Music Perception
Summer 2003, Vol. 20, No. 4, 411–429

Pitch Spelling: A Computational Model

EMILIOS CAMBOUROPOULOS
Aristotle University of Thessaloniki, Greece

In this article, cognitive and musicological aspects of pitch and pitch interval representations are explored via computational modeling. The specific task under investigation is *pitch spelling*, that is, how traditional score notation can be derived from a simple unstructured 12-tone representation (e.g., pitch-class set or MIDI pitch representation). This study provides useful insights both into the domain of pitch perception and into musicological aspects of score notation strategies. A computational model is described that transcribes polyphonic MIDI pitch files into the Western traditional music notation. Input to the proposed algorithm is merely a sequence of MIDI pitch numbers in the order they appear in a MIDI file. No a priori knowledge such as key signature, tonal centers, time signature, chords, or voice separation is required. Output of the algorithm is a sequence of "correctly" spelled pitches. The algorithm is based on an interval optimization approach that takes into account the frequency of occurrence of pitch intervals within the major-minor tonal scale framework. The algorithm was evaluated on 10 complete piano sonatas by Mozart and had a success rate of 98.8% (634 pitches were spelled incorrectly out of a total of 54,418 notes); it was tested additionally on three Chopin waltzes and had a slightly worse success rate. The proposed pitch interval optimization approach is also compared with and tested against other pitch-spelling strategies.

Received March 24, 2002, accepted February 15, 2003

ENHARMONICALLY equivalent notes usually correspond to identical pitch (intonation distinctions are possible on some instruments, but this phenomenon is not addressed in this study). This fact has led many researchers to consider the use of the traditional pitch representation system as redundant and to abandon it in favor of simpler chromatic representations. For instance, Parncutt maintains that "enharmonic distinctions may be regarded as superfluous–as notational artefacts" (Parncutt, 1999, p. 153) and advocates the introduction and evaluation of new chromatically based notation

Address correspondence to Emilios Cambouropoulos, Department of Music Studies, Aristotle University of Thessaloniki, Thessaloniki 54006, Greece. (e-mail: emilios@mus.auth.gr)

ISSN: 0730-7829. Send requests for permission to reprint to Rights and Permissions, University of California Press, 2000 Center St., Ste. 303, Berkeley, CA 94704-1223.

systems (Parncutt, 1999; Parncutt & Stuckey, 1992). He acknowledges, however, that there is an indirect link between enharmonic spellings and musical structure: "enharmonic spelling does not depend directly on musical meaning, nor does meaning depend directly on enharmonic spelling; the relationship between these is *indirect*, and mediated by context" (Parncutt, 1999, p. 154).

Longuet-Higgins (1976) takes a different stance on this issue. He maintains that notational orthography embodies important structural relationships between the tones of a musical piece and he suggests that it is of interest to the cognitive psychologist to study how a musical listener perceives these relationships and is able, in the case of musicians, to notate them "correctly" in the standard musical notation (as in the case of the "aural test" in which subjects write down a melody they have never heard before). He suggests that "the distinction between A♭ and G♯ [see Figure 1] is analogous to the difference between the homophones "here" and "hear" in English; even though these words sound exactly alike, they are interpreted and spelt quite differently according to the context in which they are heard" (Longuet-Higgins, 1976, p. 647). More recently, Temperley (2001) also advocates the use of enharmonic spellings—he introduces the term *tonal pitch class* to refer to enharmonic pitch names and proposes a preference rule system for tonal pitch class labeling.

Even though enharmonic spelling is not a *necessary* prerequisite for pitch perception (a listener does not have direct access to this information) it does, however, reflect underlying tonal qualities of pitch and, thus, may facilitate other musical tasks such as harmonic analysis, melodic pattern matching, and motivic analysis. Depending on the musical task at hand, a more refined and elaborate representation may be more efficient (despite its seeming redundancy at the lowest pitch level) as it allows higher level musical knowledge to be represented and manipulated in a more precise and parsimonious manner.

For instance, two enharmonic intervals in a tonal musical domain are different (perceptually and musicologically) although they consist of exactly the same number of semitones. The reason for this distinction lies in the structural properties that are assigned to each interval depending on the structural context in which it appears. For example, an isolated ascend-

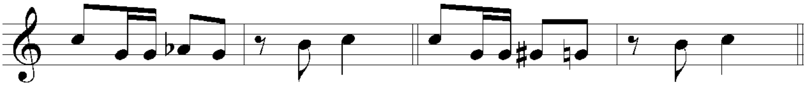


Fig. 1. A music student who transcribed the musical cliché as in the second version "would lose marks for having misinterpreted the tonal relation of the fourth note to its neighbours" (Longuet-Higgins, 1976, p. 647).

ing interval of 3 semitones is "normally" heard in the tonal domain as a minor third; if this interval is preceded and followed by an ascending semitone, it is recognized as an augmented second interval, as this specific sequence is encountered only between the sixth and seventh scale degrees of a harmonic minor scale (Figure 2a,b). As another example, a chord that is a stack of one 4-semitone and two 3-semitone intervals is "normally" recognized as a dominant seventh chord, but in some musical contexts it may be interpreted as a German sixth chord that has a very different effect (Figure 2c,d). Our mind, taking into account musical context, attempts to interpret low-level musical input and to place it within higher level tonal schemata. Explicit labeling of such enharmonic intervals that encapsulates higher level tonal properties may facilitate further musical analytic tasks (e.g., a harmonic analysis model may produce higher quality results more efficiently if it has access to enharmonic spelling of notes as in the case of an augmented sixth interval in a German sixth chord; or a melodic pattern matching algorithm may generate more reliable matches if it can distinguish between a minor third and an augmented second interval).

