

File ID	uvapub:76390
Filename	324424.pdf
Version	final

SOURCE (OR PART OF THE FOLLOWING SOURCE):

Type	article
Title	Compactness in the Euler-lattice: A parsimonious pitch spelling model
Author(s)	A.K. Honingh
Faculty	FNWI: Institute for Logic, Language and Computation (ILLC)
Year	2009

FULL BIBLIOGRAPHIC DETAILS:

<http://hdl.handle.net/11245/1.324424>

Copyright

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content licence (like Creative Commons).

Compactness in the Euler-lattice: A parsimonious pitch spelling model

ALINE K. HONINGH*

Music Informatics Research Group
Department of Computing
City University London

• ABSTRACT

Compactness and convexity have been shown to represent important principles in music, reflecting a notion of consonance in scales and chords, and have been successfully applied to well-known problems from music research. In this paper, the notion of compactness is applied to the problem of pitch spelling. Pitch spelling addresses the question of how to derive traditional score notation from 12-tone pitch classes or MIDI. This paper proposes a pitch spelling algorithm that is based on only one principle: compactness in the Euler-lattice. Generally, the goodness of a pitch spelling model is measured in terms of its spelling accuracy. In this paper, we concentrate on the parsimony, cognitive plausibility and generalizability of the model as well. The spelling accuracy of the algorithm was evaluated on the first book of Bach's Well-tempered Clavier and had a success rate of 99.21%. A qualitative discussion of the model's cognitive plausibility, its parsimony and its generalizability is given.

Keywords: Pitch spelling, Euler-lattice, parsimony, compactness.

INTRODUCTION

COMPACTNESS AND CONVEXITY IN THE EULER LATTICE

It is well known that subsets of the two-dimensional space \mathbb{Z}^2 can represent prominent musical and music-theoretical objects such as scales, chords and chord vocabularies. It has been noted that the major and minor diatonic scale form convex subsets in this space (Balzano, 1980). Convexity and compactness have arisen as more widespread concepts in music (Honingh & Bod, 2005; Honingh, 2006b).

The tone space known as the Euler lattice can be visualized as a set of (5-limit just intonation) frequency ratios, note names, or pitch classes represented on a two

* Escom Young Researcher Award 2006.

dimensional lattice \mathbb{Z}^2 (see figure 1) (Honingh, 2003; Honingh & Bod, 2005; Honingh, 2006b).

This lattice representation and minor variants of it appear in numerous discussions on tuning systems, for example Helmholtz (1954/1863), Riemann (1914), Fokker (1949) and Longuet-Higgins (1962a, 1962b). Longuet-Higgins and Steedman (1987/1971) have used this lattice for key finding. The geometrical notions of compactness and convexity have been applied to this tone space and have been used to classify musical pitch structures. Convexity on a two dimensional lattice has been defined as follows: A set S is convex if all elements of \mathbb{Z}^2 embedded in \mathbb{R}^2 that are in the convex hull of S in \mathbb{R}^2 are also in S . In other words: A set is convex if, drawing lines between all points in the set, all elements of the \mathbb{Z}^2 lattice that lie within the spanned area are elements of the set (Honingh & Bod, 2005). The compactness of a set of points in the Euler-lattice has been defined as the sum of the distances between all pairs of points. The lower the value of the sum, the more compact the set is. It has been shown that musical scales and chords appear as convex and compact sets in the Euler lattice (Honingh & Bod, 2005; Honingh, 2006b), and it has been suggested that convexity serves as a condition of well-formedness of scales. Both convexity and compactness of musical pitch structures have been explained in terms of consonance (Honingh & Bod, 2005; Honingh, 2006b). Convexity and compactness have been used in various computational applications from music research. Compactness can be used in a model to find the preferred intonation of a chord in isolation (Honingh, 2006a). Furthermore, convexity can be used in a modulation finding model (Honingh, 2007a). In this paper, we will show that the notion of compactness can be used in a pitch spelling model.

PITCH SPELLING

The process of pitch spelling addresses the question of which note names should be given to specific pitches. In most computer applications tones are encoded as MIDI pitch numbers which represent the different semitones. For example, middle C is represented by pitch number 60, the $C\sharp/D\flat$ immediately following middle C is represented by pitch number 61, and so on. It may be clear that MIDI pitch numbers do not distinguish between enharmonically equivalent notes. However, in tonal music, there is a lot of information in the note names about harmony, melody, scales, and intonation. Therefore, it is very useful to be able to disambiguate the music encoded as MIDI pitch numbers and transcribe it into note names. Pitch spelling is the process that deals with this problem. It is interesting that pitch spelling is a typical task that musically trained people can do very well, but until now, no computer program has been able to spell all pitches of a given input correctly. This suggests that pitch spelling is also interesting from a perceptual point of view. If a cognitively plausible model of pitch spelling is created, it could give more insight in the human pitch spelling process.

