural minor scale, while $\sharp\hat{6}$ and $\sharp\hat{7}$ refer to the sixth and seventh scale degrees of the melodic minor scale. We assume that K_m and S_{mvi} *determine* the pitch class, which we denote as $o(k, s)$, in the straightforward way. For instance, $K_m = $ C major, $S_{mvi} = \sharp\hat{4}$ together imply pitch class 6 ($o_{mvi} = 6$), while $K_m = $ E minor, $S_{mvi} = \sharp\hat{6}$ say $o_{mvi} = 1$. That is, $o(\text{C major}, \sharp\hat{4}) = 6$ and $o(\text{E minor}, \sharp\hat{6}) = 1$.

The two different versions of some solfege variables, such as $\flat\hat{3}$ and $\sharp\hat{2}$, are used to distinguish between the *spellings* of the note — our eventual goal. Notationally speaking, any solfege variable with a sharp (flat) must be spelled by raising (lowering) the corresponding scale tone. For instance, if $\sharp\hat{4}$ appears in the key of C minor, whose scale degree $\hat{4}$ is F, then the note would be spelled as F$\sharp$. Notationally, the $\sharp$ accidental would be used only if it is necessary to "trump" the key signature, which may differ from that of the *local* key of the passage. Similarly, if $\sharp\hat{2}$ appears in E major, whose scale degree $\hat{2}$ is F$\sharp$, the note would be spelled as F *double sharp*. Notationally, this would always appear with the *double sharp* symbol unless the key signature actually had a double sharp. We hope to never see this latter situation!

While the observable pitch classes, o, depend deterministically on the K and S variables, the modeling of K and S is more interesting. Naturally, inspection of actual music data would uncover both vertical and horizontal dependence among the S variables. A relatively simple model would ignore horizontal dependence and treat, for a *fixed* measure m, the $\{S_{mvi}\}$ as a random sample from some distribution depending on the mode of K_m, $b(K_m)$. This distribution might give the highest probability to tonic triad notes, the second highest probability to the remaining scale notes, and the lowest probability to non-scale tones. Such modeling must take into account the mode of the key, since different non-scale tones exist for the two modes as in eqs 1, 2. We have described these key-conditional random sample models as "bag of notes" models in earlier work [14], and used them for harmonic analysis to reasonably good effect. A bag of notes model can be used for note-spelling, as in [7], by spelling each note using the local key and perhaps other variables — e.g. in D major, pitch class 6 will be spelled as F$\sharp$ rather than G$\flat$. However, such a model ignores the *voice leading*

tendencies, which describe horizontal motion — often an important issue in determining correct pitch spellings (e.g. $\sharp\hat{5}$ often moves to $\hat{6}$).

In this work we capture the harmonic nature of pitch spelling by modeling a sequence of hidden keys, one for each measure, as a Markov chain and allowing the voices to depend on this key. To capture the voice leading tendencies, we model each voice as a Markov chain whose transition probabilities capture voice leading patterns. The voices are assumed to be *conditionally independent*, given the key sequence, $K = k$, thereby assuming that all interaction between the voices is accounted for by K. This assumption allows us to partially decouple the voices during our computations.

As with all models, our assumptions oversimplify the true state of affairs; however, we do manage to capture what we expect to be the most important considerations. For instance, the Markov chain K attempts to estimate the time-varying key, which is usually the most important element for pitch spelling — other considerations aside, in D major F$\sharp$ is preferred to G$\flat$. However, the model is also able to capture the "inertia" of the accidental spellings, which have a strong tendency to resolve in the direction of their accidentals. Furthermore, key and spelling enjoy a kind of symbiosis which, we believe, enables each to help clarify the other. While the effect of key on spelling is rather obvious and has already been mentioned, spelling tendencies can help influence the choice of key. For instance, an alternation between pitch classes 6 and 7 in a measure may argue against the key of C major using "bag of notes" reasoning since both F$\sharp$ and G$\flat$ are unlikely in C major. However, when seen as $\sharp\hat{4}$ resolving to $\hat{5}$, a relatively common occurrence, C major becomes a much more reasonable hypothesis. Our approach capitalizes on this interplay between key and spelling by doing simultaneous recognition of both attributes.