In this article, it is hypothesized that a strong link exists between enharmonic pitch spelling and tonal structure, and that this is primarily true for tonal music of the classical era (this hypothesis is probably only partially valid for later romantic music and invalid for atonal music). A second hypothesis is that listeners internalize frequencies of occurrence of musical intervals (e.g., a perfect fifth is a common interval whereas a diminished fifth or an augmented second are relatively rare), and such frequencies can be used effectively for inferring higher-level tonal information.

As a means to test these hypotheses, a simple algorithm has been developed, based on the frequency of occurrence of pitch intervals, that automatically transcribes MIDI pitch to enharmonically spelled notes. If the algorithm can notate successfully (according to the musical score) most of the input pitches, then we have some evidence that our first hypothesis is correct, that is, enharmonic pitch spelling is not arbitrary but it relates strongly to tonal pitch structure. Additionally we gain useful insights into possible cognitive principles and mechanisms—more specifically, "interval

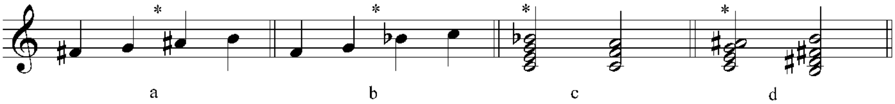


Fig. 2. The middle 3-semitone interval is perceived as an augmented second in example (a) and as a minor third in (b); the first chord (a stack of one 4-semitone and two 3-semitone intervals) is interpreted as a dominant seventh chord in (c) and as a German sixth chord in (d).

optimization" based on the frequency of occurrence of pitch intervals—that may be at work during the pitch transcription task.

The proposed algorithm transcribes polyphonic MIDI pitch files into the Western traditional pitch notation with a high success rate. The algorithm was tested on 10 complete piano sonatas by Mozart and had a success rate that is 98.8%; that is, 634 pitches were misspelled out of a total of 54,418 notes (14,070 notes required sharps or flats). Additionally, the algorithm was tested on three Chopin waltzes and had a success rate 95.8%, that is, 202 pitches misspelled out of a total of 4876 notes (2955 notes required sharps or flats). No a priori knowledge such as key signature, tonal centers, time signature, voice separation, or alignment of notes into chords is required. The input data is merely a sequence of MIDI pitch numbers in the order they appear in a MIDI file. The output of the algorithm is a sequence of "correctly" spelled pitches.

In the first part of this article, some theoretical issues regarding pitch intervals are discussed and two existing approaches to the pitch-spelling problem are presented. In the second part, the spelling algorithm is described and some detailed evaluation and comparison tests are presented. In the course of this discussion, musical examples are given that highlight various aspects of the pitch-spelling task.

In terms of the comparisons and tests given next, it should be noted that the aim of this article is not to compare Temperley's algorithms (1997, 2001) with the current pitch-spelling algorithms but rather to compare the "line of fifths" model on which Temperley's algorithm is based with the "interval optimization" approach on which the proposed algorithm is based. For this purpose, a single algorithm that embodies the two different approaches is proposed. This way, a more controlled environment is created in which the effectiveness of each method can be tested more objectively (i.e., everything remains identical in the program except the order of intervals since this is implied by the line-of-fifths model and by the proposed interval classification).

Pitch Spelling and Interval Optimization

Pitch spelling can be seen as a process that naturally follows the application of key-finding algorithms (Bharucha, 1987; Krumhansl, 1990; Longuet-Higgins & Steedman, 1971; Vos & Van Geenen, 1996). For instance, Longuet-Higgins and Steedman (1971) suggest, in a bold statement, that after a key is determined and relations of nonkey notes to the main key notes are established "it is a trivial matter to transcribe the solution into standard musical notation." Krumhansl (1990) states that "once a key (or key region) has been determined, the correct spellings of the tones will be able to be determined in most cases" (p. 79). Rowe (2001) proposes a spelling algorithm that is based on a stacked-thirds technique and also requires some tonal preprocessing, namely, that the root of each chord be determined in advance (this algorithm implements Parncutt's root determination algorithm—Parncutt, 1997).

It is less common to have pitch-spelling algorithms used as a precursor to harmonic analysis. One such case is Temperley's algorithm; he states that one of the main claims of his model is that "spelling can be accomplished without relying on 'top-down' key information" (Temperley, 2001, p. 126). Of course, the tasks of spelling pitches and key finding are strongly linked since they both relate in one way or another to properties of diatonic scales and more generally to the hierarchic organization of pitches in a tonal system. It is interesting, however, to explore the possibilities of notating a musical score correctly without having access to established tonal regions and keys. It would also be interesting to attempt to develop key-finding algorithms that take as input "correctly" spelled pitches (e.g., by "counting" sharps and flats and matching these against known templates of accidentals for all the major and minor keys, or by taking advantage of the anchoring effects of "rare" intervals for finding tonal centers).

A pitch-spelling algorithm that has been developed by the author (Cambouropoulos, 1996) selects appropriate traditional pitch names on the basis of a transcription procedure that optimizes the "quality" of traditional intervals; that is, it avoids diminished, augmented, and chromatic intervals—at this early stage, the algorithm was applied only to monophonic pitch sequences even though its extension for polyphonic music is a rather straightforward process as will be shown later.

Another pitch-spelling algorithm described by Temperley (1997, 2001) spells pitches so that they are as close as possible together on the "line of fifths" (Figure 3). Temperley's algorithm uses a "center of gravity" method that finds the mean line-of-fifths position of all previous pitches weighted for recency and then minimizes the line-of-fifths distance of each new event to the center of gravity. Additionally, his model has two other rules that make use of higher level musical knowledge: the voice leading rule and the harmonic feedback rule. In the current study, only the fundamental line-of-fifths principle that is the basis of this algorithm is examined.