There has been an increasing interest in pitch spelling algorithms over the last

25/18	25/24	25/16	75/64	225/128	F#	C#	G#	D#	A#	6	1	8	3	10									
40/27	10/9	5/3	5/4	15/8	45/32	135/128	G	D	A	E	B	F#	C#	7	2	9	4	11	6	1			
32/27	16/9	4/3	1	3/2	9/8	27/16	81/64	Eb	Bb	F	C	G	D	A	E	3	10	5	0	7	2	9	4
256/135	64/45	16/15	8/5	6/5	9/5	27/20	81/80	Cb	Gb	Db	Ab	Eb	Bb	F	C	11	6	1	8	3	10	5	0
256/225	128/75	32/25	48/25	36/25	27/25	Ebb	Bbb	Fb	Cb	Gb	Db	2	9	4	11	6	1						

Figure 1.
Euler lattice representing frequency ratios, note names, and pitch classes.

decades, and various algorithms have been proposed (Longuet-Higgins, 1987; Temperley, 2001; Meredith, 2003, 2006, 2007a, b; Cambouropoulos, 2001, 2003; Chew & Chen, 2003a, b, 2005).

An encoding system for pitch, closely related to MIDI, is the pitch class system in which the semitones in one octave are indicated by the numbers 0 to 11. Choosing the C to be represented by pitch class 0, the encoding is given in figure 2.

0	1	2	3	4	5	6	7	8	9	10	11
Dbb		Ebb		Fb	Gbb		Abb		Bbb		Cb
	Db		Eb			Gb		Ab		Bb	
C		D		E	F		G		A		B
	C#		D#			F#		G#		A#	
B#	C##		D##	E#		F##		G##		A##	

Figure 2.

Encoding of note names into pitch classes.

This representation is similar to MIDI but uses octave equivalence, and will be used throughout this paper.

Most experts seem to agree that the pitch name of a note in a passage of tonal music is primarily a function of 1) the key at the point where the note occurs, and 2) the voice-leading structure of the music in the note's immediate context (Meredith, 2007a). To understand that the local key plays an important role, imagine a passage in C major. Using the pitch class system (fig. 2), pitch class 4 is most likely spelled as an *E* since this note is part of the scale of C major. Any other possibility to spell pitch class 4 (*D##*, *Fb*) would not be part of the scale of C major and therefore unlikely to be spelled as such. To understand that voice leading also has an influence on the pitch spelling process, consider figure 3 in which an ascending and descending chromatic scale is displayed. In case of a semitone distance between consecutive notes, in ascending direction the preceding notes are notated with sharps and in descending direction the preceding notes are notated with flats.



Figure 3.

Voice leading example: ascending and descending chromatic scale.

In this paper, we present a first version of a new pitch spelling algorithm, based on the principle of compactness in the Euler lattice.

EVALUATION CRITERIA FOR PITCH SPELLING MODELS

The correctness rate of a pitch spelling algorithm is generally indicated by the percentage of notes spelled correctly, a measure that has been named “spelling accuracy” (Meredith, 2007a). Using this spelling accuracy gives one possibility of measuring the “goodness” of a pitch spelling model. However, more possibilities of selecting the most suitable pitch spelling algorithm exist which can depend on the nature of the task. Several criteria of how to evaluate the performance of a pitch spelling algorithm have been listed and discussed by, Meredith (2007a).

In this paper, we are interested in the following evaluation criteria: spelling accuracy, cognitive plausibility, parsimony and generalizability. The reason for this choice is, first of all, that these three criteria can be applied to the first version of the pitch spelling model that we present in this paper. Furthermore, since relatively few authors have considered the evaluation criteria of parsimony and cognitive plausibility and no other authors have addressed the criterion of generalizability, it is interesting to focus on these here.