2.1 The Model

A directed acyclic graph representation of our modeling assumptions is given in Figure 1, which shows a Markov chain for the key variables on top — one for each measure. The key variables influence every variable in the same measure, though these dependencies are not drawn in the graph for the sake of clarity. Once the key sequence is fixed, each voice evolves independently from the other voices as a Markov chain, but still depending on the key sequence. Thus the voices interact, but only through the key. Both key and solfege variable together determine the observable MIDI pitch classes, which we denote with solid circles.

Using the principles articulated so far, as well as a couple of others to be discussed, the joint distribution on K, S, and O, $p(k, s, o)$, can be factored as follows.

$$
\begin{aligned}
p(k,s,o) &= p(k_1)\prod_{m=1}^{M-1} p(k_{m+1}|k_m) && (3)\\
&\quad\times\prod_{v=1}^{V} p(s_{mv1}|k_{m-1}, s_{m-1vI}, k_m) && (4)\\
&\quad\times\delta_{o(k_m,s_{mv1})}(o_{mv1}) && (5)\\
&\quad\times\prod_{i=2}^{I} p(s_{mvi}|k_m, s_{mvi-1}) && (6)\\
&\quad\times\prod_{i=2}^{I} \delta_{o(k_m,s_{mvi})}(o_{mvi}) && (7)
\end{aligned}
$$

where

$$\delta_{o(k,s)}(o) = \begin{cases} 1 & o(k,s) = o \\ 0 & \text{otherwise} \end{cases}$$

Note the "nesting" of the products in these equations. The basic structure of Eqns. 3-7 is simply the result of the assumptions that the key variables are a Markov chain and that the voices are conditionally independent given the key sequence. Additionally we assume that the first solfege variable in a voice, s_{mv1} depends only on the two nearest key variables, k_m, k_{m-1}, as well as its predecessor, s_{m-1vI}. Similarly, we assume that within a measure $(i > 1)$ a solfege variable depends only on the current key, k_m, and its predecessor solfege variable, s_{mvi-1}.

These assumptions are further specialized in the following, in which we write $k = (t, b)$, where t is the tonic $(\in \{0, \ldots, 11\})$ and b the mode (major/minor) of the key. First the "within" measure transitions are modeled by

$$p(s_{mvi+1}|s_{mvi}, k_m) = p_S(s_{mvi+1}|s_{mvi}, b_m) \quad (8)$$

where the latter member $p_S(s'|s, b)$ depends on the previous solfege variable and the current mode. In addition, we further refine Eqn. 4 to:

$$
p(s_{mv1}|s_{m-1vI}, k_{m-1}, k_m) = \begin{cases} p_0(s_{mv1}|b_m) & \text{voice } v \text{ empty in meas } m-1 \\ p_T(s_{mv1}|s_{m-1vI}, k_{m-1}, k_m) & k_{m-1} \neq k_m \\ p_S(s_{mv1}|s_{m-1vI}, b_m) & \text{otherwise} \end{cases}
$$

Thus, there are three situations. When there is no obvious information regarding voice leading, as when a voice begins from scratch, we choose the first note of the voice from the distribution $p_0(s|b)$, which depends only on the mode of the key of the measure. When we have a key change, we use the transition probability, p_T, which can be quite simply parameterized, though we omit the details. Otherwise we do have useful voice leading information and follow the same assumptions of Eqn. 8 using $p_S(s'|s, b)$.

The key transition probabilities can also be simplified by forcing the translation invariant nature of key transi-

tions.

$$\begin{aligned} p(k_{m+1}|k_m) &= p(t_{m+1}, b_{m+1}|t_m, b_m) \\ &= p(b_{m+1}|t_m, b_m)p(t_{m+1}|b_{m+1}, t_m, b_m) \\ &= p_B(b_{m+1}|b_m)p_T(t_{m+1} - t_m|b_{m+1}, b_m) \end{aligned}$$

where $t_{m+1} - t_m$ is taken modulo 12. Here $p_B(b'|b)$ describes the probability distribution for mode transitions, obviously favoring staying in the current mode. Also $p_T(\Delta t|b, b')$ gives the probability of the *relative difference* in tonic, conditioned on the previous and next mode. Of course when the two modes are equal, as the most often will be, we will favor $\Delta t = 0$ — the key stays the same. Presumably, the probability of $\Delta t = 0$ will be less when the mode changes, though it may still be relatively high to capture the relative familiarity of moving between a major key and its parallel minor.