What is the relation between the line-of-fifths approach and the interval optimization approach? Is there a common underlying principle? Which method is more effective? These issues will be addressed in the following

... B𝄫 F♭ C♭ G♭ D♭ A♭ E♭ B♭ F C G D A E B F♯ C♯ G♯ D♯ A♯ E♯ B♯ Fx Cx...

Fig. 3. The "line of fifths" is a stretched out "circle of fifths" (its pitch elements are referred to as *tonal pitch classes* as opposed to the 12 *neutral pitch classes*).

paragraphs. It will be maintained that the "line of fifths" approach is actually a special case of the general "interval optimization" approach. It will be shown that although both approaches are very effective, the proposed interval classification yields overall better results.

Implicit in the principle of using the narrowest possible line-of-fifths spelling band is a hierarchic ordering of pitch *intervals*: tonal pitch classes closer together are preferred and so are the corresponding intervals they form. Temperley (2001) indirectly acknowledges the importance of intervals, suggesting that the tonal pitch class representation is "relative rather than absolute: what is important is the relative positions of events on the line of fifths, not their absolute positions" (Temperley, p. 127). He does not, however, take the step to introduce tonal pitch class intervals as the basis on which to develop his pitch-spelling algorithm.

The line of fifths implies the ordering of pitch intervals that is depicted in Table 1. For instance, two adjacent tonal pitch classes (one step apart in the line of fifths) form a perfect fifth or perfect fourth, two tonal pitch classes that are in a distance of two steps form a major second or minor seventh, and so on. The pitch-spelling algorithm proposed by Temperley (1997, 2001) effectively avoids *intervals* that result in greater distances between pitches in the line of fifths (i.e., avoids intervals toward the right end of Table 1).

Cambouropoulos (1996) has proposed a hierarchy of pitch intervals according to their frequency of occurrence among the degrees of the major-minor scales (i.e., major scale, and harmonic and melodic minor scales). Perfect intervals that are the most frequent intervals form class A, major and minor intervals form class B, rare intervals such as many augmented and diminished intervals form class C, and intervals not encountered between scale degrees (e.g., augmented and diminished eighth, augmented third, diminished sixth) form class D (see Table 2). This classification seems to be in agreement with music theoretic approaches whereby rare intervals have a special status/function in tonal music (Brown, 1988; Browne, 1981; Butler & Brown, 1984) and class D intervals have "little beyond a theoreti-

TABLE 1
Ordering of Pitch Intervals According to the Distance of Their Constituent Pitches in the Line of Fifths

Distance	1	2	3	4	5	6	7	8	9	10	11	12
Intervals	P4 P5	M2 m7	m3 M6	M3 m6	m2 M7	A4 d5	A1 d1	d4 A5	A2 d7	d3 A6	A3 d6	d2 A7

P = perfect, M = major, m = minor, a = augmented, d = diminished, intervals 1a/1d is the chromatic semitone interval or augmented/diminished octave.

Table 2
Classes of Pitch Intervals Ordered According to Their Frequency of Occurrence in the Major-Minor Scale Framework

Class	A	B				C				D		
Intervals	P4	m2	M2	m3	M3	A2	d3	d4	A4	d1	A3	d2
	P5	M7	m7	M6	m6	d7	A6	A5	d5	A1	d6	A7

Note—There is no ordering of intervals within the same class.

cal existence" (*The Oxford Dictionary of Music,* 2nd ed., 1994). The proposed pitch-spelling algorithm prefers intervals from classes A and B, whereas the intervals of class D are most strongly avoided.

Regarding intervallic rarity, the following should be noted without entering a long discussion: studies such as those by Browne (1981), Butler and Brown (1984), and Brown (1988) identify the minor second and the tritone as rare intervals with a special status in the diatonic set. In the current study (also Cambouropoulos, 1996), augmented and diminished intervals are considered rare (Class C and D) whereas the minor second (Class B) is not included in the set of rare intervals. Referring to a statistical analysis of pitch interval frequencies, Krumhansl (1990) states that "minor seconds . . . are relatively frequent in the sample as a whole" and referring to the tritone that "this 'rare' interval is indeed rare" (p. 135). Whether the minor second is a rare interval or not is not the topic of this article. It is simply herein hypothesized that the chromatic semitone is a "truly" rare interval (Class D) whereas the enharmonically equivalent minor second is more common (Class B). In any case, there is agreement that relatively "rare" intervals play a special role in tonal perception (e.g., as distinctive cues for key-finding).

Within a general interval optimization framework, the line-of-fifths model is a special case in that it provides one possible ordering of pitch intervals. Another possibility is the four-level classification just described (many other possibilities exist). The main issue here is to determine which ordering of pitch intervals is most adequate for pitch-spelling algorithms. From a theoretical perspective, the line-of-fifths model is attractive because of its elegance (geometric models are usually economic and elegant) and, also, of its acoustic-based origin (the perfect fifth interval); the frequency-of-occurrence model, on the other hand, is attractive because it is grounded on a stronger cognitive basis (listeners internalize simple statistical interval information through extended exposure to musical material—Krumhansl, 1990). In this article, in more practical terms, the two aforementioned pitch interval hierarchies are compared and tested within a context of a single simple algorithm and against the same musical data set. Further research, however, is required for establishing the most appropriate ordering *in general* (if a single one exists!).