Spelling accuracy evaluates the performance of a pitch spelling model in the most basic way. A pitch spelling model is designed to spell pitches correctly (what “correctly” means here, will be discussed later), and the spelling accuracy measures how well it can do this. Therefore, if the spelling accuracy of a pitch spelling model is not above a certain percentage, the model will often be rejected without even considering other evaluation criteria. However, to be able to choose among several models that all have a high spelling accuracy, it is useful to consider other evaluation measures.

The parsimony of a pitch spelling model represents the simplicity of the model. The law of parsimony is often referred to as Occam’s Razor. Parsimony is a criterion for deciding among scientific theories or explanations, which has been applied to music as well (*e.g.* Bod, 2002b). For a discussion on model selection, see Honing (2006). Simplicity has even been regarded as a cognitive principle (Chater, 1999) and has been recognized in the Gestalt rules of perception (Wertheimer, 1923), which makes it possible to link the parsimony to the cognitive plausibility of a model.

The cognitive plausibility of a pitch spelling model represents the degree to which the model is a good representation of the human pitch spelling process. This evaluation measure is of course only important if there exists a desire to claim cognitive relevance for a model. Even without being cognitively plausible, a model can still be valuable. However, if a model can be proved to have a high degree of cognitive plausibility, this model can be studied to learn about the cognitive processes that are involved in pitch spelling.

The generalizability of a model describes how well the model, or its underlying principles, can be applied to other, different problems. A model that has been shown

to have more than one application can be valuable for its multi-functionality and may be expected to have even more applications.

PITCH SPELLING USING COMPACTNESS

COMPACTNESS

In this paper, we present a pitch spelling model that is based on two empirical observations: 1) the compactness of a set of notes in the Euler-lattice is an indication of the consonance of this set, and 2) the major and minor diatonic scales, as well as all diatonic chords can be found in compact regions in the Euler-lattice (Honingh, 2006a). Since tonal music is usually built from diatonic scales and chords, this property of compactness may be used as a tool in a pitch spelling algorithm.

We will first give an informal introduction to compactness, after which we will formalize it. The Euler-lattice can be represented in several forms as was shown in figure 1 (Honingh, 2003; Honingh & Bod, 2005). In figure 4, the tone spaces built from note names and pitch classes are shown. On the horizontal axis, the sequence of note names and pitch classes are ordered in fifths, on the vertical axis, they are ordered in major thirds. Both tone spaces can be expanded in horizontal and vertical direction, but in figure 4, only part of it is shown.

	F#	C#	G#	D#	A#			6	1	8	3	10			
G	D	A	E	B	F#	C#		7	2	9	4	11	6	1	
Eb	Bb	F	C	G	D	A	E	3	10	5	0	7	2	9	4
Cb	Gb	Db	Ab	Eb	Bb	F	C	11	6	1	8	3	10	5	0
	Ebb	Bbb	Fb	Cb	Gb	Db		2	9	4	11	6	1		

Figure 4.
Tone spaces constructed from note names and pitch classes.

Projecting the tone space of pitch classes onto the tone space of note names by mapping pitch class 0 onto the note name *C*, it becomes clear that pitch class 1 indicates, *C#* or *Db*, pitch class 2 indicates *D* or *Ebb*, etc. This projection immediately shows the problem of pitch spelling: when should pitch class 1 be translated as *C#* and when as *Db*? These kinds of problems hold for all pitch classes 0 to 11, as may be clear from figure 4. As already mentioned, the property of compactness — the degree to which elements of a set are close together — may help to solve the problem. For example, the set of pitch classes 0, 4, 7 can refer to a variety of possible sets, since pitch class 0 could refer to ..., *B#*, *C*, *Dbb*, ..., pitch class 4 could refer to ..., *D##*, *E*,

F_{\flat} , ..., and pitch class 7 could refer to ..., F_{\sharp} , G , A_{\flat} , ... From all these possible sets of note names, the set C , E , G (and transformations of the set resulting from diminished second transpositions, like D_{\flat} , F_{\flat} , A_{\flat}) represents the most compact configuration of the set. The set 0, 4, 7 is therefore most likely spelled as C , E , G , and hence the compactness of the set might be a useful tool to find the right pitches. The compactness of a set of points in the lattice will be defined here as the sum of the Euclidean distances between all pairs of points.

THE COMPACTNESS MODEL

The pitch spelling model we will now describe, is based on two very simple rules. When the music is segmented into small sets of notes,

1. Choose the spelling that is represented by the most compact set.
2. Among the sets that are equally compact, the set that is closest in key to the previous set is chosen.