3 COMPUTING THE MOST LIKELY CONFIGURATION

In this section we exploit the conditional independence of the voices to find the most likely configuration of the hidden key and solfege variables, given the pitch class observations. The essential idea is the same as in the computation of the most likely configuration of a Bayesian belief network — we find a groups of hidden variables that "separate" the past and the future variables. Thus the probabilities of the various "paths" to such a group can be compared without consideration of future evolution. While presented for the sake of "full disclosure" and completeness, this section can be skipped without loss of continuity.

For each measure $m = 1, \ldots, M$, we define the vector X_m to be composed of K_m as well as all S_{mvi} variables (the hidden variables). Similarly, we define the vector o_m to be the collection of all pitch class observations o_{mvi} in measure m. In addition we partition the hidden variables, X_m, as

$$\begin{aligned} Y_m &= (K_m, S_{m1I}, \ldots, S_{mVI}) \\ Z_m &= \text{the remaining } S_{mvi} \text{ variables in } X_m \end{aligned}$$

so that the disjoint Y_m and Z_m together compose X_m. Figure 1 shades the Y_m variables for each depicted measure . Our algorithm finds the most likely hidden configuration by recursively computing the function

$$\begin{aligned} p^*(y_m) &= \max_{x_1,\ldots,x_{m-1},z_m} p(x_1, \ldots, x_{m-1}, y_m, z_m, o_1, \ldots, o_m) \\ &= \max_{x_1,\ldots,x_{m-1},z_m} p(x_1, \ldots, x_m, o_1, \ldots, o_m) \end{aligned}$$

using the essential ideas from dynamic programming. Each y_m configuration separates the "past" from the "future" in our model. That is, any path connecting variables on either side of y_m must contain a y_m variable. Exploiting such separations is always the core idea of dynamic programming.

To begin, we note that

$$\begin{aligned} p^*(y_1) &= \max_{z_1} p(x_1)p(o_1|x_1) \\ &= \max_{\substack{s_{111} \ \ldots \ s_{11I-1} \\ s_{121} \ \ldots \ s_{12I-1} \\ \vdots \ \ddots \ \vdots \\ s_{1V1} \ \ldots \ s_{1VI-1}}} \begin{matrix} p(k_1) \\ \prod_{v=1}^{V} p(s_{1v1} \ldots s_{1vI}|k_1) \\ \prod_{i=1}^{I} p(o_{1vi}|k_1, s_{1vi}) \end{matrix} \\ &= p(k_1) \prod_{v=1}^{V} \max_{s_{1v1}\ldots s_{1vI-1}} \begin{matrix} p(s_{1v1} \ldots s_{1vI}|k_1) \\ \prod_{i=1}^{I} p(o_{1vi}|k_1, s_{1vi}) \end{matrix} \\ &= p(k_1) \prod_{v=1}^{V} q^*_{1vI}(s_{1vI}|k_1) \end{aligned}$$

where we define

$$q^*_{1vi}(s_{1vi}|k_1) = \max_{s_{1v1}\ldots s_{1vi-1}} \begin{matrix} p(s_{1v1} \ldots s_{1vi}|k_1) \\ \prod_{j=1}^{i} p(o_{1vj}|k_1, s_{1vj}) \end{matrix}$$

for $i = 1, \ldots, I$. We can compute q^*_{1vi} recursively using the usual dynamic program argument:

$$q^*_{1v1}(s_{1v1}|k_1) = p(s_{1v1}|k_1)p(o_{1v1}|k_1, s_{1v1})$$

and

$$\begin{aligned} q^*_{1vi+1}(s_{1vi+1}|k_1) &= \max_{s_{1v1}\ldots s_{1vi}} \begin{matrix} p(s_{1v1} \ldots s_{1vi+1}|k_1) \\ \prod_{j=1}^{i+1} p(o_{1vj}|k_1, s_{1vj}) \end{matrix} \\ &= \max_{s_{1vi}} \begin{matrix} p(s_{1vi+1}|s_{1vi}, k_1) \\ p(o_{1vi+1}|k_1, s_{1vi+1}) \end{matrix} \\ &\quad \times \max_{s_{1v1}\ldots s_{1vi-1}} \begin{matrix} p(s_{1v1} \ldots, s_{1vi}|k_1) \\ \prod_{j=1}^{i} p(o_{1vj}|k_1, s_{1vj}) \end{matrix} \\ &= \max_{s_{1vi}} \begin{matrix} p(s_{1vi+1}|s_{1vi}, k_1) \\ p(o_{1vi+1}|k_1, s_{1vi+1}) \\ q^*_{1vi}(s_{1vi}) \end{matrix} \end{aligned}$$

Having computed $p^*(y_1)$ we can compute the general $p^*(y_m)$ recursively as well.

$$\begin{aligned} p^*(y_m) &= \max_{x_1,\ldots,x_{m-1},z_m} \begin{matrix} p(x_1 \ldots, x_m) \\ \prod_{\mu=1}^{m} p(o_\mu|x_\mu) \end{matrix} \\ &= \max_{x_1\ldots,x_{m-2},z_{m-1},y_{m-1},z_m} \begin{matrix} p(x_1 \ldots x_{m-1}) \\ \prod_{\mu=1}^{m-1} p(o_\mu|x_\mu) \\ p(x_m|y_{m-1}) \\ p(o_m|x_m) \end{matrix} \\ &= \max_{y_{m-1}} p^*(y_{m-1}) \max_{z_m} p(x_m|y_{m-1})p(o_m|x_m) \\ &= \max_{y_{m-1}} p^*(y_{m-1}) q^*_m(y_m|y_{m-1}) \end{aligned} \tag{9}$$

where

$$q^*_m(y_m|y_{m-1}) = \max_{z_m} p(x_m|y_{m-1})p(o_m|x_m)$$

Then q^*_m can be computed in terms of factors that depend only on the individual voices by

$$
\begin{aligned}
q^*_m \quad (\quad & y_m | y_{m-1}) \\
= \quad & \max_{z_m} p(x_m|y_{m-1}) p(o_m|x_m) \\
= \quad & \max_{\substack{s_{m11} \quad \dots \quad s_{m1I-1} \\ s_{m21} \quad \dots \quad s_{m2I-1} \\ \vdots \quad \ddots \quad \vdots \\ s_{mV1} \quad \dots \quad s_{mVI-1}}} \begin{array}{l} p(k_m|k_{m-1}) \\ \prod_{v=1}^{V} p(s_{mv1} \dots, s_{mvI} | \\ k_{m-1}, k_m, s_{m-1vI}) \\ \prod_{i=1}^{I} p(o_{mvi}|k_m, s_{mvi}) \end{array} \\
= \quad & p(k_m|k_{m-1}) \prod_{v=1}^{V} q^*_{mvI}(s_{mvI}|k_m, k_{m-1} s_{m-1vI})
\end{aligned}
$$

where

$$
\begin{aligned}
& q^*_{mvi}(s_{mvi}|k_m, k_{m-1} s_{m-1vI}) \\
& = \max_{s_{mv1} \dots s_{mvI-1}} \begin{array}{l} p(s_{mv1} \dots, s_{mvI} | k_{m-1}, k_m, s_{m-1vI}) \\ \prod_{j=1}^{i} p(o_{mvj}|k_m, s_{mvj}) \end{array}
\end{aligned}
$$

q^*_{mvi} can also be computed recursively with only minor revision of the derivation of q^*_{1vi}, though we have omitted the long but straightforward calculation for the sake of brevity.

Once the recursions have been computed to the end of the piece, we have $p^*(y_M)$ for all configurations of y_M. If we define $\hat{y}_M = \arg\max_{y_M} y^*(y_M)$, then from eqn. 9 we see that $p^*(y_{\hat{M}})$ is the probability of the *globally* maximizing configuration of hidden variables. It is a simple matter to "undo" our calculations to identify this global maximizer. We do this by substituting $\arg\max$ for $\max$ in eqn. 9:

$$\hat{y}_{m-1} = \arg\max_{y_{m-1}} p^*(y_{m-1}) q^*_m(\hat{y}_m | y_{m-1})$$

for $m = M - 1 \dots, 2$. Having found the optimal configuration for each of the Y_m variables, we can undo the q^*_{mvi} calculations to fill in the missing values of the Z_m variables.