The pitch interval ordering that is based on the line of fifths is the same as the frequency-of-occurrence ordering for the perfect, major, and minor intervals. Some differences emerge in the ordering of the augmented and diminished intervals. For instance, the diminished and augmented first/eighth (e.g., chromatic semitone) is preferred over the diminished third and augmented sixth, or the augmented second and diminished seventh. Such differences can have a significant effect on the transcription process. Consider, for instance, the example in Figure 4; the first spelling (A) of the four pitches in each staff is given by the frequency-of-occurrence ordering (diminished third and augmented sixth preferred) whereas the second spelling (B) is given by the line-of-fifths algorithm (chromatic semitone and augmented eighth preferred). Of course, spelling depends on a broader context, but it would seem more plausible that the first spellings (A) are more adequate for a classical tonal context.

It is important to establish an ordering between the alternative *enharmonic* spellings of intervals (whether a minor third should be preferred over an augmented second) but not among different size intervals (no need to set any preferences among the various perfect, minor, and major intervals). The ordering implied by the line of fifths seems to be overrefined, at least for pitch-spelling tasks. Roughly, two preference categories should be sufficient (see Table 3). It should be noted, however, that preferences among the "less preferred" row of intervals in Table 3 becomes very important when intervals of different sizes are considered at the same time as they compete against each other during the optimization process (e.g., if in one transcription an augmented second occurs and in an alternative transcription of the same musical excerpt a diminished sixth occurs, the model should be able to make a final selection between the two).

In the present study, no distinction is made between melodic and harmonic intervals. Despite the fact that the frequency of occurrence between melodic and harmonic intervals is obviously different (e.g., the second in-

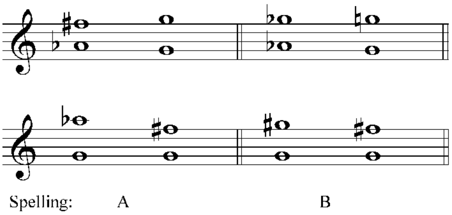


Fig. 4. Two sets of pitches (one in each staff) are spelled according to the frequency-of-occurrence ordering of pitch intervals (spelling A) and to the line-of-fifths ordering (spelling B).

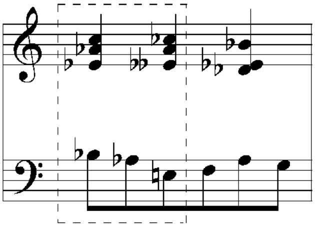
Table 3
Pitch Interval Preference Categories for Pitch Spelling

	Number of Semitones					
	1 or 11	**2 or 10**	**3 or 9**	**4 or 8**	**5 or 7**	**6**
Preferred	m2/M7	M2/m7	m3/M6	M3/m6	P4/P5	A4/d5
Less preferred	A1/d1*	d3/A6	A2/d7	d4/A5	A3/d6*	–

Note—Intervals indicated by asterisks may form a third even less preferred category (see text).

terval is the most frequent melodic interval whereas this interval is a rather uncommon harmonic interval), it is herein assumed that this does not pose a serious problem. The reason is that no distinction is made between interval types within the same category (e.g., diatonic intervals are simply considered as being frequent without any refined preferences such as second over third intervals or vice versa). The assumption is that diatonic intervals are more frequent than category C intervals (e.g., augmented second or diminished fifth), which in turn are more frequent than category D intervals (e.g., chromatic semitone or augmented sixth; see Table 3). However, this assumption is not without problems—the section Further Improvements and Applications includes a discussion of voice-leading concerns that have not been taken into account in the current approach.

Finally, it should be noted that, in the context of this study, doubly augmented and doubly diminished intervals have not been considered because they are extremely rare (e.g., E-E𝄫 in Figure 5). This way, for each interval, the choice is between two enharmonic interval categories (except for the tritone for which the choice is between the equally preferred augmented fourth and diminished fifth intervals).



MIDI pitch sequence (square region): [72, 68, 63, 58, 56, 52, 62, 68, 71]

Fig. 5. Measure 108 from Chopin's Waltz, Op. 64, No. 3. The composer uses an E natural concurrently with an E𝄫 (see middle chord); such extremely unusual intervals are disallowed by the algorithm. The program makes two mistakes in the transcription of this excerpt: D instead of E𝄫 and B instead of C♭.

The Pitch-Spelling Algorithm

In this section, a simple version of the proposed algorithm is described and evaluated; further extensions and refinements are also suggested. The proposed algorithm has been kept as simple as possible because the main aim is, on one hand, to study the interval optimization process and, on the other hand, to provide a very easy-to-implement algorithm. The design of the algorithm draws insights from research in music cognition, and various mechanisms have cognitive relevance; however, the goal of the algorithm is not to simulate the *exact* cognitive mechanisms by which a listener/musician is able to perceive and to notate correctly a heard musical excerpt in the standard musical notation.

THE ALGORITHM

Input to the algorithm is a string of MIDI pitch values in the order they appear in the original MIDI file. The algorithm can be applied both on quantized and performed (nonquantized) pieces without any modification. No a priori knowledge is required such as vertical organization of pitches into chords, horizontal organization into voices, key signature, time signature, tonal centers, and so on.

The algorithm uses a shifting overlapping windowing technique (Figure 6). All the pitches in each window are spelled according to certain criteria (listed below), but only the ones in the middle one-third section of the window are retained (suggested size of window is 9 or 12 pitches). Then, the window is shifted by one third and the same process is applied recursively until the end of the pitch sequence is reached. Allowing a larger section to be spelled in each step (preserving only the middle section) gives greater stability to the pitch-spelling process because a larger pitch context is taken into account and abrupt changes at the edges of the window are avoided. The pitch context includes both preceding and subsequent notes; using subsequent notes accounts for the fact that a spelling option may be revised when new notes are received and interpreted by a listener (e.g., the A♯ in the German sixth chord of Figure 2d can be assigned only in retrospect).