As may be clear, in the pitch class tone space there is always more than one set with the same shape and therefore the same compactness. This is illustrated in figure 5. These two rules can be summarized by one principle, that of compactness, since the second rule selects the set that forms together with the previous set the most compact structure.

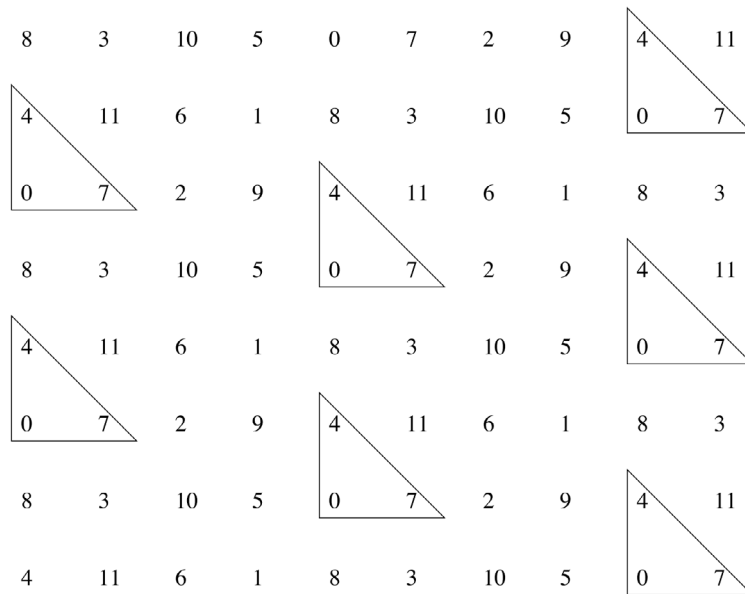


Figure 5.

Piece of the pitch class tone space which illustrates that there exists more than one set {0, 4, 7} with the same compactness.

The input for our model is a piece of music represented as pitch classes ordered according to onset time. The piece is segmented in chunks of n notes. For each chunk, the algorithm searches the most compact representation. For example, for $n = 7$, the algorithm starts with the first 7 pitch classes, the second set contains pitch classes 8 to 14, and so on. For equally compact representations of sets, rule 2 applies. For the first set of the piece, among the equally compact sets, the set that has the projection on the note name space with the least number of accidentals is chosen. For the sets thereafter, choosing between equivalent compact sets is based on closeness to an average of the previous selected sets.

In figure 7, an example is given of the pitch spelling process of the first bar from Fugue II from Bach's Well-Tempered Clavier book I. This bar is displayed in figure 6. The notes of this bar, given in pitch classes, are: 0, 11, 0, 7, 8, 0, 11, 0, 2. From the most compact sets, the one with the least number of accidentals is chosen, as can be seen from the projection in figure 7. This set, $C, B, C, G, A_b, C, B, C, D$ indeed represents the correct notes from the first bar of the fugue.



Figure 6.
First bar from Fugue II from Bach's Well-Tempered Clavier book I.

4	(11)	6	1	(8)	3	10	5	(0)	E	B	F#	C#	G#	D#	A#	E#	B#
(0)	(7)	(2)	9	4	(11)	6	1	(8)	C	G	D	A	E	(B)	F#	C#	G#
(8)	3	10	5	(0)	(7)	(2)	9	4	A _b	E _b	B _b	F	(C)	(G)	(D)	A	E
4	(11)	6	1	(8)	3	10	5	(0)	F _b	C _b	G _b	D _b	(A _b)	E _b	B _b	F	C
(0)	(7)	(2)	9	4	(11)	6	1	(8)	D _{bb}	A _{bb}	E _{bb}	B _{bb}	F _b	C _b	G _b	D _b	A _b

Figure 7.
Encoding of first bar from Fugue II from Bach's Well-Tempered Clavier.

It can happen that a complete piece is spelled (according to the algorithm) in a different key than the original. For example, a piece written in C_{\sharp} major (with 7 sharps) will be spelled by the algorithm as a piece in D_b major (with 5 flats) because the latter key contains fewer accidentals. This does not mean that the algorithm

incorrectly spelled the piece, it may be notated correctly but in a different key. Therefore, we want our pitch spelling algorithm to allow for so-called enharmonic spellings. To this end, the definition we use here (which is among others used by Meredith (2003, 2007a); Temperley (2001)) for a correctly spelled piece of music is: A piece is spelled correctly if every note name assigned by the algorithm is the same interval away from the corresponding note name in the original score. The algorithm therefore generates three spellings. One spelling is directly generated by the algorithm, one spelling is generated from the first by transposing all notes a diminished second up, and one spelling is generated from the first by transposing all notes a diminished second down, such that three enharmonic spellings result. The spelling with the smallest number of errors is then considered to be the correct spelling for the piece of music. In solving this problem of enharmonic spellings, we followed Meredith (2007a).