4 TRAINING AND EXPERIMENTS

Since our model is a Bayesian belief network, we can train the models parameters using the usual junction tree and message passing paradigms. However, we found it simpler to adopt the familiar forward-backward (Baum-Welch) training algorithm to this particular case. Having implemented the training in this way, we were disappointed to see a slight degradation in our results when compared to hand-set parameters. However, this is not really surprising. Baum-Welch training is an example of the EM algorithm which seeks to maximize the *marginal* likelihood of the observed data, having integrated out over all unobserved variables. As many researchers have observed, this is not the criterion we are really interested in optimizing; we would prefer to minimize the number of errors on a training set. Especially with error rates as low as they are in this domain, it is not reasonable to expect an increase in performance using a model trained by EM. In essence, training gets "credit" for making more likely certain configurations that are already correctly recognized. With already low error rates, these already correct configurations may well dominate the learning process.

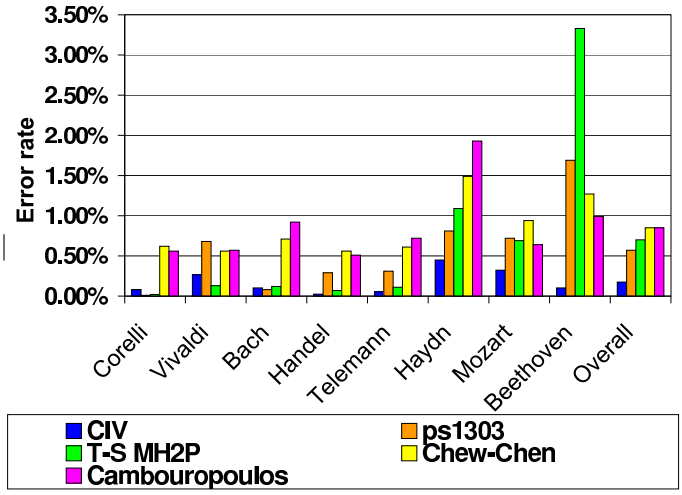

Figure 2. The error rates of the algorithms grouped by composer.

Composer	Error Rate
Corelli	0.081
Vivaldi	0.265
Telemann	0.053
Handel	0.024
Bach	0.102
Haydn	0.449
Mozart	0.322
Beethoven	0.102
Total	0.175

Table 1. Percentage Error rate for CIV algorithm

Meredith [10] compares the success rate of several pitch spelling algorithms. We have used the exact same corpus as [10] to judge the performance of our algorithm. Meredith's corpus contains 216 pieces by eight different composers, composed between the years 1680-1810. A key feature of this corpus is that it has the same number of notes for each composer, thus the number of movements or pieces per composer differs.

We compare the performance of our *Conditionally Independent Voices* (CIV) algorithm with the other algorithms from [10] using error rate as our criterion — that is, the number of misspelled notes divided by the number of total notes. We have used the overall error rate, as well as the error rate for each composer, to facilitate comparisons.

Meredith[10] reimplemented the algorithms as described by their authors. In this process, several versions of each algorithm were produced, considering from these the version giving the best performance on the test set. To

a small extent, this process tunes the algorithms on the test data, though we still expect the reported performance will generalize to other similar test data. In all cases the number of tuned parameters was quite small, making overfitting of the rather large data highly unlikely. The error rates we present for the algorithms of [10] are taken directly from this source.

As it may be noted from Figure 3, the CIV algorithm compares favorably with the other algorithms of [10] when aggregated over the entire corpus, as well as having the best result on 5 of the 8 composers. The precise errors for the algorithm are given in Table 1 by composer.

In these experiments, the parameters of our model were tuned by hand using a subset of the corpus which acted as our "training data", and included all of the Beethoven corpus (5 symphony movements), about a third of the Haydn corpus (5 string quartet movements), and another third of the Mozart corpus (one concerto movement). While it is not methodologically ideal to tune parameters on the test data, we had no choice but to do that, as we did not have other pitch spelled data, and needed to test on the entire corpus to compare our results with [10]. In our case, as with [10], the number of parameters was very small in comparison to the size of the test set. Thus we believe that our results will generalize to similar data as well as those of [10], making the comparisons valid.