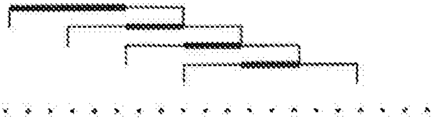


Fig. 6. Shifting overlapping window technique. For each window only the middle section of spelled pitches (bold line) is retained. Dots represent the pitches of the input sequence.

For each window, the pitch-spelling process is based on an optimization procedure that relies on two fundamental principles:

1. *Notational parsimony* (i.e., spell notes making minimum use of accidentals)
2. *Interval optimization* (i.e., avoid augmented and diminished intervals of classes C and D).

The first principle essentially avoids enharmonic spelling of notes that can be notated without any accidentals, e.g., C is preferred over B♯ and D𝄫. The second principle attempts to spell notes in a way that the more frequent diatonic intervals are used (i.e., perfect, major, and minor intervals) and the augmented and diminished intervals are avoided.

Penalty values are introduced for the notational parsimony principle and for the different categories of intervals presented in Table 2:

Notational parsimony	
“Normal” spelling of note	0
Enharmonic spelling of note	2
Interval optimization:	
Intervals of class A or B	0
Intervals of class C	1
Intervals of class D	2

Some examples of interval penalty values according to these principles are presented: for the interval F♯-G♯, the penalty value is $p = 0$; for F-G♯, $p = 1$ (augmented second interval); for E♯-G♯, $p = 2$ (enharmonic spelling of first note); for E♯-G, $p = 3$ (enharmonic spelling of first note and interval of category C, i.e., diminished third); and for E-E♯, $p = 4$ (enharmonic spelling of second note and interval of category D, i.e., chromatic semitone). The above penalty values are not absolute but rather indicative of the possible contribution of each factor in the selection process (the specific values have been selected after trial-and-error optimization on the Mozart and Chopin files). The relative strengths between the two principles is crucial in allowing more or fewer enharmonic spellings of notes—this may depend on musical style (e.g., Chopin makes heavier use of enharmonic spellings than Mozart does).

For each window, *all* possible spelling sequences are computed—sequences that contain *both* double sharps *and* double flats are disallowed by creating the different spelling sequences only from within two different tonal-pitch-class areas wherein one area excludes double sharps and the other excludes double flats (see dotted areas in Figure 7). This way, computational efficiency is also improved (worst case is $2 \cdot 2^n$ spelled sequences where n is the window size—for $n = 9$ there are 1024 sequences).

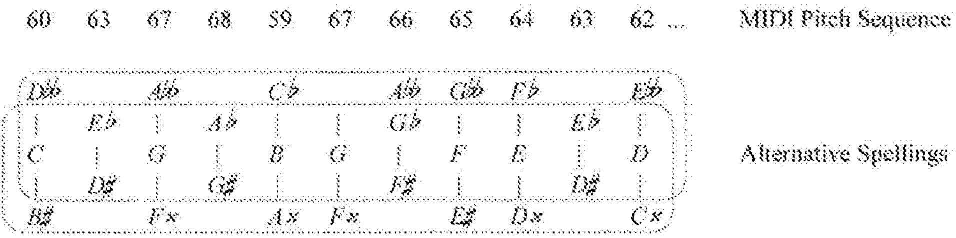


Fig. 7. Beginning of the theme of Bach's *Musical Offering*. Spelling sequences are selected from within the two different boxes of the available tonal pitch classes.

For each spelled pitch sequence in the window, all the penalty values given above for every possible interval (i.e., *all* permuted intervals between contiguous and noncontiguous pitches in the sequence—number of intervals is equal to $n \cdot (n - 1)/2$) are summed and an overall penalty value is computed. The sequence with the lowest penalty value is selected. For instance, in Figure 5 a window of size $n = 9$ is illustrated (dotted square region) along with the corresponding MIDI pitch sequence—the algorithm optimizes the interval content of this window for all intervals that are delimited by all the possible combinations of the nine pitches (the algorithm makes no distinction between intervals formed by notes belonging to contiguous or noncontiguous sonorities, nor is there any distinction between melodic, harmonic, or "diagonal" intervals—such information is not taken into account because input to the algorithm is merely an unstructured sequence of MIDI pitch values).

The implementation of the spelling algorithm in this study actually employs a special technique in order to provide more stability to the transcription process. More specifically, in each window, the first three notes retain the spellings defined in the immediately previous window of the algorithm, that is, for these pitches, the spellings are fixed and all possible pitch spellings are only tried out for the remaining six pitches. This way, it not only becomes more difficult for the algorithm to flip over from the flat region to the sharp region and vice versa, but also the efficiency of the algorithm increases as the actual number of pitches to be spelled is decreased in each window.

EVALUATION AND COMPARISON OF PITCH-SPELLING MODELS

The proposed pitch-spelling algorithm was tested on a set of 10 complete piano sonatas (K279–K284, K330–K333) by Mozart. This data set comprises 54,418 notes of which 14,070 notes are notated with sharps or flats (natural signs are not counted). The MIDI pitch versions of the sonatas were spelled by the algorithm and compared with the original scores;

the mismatches between the two were determined, giving a percentage of correct spelled notes over the total number of notes in the score.

Additionally, the spelling algorithm was tested on the 3 waltzes, Op. 64, No. 1–3 by Chopin. This data set comprises 4876 notes, of which 2955 notes are notated with sharps or flats. Because Chopin uses a more chromatic tonal language than Mozart, it is predicted that the algorithm will not perform as well as for pieces of the classical era.