Each MIDI file is segmented in sets each consisting of n notes. If the number of notes the whole musical piece consists of, is not a multiple of n , the last pitches are undetermined. To overcome this problem, after the last set of n pitches, the remainder of pitches form a set (which contains less than n pitches) to be spelled using the same algorithm.

The input to our model are scores encoded in OPND (onset, pitch-name, duration) format (Meredith, 2003). Each OPND representation is a set of triples (t, n, d) given the onset Time, the pitch Name and the Duration of a single note, or sequence of tied notes in the score. The triples are ordered in a file according to onset time. For a chord, the triples are ordered from the lowest to the highest note in the chord. The note names from the OPND file are used at the end to check whether the pitches are correctly spelled. The actual input of the program are MIDI numbers that are obtained from the note names using the OPND format introduced by Meredith (2003). The compactness algorithm uses only the pitch information of the OPND format; the onset time and the duration of the notes are neglected. Furthermore, the pitch information is transcribed to MIDI numbers which is in turn transcribed to pitch classes modulo 12. The algorithm spells the notes only on the basis of these pitch classes, ordered in onset time. In case of equal onset time (for chords), the pitches are ordered from low to high frequency. As a consequence, the algorithm does not distinguish simultaneously played notes from consecutive notes.

RESULTS

The spelling accuracy of the compactness model has been tested on the preludes and fugues of Bach's Well-Tempered Clavier. The fact that more authors have used this test corpus allows us to compare our algorithm with other models. As we mentioned before, spelling accuracy is of course only one of the many possible evaluation measures. However, most of the evaluation measures are only interesting to consider if the spelling accuracy of the model is above a certain percentage, since a pitch spelling model is designed to spell pitches. Results are given in table 1 for n ranging from 1 to 7, where n is the number of notes in the set being considered.

Table 1
Results for the pitch spelling algorithm based on compactness,
as a function of the number of notes n used in the segmentation

n	percentage correctly spelled notes
1	65.76 %
2	96.57 %
3	96.42 %
4	98.80 %
5	98.58 %
6	98.98 %
7	99.21 %

For $n = 1$, the algorithm reduces to rule no. 2 described in the previous section, since the compactness of one single point always equals zero. It is therefore interesting to see that, by considering the compactness of only two notes, the result increases already by around 30%. For the best result at $n = 7$, the spelling accuracy for all separate preludes and fugues are given in table 2.

While the computational time of calculating the compactness of a set of points is quadratic in the number of notes n (*i.e.* $\frac{1}{2} n (n - 1)$, calculating the distances between all pairs of points), the number of possible configurations in the Euler lattice for which compactness has to be computed is exponential in n . When using a 9×9 lattice, pitch classes appear between 6 and 9 times (for example there are 9 locations where pitch class 0 is situated — when choosing this class in the origin). Therefore, a set of n notes has a minimum of 6^n and a maximum of 9^n configurations. For increasing n , this number becomes high, and slows down the pitch spelling process. Numbers like $n = 20$ are not uncommon for representing one or two bars in music. A first improvement to diminish the computational time of the algorithm has been made by rejecting certain configurations — the ones that are definitely not the most compact — in an early stage. In the algorithm, the compactness of an n -note set is compared with the most compact set up to that moment. If a subset of this n -note set is less compact than the most compact set, then the particular n -note set and also all other sets containing this subset are by definition less compact than the most compact set. Thus, their compactness does not need to be evaluated. This reasoning has been incorporated in the algorithm, and increased the speed of the spelling process considerably. However, still, the algorithm requires time exponential in n , therefore $n = 7$ is the practical limit here.