Our algorithm has trouble correctly spelling a few harmonic situations such as: Ge+6 $\rightarrow$ Cadential 6/4 in major, $\mathrm{vii}^o 7$/V $\rightarrow$ Cadential 6/4 in major, secondary dominants and secondary leading tone chords in general, and also delayed resolution (G$\sharp$ $\rightarrow$ B $\rightarrow$ A instead of G$\sharp$ $\rightarrow$ A). We were expecting such errors since the premises of our model oversimplify the true state of affairs somewhat. As this last example shows, the resolution of a note may not always be the following note in the voice, thus thwarting our model, which only has "one-step" memory of pitch. Other situations require a deeper notion of the harmonic state than provided by the local key, as in the German augmented sixth chord, which seems nearly impossible to spell correctly without recognizing it as such. It seems, however, that simple voice leading tendencies often give the same result as a deeper harmonic analysis, thus explaining the success of our model. Finally, we anticipate that our algorithm might have problems with chordal figurations (arpeggiated chords) in which several voices are represented in a single voice. This might be fixed by preprocessing the data with an algorithm that would turn the figuration into voices. (We found this to be an aggravating tendency of our own voice recognition algorithm, which may well be an asset here!) It is worth noting that the Meredith test corpus contains almost no figurations.

5 REFERENCES

[1] Meredith D. "Comparing Pitch Spelling Algorithms on a Large Corpus of Tonal Music", *Computer Music Modeling and Retrieval, Second International Symposium, 2004* CMMR, Esbjerg, Denmark, 2004.

[2] Meredith D. "Pitch Spelling Algorithms", *Proceedings of the Fifth Triennial ESCOM Conference* pp. 204-207, Hanover, Germany, 2003

[3] Cambouropoulos E., "Automatic Pitch Spelling: From Numbers to Sharps and Flats," *Proceedings of VIII Brazilian Symposium on Computer Music* Fortaleza, Brazil, 2001.

[4] Cambouropoulos E., "Pitch Spelling: A Computational Model", *Music Perception* 20(4):411-429, 2003.

[5] Chew E., Chen, Y.-C. "Determining Context-Defining Windows: Pitch Spelling Using the Spiral Array", *Proceedings of the Fourth International Conference on Music Information Retrieval, ISMIR 2003* pp. 223-224, Baltimore, USA, 2003.

[6] Longuet-Higgins H. C., "The Perception of Melodies," *Mental Processes: Studies in Cognitive Science* pp. 105-129. British Psychological Society/MIT Press, London, England and Cambridge, Mass., 1987.

[7] Stoddard J., Raphael C., Utgoff P. "Well-tempered Spelling: A key-invariant Pitch Spelling Algorithm", *Proceedings of the Fifth International Conference on Music Information Retrieval, ISMIR 2004* pp. 106-111, Barcelona, Spain, 2004.

[8] Aldwell and Schachter, "Harmony and Voice Leading," 3rd Edition, pp. 505-508, Thompson Schirmer, 2003.

[9] Rimsky-Korsakov N., "Practical Manual of Harmony," 1886 translated from 12th Russian Edition by Joseph Achron, Carl Fischer, 1930.

[10] Meredith D., Wiggins G., "Comparing Pitch Spelling Algorithms", *Proceedings of the Fourth International Conference on Music Information Retrieval, ISMIR 2005* pp. 280-287, London, 2005.

[11] Lilian J., Hoos H., "Voice Separation — A Local Optimisation Approach", *Proceedings of the Third International Conference on Music Information Retrieval, ISMIR 2004* pp. 39-46, Paris, 2004.

[12] Cambouropoulos E., "From MIDI to Traditional Musical Notation", *Proceedings of the AAAI Workshop on Artificial Intelligence and Music* Austin, Texas, USA, 2000.

[13] Temperley D., "The Cognition of Basic Musical Structures", MIT Press, Cambridge, USA, 2001.

[14] Raphael C., "Harmonic Analysis with Probabilistic Graphical Models," *Proceedings of the Fourth International Conference on Music Information Retrieval, ISMIR 2003* pp. 177-182, Baltimore, USA, 2003.