In a first test, the spelling algorithm was applied to the two musical data sets using the pitch interval ordering in Table 1 that corresponds to the line of fifths. The distance values indicated in the first row of the table were used as penalty values (instead of the values proposed in the previous section). As for the notational parsimony principle, an enharmonically spelled note was given a penalty value of 13 (this is larger by one than the highest distance value in Table 1—more experimentation would be necessary for determining the most appropriate value). This test gives the results depicted in Table 4.

In a second test, the augmented and diminished first interval (includes the chromatic semitone, augmented and diminished eighth) that appear in position 7 of Table 1 was taken to position 12 of the table and the intervals following it were displaced by one position to the left. The aim of this test was to see how the ordering of pitch intervals may affect the spelling process (the pitch-spelling program and settings are exactly the same as in the previous test). This single change improved the results by 77 correct spellings in the Mozart data set (i.e., around 10% improvement in the number of errors) and by 87 correct spellings in the Chopin data set (i.e., around 30% improvement in the number of errors)—see Table 5.

TABLE 4

Results of First Test of Pitch-Spelling Model

	Total No. of Notes	No. of Misspelled Notes	Correct Spelling (%)
Mozart (10 sonatas)	54,418	778	98.6
Chopin (3 waltzes)	4,876	290	94.0

TABLE 5

Results of Second Test of Pitch-Spelling Model

	Total No. of Notes	No. of Misspelled Notes	Correct Spelling (%)
Mozart (10 sonatas)	54,418	701	98.7
Chopin (3 waltzes)	4,876	203	95.8

In a final test, the frequency-of-occurrence pitch interval ordering (Table 2) with the penalty values given in the previous section was used. For this ordering, the algorithm generates even better results (Table 6). The 98.8% success rate given for the Mozart data set is identical to the 98.8% success rate given by Temperley (2001, p. 136) for the Kostka-Payne corpus. The Mozart data set is more than five times larger than the Kostka-Payne data set but the Kostka-Payne data set includes a broader range of musical styles, making it a more reliable test sample. The two full algorithms, however, cannot be directly compared because of their partially different viewpoints and stages of development—for instance, the proposed algorithm has no voice-leading component that might prove very useful for transcribing the Chopin data set. Further research is required for a more substantive comparison of Temperley's complete pitch-spelling model with a more sophisticated algorithm based on the current proposal that takes into account higher level musical knowledge.

This technique of a step-by-step transcription by overlapping sections seems to be close to the processes that take place while a listener is notating little-by-little a heard melody (melodic dictation). The listener hears and notates a few bars at a time, making possible alterations to the immediately preceding notes if this is required by the new input. The proposed algorithm essentially takes into account *subsequent* pitches in order to determine the spelling of notes in the middle of the current window. This step-by-step process also enables the algorithm to move smoothly over different tonal regions as illustrated in Figure 8.

The proposed algorithm spells correctly large sections of musical works but makes a number of spelling errors as well (see examples in Figures 9–12). These errors are due to a number of factors: (a) inherent problems of the algorithm principles (e.g., there is a trade-off for the different pitch interval orderings—it is likely that there exists no single ordering that is appropriate in all cases), (b) technical problems relating to the edges of selected windows in the shifting overlapping windowing technique, (c) problems relating to the limited scope of the current implementation (i.e., voice-leading concerns are not currently addressed nor are various structural factors taken into account), and (d) problems inherent to pitch-spelling

TABLE 6
Results of Third Test of Pitch-Spelling Model

	Total No. of Notes	No. of Misspelled Notes	Correct Spelling (%)
Mozart (10 sonatas)	54,418	634	98.8
Chopin (3 waltzes)	4,876	202	95.8

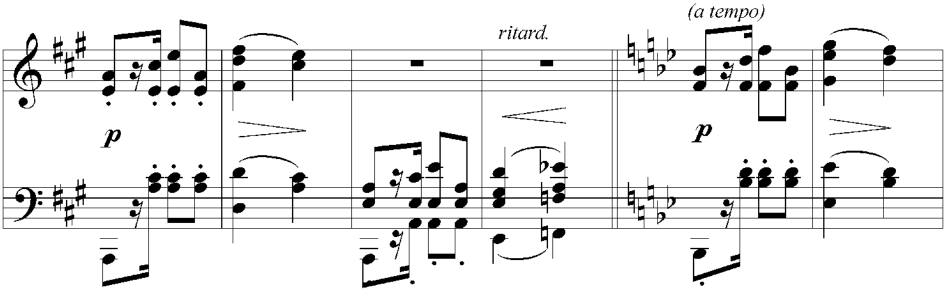


Fig. 8. This excerpt from Schumann's *Faschingschwank aus Wien, Op. 6: III, Scherzino* contains an abrupt modulation; it is spelled correctly by the algorithm. The overlapping step-by-step transcription process enables a smooth transition into the new tonal region and the successful spelling of pitches in both areas.

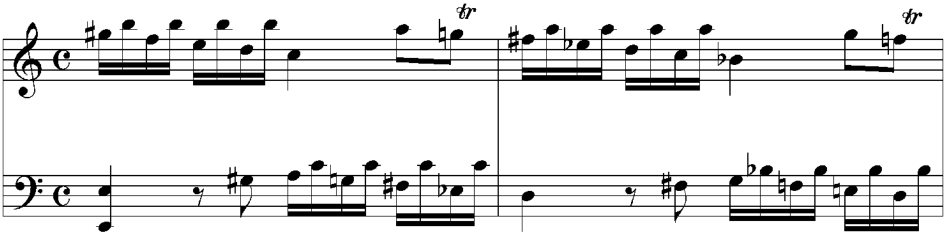


Fig. 9. Musical excerpt from Mozart's Sonata in C major K279 spelled correctly by the algorithm.