ERROR ANALYSIS

Studying the errors, *i.e.* the pitches that were not correctly spelled, we can observe some interesting problems with the compactness algorithm. A voice leading problem exists due to the fact that the compactness of a set is independent of the order of the

Table 2
 Results of pitch spelling algorithm for $n = 7$, for all preludes and fugues
 from the first book of Bach's Well-Tempered Clavier

no.	prelude		fugue	
	no. of notes	correctness	no. of notes	correctness
1	549	99.45 %	729	99.86 %
2	1091	99.08 %	751	99.47 %
3	810	99.26 %	1408	99.43 %
4	658	99.09 %	1311	99.39 %
5	718	98.61 %	772	100.00 %
6	784	96.94 %	715	98.46 %
7	1411	99.57 %	886	99.44 %
8	681	98.24 %	1378	98.84 %
9	421	98.57 %	732	99.86 %
10	1148	99.39 %	810	99.26 %
11	572	99.48 %	667	99.55 %
12	504	99.01 %	1309	98.24 %
13	402	99.75 %	853	99.88 %
14	604	99.01 %	807	98.88 %
15	607	99.01 %	1690	99.53 %
16	534	99.25 %	747	98.26 %
17	661	100.00 %	883	99.66 %
18	553	99.46 %	798	99.75 %
19	603	98.84 %	1172	99.74 %
20	608	97.86 %	2372	99.20 %
21	632	99.68 %	946	99.47 %
22	775	99.35 %	732	99.18 %
23	417	99.76 %	821	99.76 %
24	720	99.17 %	1792	98.88 %

notes in the set. For example, the model would always prefer the spelling $C - D_b$ over the spelling $C - C_{\sharp}$ in a 2-note set, independent of the order of those two pitches, while this is important for their spelling (see again figure 3). Having seen in section “pitch spelling” that the pitch name of a note is merely a function of 1) the local key and 2) the voice leading in the context, we now understand that the compactness method addresses only the local key.

However, the problem that causes most errors has to do with the local spelling character of the model. Errors are obtained because our algorithm does not take into account enough musical context. The compactness model adapts quickly to local key changes since the most important part of the algorithm deals with the compactness of the spelled set rather than the context. Examples of incorrectly spelled pitches are given in figure 8.

The figure represents one measure from the sixth prelude of Bach's Well-Tempered Clavier which was spelled using the compactness algorithm with $n = 4$.



Figure 8.

Measure no. 16 from prelude VI of Bach's Well-Tempered Clavier, showing three 4-note sets from which notes were misspelled.

(NB this example using $n = 4$ was chosen to give an example of some misspelled pitches, however table 2 presents the results for each piece for $n = 7$.) The circled 4-note sets in the figure indicate the three sets that have misspelled notes. The first set, $G, G, C\sharp, B\flat$ was spelled incorrectly as $G, G, D\flat, B\flat$ since the latter forms a more compact set in the tone space. In the other two circled sets all $C\sharp$'s are incorrectly spelled as $D\flat$'s as well. However, not all $C\sharp$'s are incorrectly spelled like, for example the first $C\sharp$ in the measure. Therefore, if a larger set would have been spelled by taking into account more context, these errors could probably have been avoided. Analyzing all errors from the sixth prelude (which has been chosen for analysis because of its many errors: only 96.05% was spelled correctly), it turns out that the majority of the errors are due to the misspelling of three semitones as a minor third instead of an augmented second. In the example of figure 8, the $D\flat$ was spelled as a minor third above the $B\flat$, instead of an augmented second $C\sharp$ above the $B\flat$.

EVALUATION

SPELLING ACCURACY

With our pitch spelling model we obtained a percentage of 99.21% correctly spelled pitches on the first book of Bach's Well-Tempered Clavier. Since Meredith (2003) did a comparative study on pitch spelling algorithms in which he used this corpus, our results can be exactly compared, see table 3¹. Meredith (2007a) presented new

(1) The spelling accuracy of Cambouropoulos' algorithm was originally reported by Meredith (2003) as 93.74%. However, personal communication with both Meredith and Cambouropoulos revealed that this value is not properly representative of the algorithm. Cambouropoulos' algorithm did not work very well on pieces in keys with many accidentals. In parts with modulations to keys with even more accidentals, the system would notate the pitches in a more economic way in terms of accidentals (for example, $C\sharp$ major (7 sharps) would be notated as $D\flat$ major (4 flats)). Therefore large parts from BWV 848, 853, 858, and 863 were misspelled. If these pieces were omitted from

results that are based on a large corpus and include an additional algorithm by Chew and Chen (2005). Since the focus of this article is not on improving spelling accuracy, a comparison with the results of Meredith (2003), which is based on a smaller corpus, is sufficient to put our method into context.²

Table 3
**Comparison of pitch spelling models all tested on the 41544 notes
of the first book of Bach's Well-Tempered Clavier**