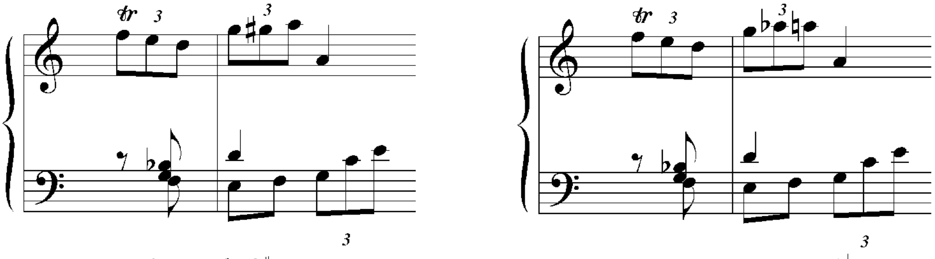


Fig. 10. Excerpt from Mozart's Sonata in C major K279. The note G♯ is misspelled as A♭ because the algorithm does not take into account voice-leading concerns (the A♭ spelling is selected because it fits better with the preceding B♭).

processes per se (e.g., uncertainty of spelling of diminished seventh chords). A large percentage of the spelling mistakes made by the algorithm are due to the type of mistake depicted in Figure 12 (especially for the Chopin data set, this is by far the main factor for the large number of mistakes in the transcription of Op. 64, No 2—the algorithm often flips over to the flat

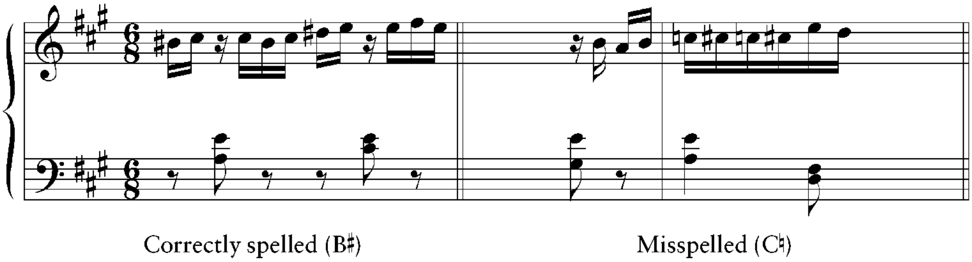


Fig. 11. Two excerpts from Mozart's Sonata in A major K331 (Var. I). The first excerpt is spelled correctly by the algorithm (notice the use of B♯). In the second excerpt, B♯ is avoided (C♮ is selected) because it creates a B-B♯ interval as well as a B♯-D interval that are strongly avoided (the algorithm has no integrated notion of voice leading).

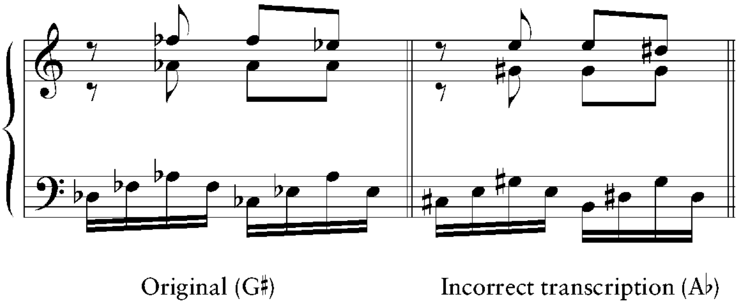


Fig. 12. Excerpt from Mozart's Sonata in B♭ major K333. The algorithm prefers the more "economic" spelling with sharps because this spelling avoids the use C♭ and F♭ (of course, in the broader context of this sonata section, the original spelling is more appropriate).

region when Chopin's spellings move "deeper" into the sharp region—additionally more idiosyncratic spellings can be found in Chopin's works, as illustrated in Figure 5).

The main findings from these tests are that, first, pitch-spelling algorithms that are based on an interval optimization process are overall very successful. The tonal information embodied in the traditional pitch interval representation is sufficient for spelling correctly the vast majority of pitches in classical tonal music. A second finding is that the ordering of traditional pitch intervals affects the pitch-spelling process; further research, however, is necessary for exploring new possibilities that may lead to even better results (pitch interval orderings may well be different for various musical styles and idioms).

FURTHER IMPROVEMENTS AND APPLICATIONS

For reasons of clarity and succinctness, the simplest possible spelling approach was used and described in this study (this approach nevertheless produces very good results and reveals its potential). There are a number of ways, however, that the algorithm can be enhanced further—some of these ideas are drawn from Temperley's pitch-spelling model.

The main improvement would be to use elementary musical knowledge to guide the spelling process. The present version of the algorithm is applied merely to the sequence of MIDI pitches. Timing information, metrical structure, note accentuation, and so on, however, can be taken into account during the spelling process. For instance, timing information can be used to calculate the distance of pitches from the center of a window and then allow pitches that are closer together to have a stronger effect on the spelling process, that is, pitches that are further away should contribute less to the overall spelling penalty—this improvement relates to the pitch recency effect taken into account in Temperley's algorithm. It seems plausible that notes such as secondary ornamental notes (e.g., passing and neighbor notes) should affect less the tonal core of a given musical section; intervals between notes that appear on metrically stronger positions or are more accented (e.g., longer duration, extreme pitch register, etc.) should contribute more to the overall spelling penalty value. Structural relationships between notes may contribute to establishing a more refined hierarchic organization of pitches and intervals that in turn can improve the spelling method.

Voice leading is also an important component of pitch spelling (Temperley's algorithm includes a special Voice-Leading Rule). In order to take voice leading into account, the various parts/voices of a musical work must be known (see, e.g., the streaming algorithm presented in Cambouropoulos, 2000). If the various melodic streams are predetermined, additional rules can cater to voice-leading effects. For instance, such a rule, for melodic sequences made up of three notes, is proposed by Cambouropoulos (1996): *Amongst equally rating spellings, prefer the spelling in which higher "quality" intervals appear last.* This rule accounts for asymmetric temporally ordered aspects of musical perception (Deutsch, 1984; Krumhansl, 1990) according to which listeners, for example, tend to hear the last note of an interval as more prominent. When there are two alternative spellings of two intervals, the system should prefer the sequence in which the last interval belongs to a "better" quality class. This rule gives precedence, for instance, to the sequence G -G♯-A over the equivalent G-A♭-A, or to the sequence A-A♭-G over the equivalent A-G♯-G.

Finally, more sophisticated search strategies can be employed for the window interval optimization methodology. Rather than the brute-force search mechanism used in the current version, various heuristics could be employed to limit the search space and render the algorithm more cognitively plausible (e.g., for a given window, the pitch-spelling process can start from the two simplest options: sharp and natural spellings, and flat and natural spellings—if the penalty value is zero for one of these, then select this spelling and stop further search for this window).

Perhaps the most obvious and practical use of a pitch-spelling algorithm is transcription of MIDI pitch into traditional note names for musical notation software applications. Most musical notation packages allow the user to set manually the key signature of a musical work. As this initial key signature determines a fixed spelling of pitches for the full length of the piece, spelling can be severely disrupted in many cases such as abrupt modulations (see Figure 8). A flexible pitch-spelling algorithm could be useful for such applications.

Conclusions

In this article, an algorithm was presented that attempts to transcribe polyphonic MIDI pitch files into the traditional pitch notation. The proposed simple pitch-spelling algorithm produced very good results for music of the classical tonal period (success rate 98.8% for the Mozart data set). The basic underlying principle of traditional pitch interval optimization encapsulates important properties of diatonic scales and tonal hierarchies and can guide successfully the pitch-spelling process. An interval ordering based on the frequency of occurrence of intervals within the major-minor scale framework performs better than other approaches. The results of the algorithm seem to support the initial claim that pitch spelling is not arbitrary and that properties of the tonal system relating to a hierarchic ordering of tonal pitch class intervals may play an important role in pitch spelling, and more generally, in tonal pitch perception. A pitch-spelling algorithm, such as the one suggested herein, can be very useful for many applications, primarily in score extraction programs, but also in other cases such as key-finding and tonality-inducing models.[1]

References

Bharucha, J. J. (1987). Music cognition and perceptual facilitation: A connectionist framework. *Music Perception, 5*:1–30.

Brown, H. (1988) The interplay of set content and temporal context in a functional theory of tonality perception. *Music Perception, 5*:219–250.

Browne, R. (1981). Tonal implication of the diatonic set. *In Theory Only, 5*(6,7):3–21.

Butler, D., & Brown, H. (1984). Tonal structure versus function: Studies of the recognition of harmonic motion. *Music Perception, 2*, 6–24.

Cambouropoulos, E. (1996). A general pitch interval representation: Theory and applications. *Journal of New Music Research, 25*, 231–251.

[1] The symbolic score data sets of the Mozart sonatas and Chopin waltzes were generously provided by the Austrian Research Institute for Artificial Intelligence, Vienna, Austria. Thanks are due to David Meredith, David Temperley, David Huron, and two anonymous reviewers for many interesting comments and suggestions on this work.

Cambouropoulos E. (1998). *Towards a general computational theory of musical structure.* Unpublished doctoral dissertation, University of Edinburgh, Scotland.

Cambouropoulos, E. (2000). From MIDI to Traditional Musical Notation. In *Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis*, Austin, Texas. Retrieved March 1, 2002 from http://www.ai.univie.ac.at/~emilios/pub.html

Deutsch, D. (1984). Two issues concerning tonal hierarchies: Comment on Castellano, Bharucha, & Krumhansl. *Journal of Experimental Psychology: General, 113*, 413–416.

Krumhansl, C. L. (1990). *Cognitive foundations of musical pitch.* New York: Oxford University Press.

Longuet-Higgins, H. C. (1976). The perception of melodies. *Nature, 263,* 646–653. Reprinted in Longuet-Higgins, H. C. (1987). *Mental processes: studies in cognitive science.* Cambridge, MA: The MIT Press.

Longuet-Higgins, H. C., & Steedman, M. J. (1971). On interpreting Bach. *Machine Intelligence, 6,* 221–241. Reprinted in Longuet-Higgins, H. C. (1987) *Mental processes: Studies in cognitive science.* Cambridge, MA: The MIT Press.

Parncutt, R. (1999). Systematic evaluation of the psychological effectiveness of non-conventional notations and key tablatures. In I. Zannos (Ed.), *Music and signs* (pp. 146–174). ASCO Art and Science, Bratislava.

Parncutt, R., & Stuckey, R. (1992). Towards a standard alternative notation and terminology based on the chromatic scale. *Musicometrica, 4,* 117–143.

Rowe, R. (2001) *Machine musicianship.* Cambridge, MA: MIT Press.

Temperley, D. (1997). An algorithm for harmonic analysis. *Music Perception, 15,* 31–68.

Temperley, D. (2001). *The cognition of basic musical structures.* Cambridge, MA: MIT Press.

Vos, P. G., & Van Geenen, E. W. (1996). A parallel-processing key-finding model. *Music Perception, 14,* 185–224.