Algorithm	percentage correct
Cambouropoulos	98.60 %
Longuet-Higgins	99.36 %
Temperley	99.71 %
Meredith	99.74 %
Honingh	99.21 %

It may be clear from table 3 that our compactness algorithm does not perform outstandingly compared to the other algorithms. However, the differences are small and we think that this performance is promising, given that the algorithm is based on only one simple principle, which will be the topic of the section below this one. Furthermore, it is worth noticing that some typical spelling errors produced by other algorithms are not made by the compactness algorithm. For example, Chew and Chen (2005) note that the majority of errors in their algorithm is due to the inability to detect the changes in the local tonal context, while the compactness algorithm adapts quickly to local key changes (as discussed in section "error analysis").

In the light of the above, it could be argued that the evaluation measure of spelling accuracy is not sufficient to give a good understanding of the types of errors that are made by spelling algorithms, and therefore a more specified evaluation measure is needed (see Honingh (2007b) for more on this topic).

PARSIMONY

Although the evaluation measure of parsimony or simplicity has not been addressed by many other authors, most of them have tried to formulate their pitch spelling model as parsimonious as possible in, for example, simple preference rules (Temperley, 2001). Cambouropoulos (2003) introduces the term notational parsimony to

the Bach data, the algorithm had a spelling accuracy of 98.60%. Therefore, this percentage has been included in table 3.

(2) Although table 3 shows that Meredith's algorithm has the highest note accuracy, a recent paper by Theodoru and Raphael (2007) gained an even higher note accuracy, using a probabilistic model and using voice leading information in the input.

address the process of minimizing the number of sharps and flats in a piece of music to be spelled.

As shown, the compactness pitch spelling model is based on only one simple principle (and formulated in two rules): the principle of compactness in the Euler lattice, which makes this model a parsimonious model. Furthermore, our model is parsimonious in that it uses only pitch information under octave equivalence as input. Also Cambouropoulos (2003, 2001) and Meredith (2006, the PS13s1 algorithm) use only pitch information, most other pitch spelling algorithms use some duration information as well as pitch information.

Chater (1999) and others have proposed simplicity to be a fundamental principle: "Choose the pattern that provides the simplest explanation of the available data" (Chater, 1999). If a simple model and a more complex model fit the data equally well, the simpler model is the preferred one since it contains fewer adjustable parameters. Via the Gestalt principles (Wertheimer, 1923), the principle of simplicity is linked to perception. The Gestalt principles have a history of being applied to visual perception, but have been applied to other fields as well including music (Collard *et al.*, 1981; Lerdahl & Jackendoff, 1983; Bod, 2002a, b).

COGNITIVE PLAUSIBILITY

The cognitive plausibility of a pitch spelling model refers to the degree to which the model is a good representation of the human pitch spelling process. It is assumed that musically trained people can perform a task of pitch spelling, *i.e.* notating (the pitches of) a piece of music after they have listened to it, generally well, since pitch and rhythm notation are normal tasks included in a course on basic musical skill training. It is difficult to compare the cognitive plausibility of different pitch spelling algorithms. Even in one model, many aspects are involved, and the cognitive plausibility can be different for each one of them. Therefore, we will discuss the cognitive plausibility of the presented pitch spelling system in three stages: the input, the model itself, and the output.

The input to the model is pitches under octave equivalence. The fact that no duration of the notes is involved in the input representation decreases the cognitive plausibility of the input representation. However, turning the argument around: since the spelling accuracy of our model can compete with other pitch spelling algorithms that do take into account the duration of the notes, one could wonder whether the duration of the notes is something that is actually used (and to what extent) in the cognitive process of pitch spelling.

As people listen to a piece of music, its structure unfolds as time passes. An analysis is built up gradually, and sometimes the listener revises an initial analysis of one part on the basis of what he/she hears afterwards (Temperley, 2001). The compactness model works in real-time, and indeed the pitches are spelled as time goes on and more input enters the model. A few aspects are not in agreement with this representation of the cognitive system. The input to our model is pitches divided

into chunks. To more resemble the human spelling process a sliding window should be used in which notes appear one at the time. We expect this change to have a positive effect on the spelling accuracy, however, the time-complexity will be increased a lot by this and therefore this method has not been used in this first version. To further resemble the human spelling process, a spelling algorithm should take into account present and past pitches to a certain degree to resemble a memory of what has been heard, and should be able to re-spell some pitches in the past. The compactness model takes into account the pitches in the set that is spelled, and takes into account some “past” only if more than one set are equally compact. The fact that the compactness algorithm works better for larger sets can be seen as a direct result of the past-context being important for the spelling of pitches.

The geometrical pitch representation that we use in our model is the Euler lattice. Most pitch spelling algorithms use the line of fifths as a representation of pitches (Meredith, 2007a; 2005). Many geometrical pitch representations attempt to represent perceptual features, such as consonance³ corresponding to proximity (Shepard, 1982; Krumhansl, 1990). The line of fifths is a geometrical model of pitch which reflects the consonance of intervals under octave equivalence by listing the perfect fifth as the most consonant interval. However, on the line of fifths, a major second (two fifths apart: $C - G - D$) would appear to be more consonant than a major third (four fifths apart: $C - G - D - A - E$) and minor third (three fifths apart: $E_b - B_b - F - C$), which is not in agreement with the perception of consonance of intervals in Western music. This is better represented in the Euler-lattice (fig. 1), which contains the line of fifths along one dimension. Based on this, we may say that the compactness model uses a pitch representation that is more cognitively plausible than pitch spelling models that use the line of fifths.⁴ Many pitch spelling algorithms make use of a kind of consonance of pitch intervals. In Cambouropoulos’ interval optimization model for example, the more consonant major sixth is preferred over the less consonant diminished seventh; and in Chew and Chen’s spiral array model the distances are optimized such that they correspond to Western pitch relations. Our compactness model assumes that a cluster of notes is spelled in the way that makes the cluster most “consonant”, which can for example mean, to spell all notes in the set as part of the same key. Compactness in the Euler-lattice has been shown to form a good indication of consonance (Honingh, 2006a), which is the reason that compactness has been used in the pitch spelling model presented in this paper.

(3) Here, we refer to the notion of consonance in the sense of perceived closeness of pitches. There is general agreement on the relative order of consonance of music intervals (Vidyamurthy & Chakrapani, 1992). For example, the perfect fifth is generally judged to be more consonant than the major third, which is in turn more consonant than the major second.

(4) Concerning this point, only Chew and Chen’s Spiral Array model (Chew & Chen, 2005), which is a spiral configuration of the line of fifths and includes therefore also the Euler-lattice, gives an interval distance ranking which is an even more accurate representation of Western pitch perception.

The output of a pitch spelling model is generally tested against the spelling that the composer used. In cognitive models of music it is sometimes assumed that there exists an “ideal listener” who perceives one particular analysis (Lerdahl & Jackendoff, 1983; Temperley, 2001). In the case of pitch spelling, the composer is therefore identified as this ideal listener. However, it has also been widely acknowledged that ambiguities exist, and that different people could perceive different structures, and even that one person could perceive more than one different structure (Jackendoff, 1991; Temperley, 2001). These possible interpretations of a listening would lead to different possible pitch spellings. This is important to realize when evaluating the spelling accuracy of a spelling model. It is worth (and most authors do) to look critically at the notes that the model spelled wrong, because “wrong” means here “wrong according to the composer”, and it could be that the model gave a very plausible alternative for this note. Just like other pitch spelling models, our compactness model cannot deal with ambiguities in the spelling process. However, if this aspect could be integrated in a pitch spelling model, it would certainly increase the cognitive plausibility of that model.

GENERALIZABILITY

One can wonder to what extent the compactness model is generalizable and can be applied to other, different problems. We have already mentioned that compactness on the Euler lattice has many applications in the field of music. Compact structures have been shown to describe musical pitch structures such as scales and chords (Honigh, 2006b) and compactness on the Euler lattice has been shown to be useful in an intonation model (Honigh, 2006a). One can understand that not the whole algorithm of pitch spelling is generalizable, but it is merely the underlying principle of compactness in the Euler lattice that is used. Whether the model could be applied to fields other than music as well is still to be determined.

As a special case of generalizability, portability can be discussed here. The portability of a model indicates how easy it is for this model to be ported to another test domain. Does the system have to be fully restrained, or can the same algorithm be used? Compactness on a two-dimensional lattice can be applied to other domains as well if, instead of pitches another alphabet is attached to the lattice points. In this way, the model may be generalizable to fields other than music as well.

Comparing the generalizability of our model with that of other pitch spelling models, it can be noted that not many other models are generalizable to different problems. Although Temperley’s 2001 pitch spelling model is part of a preference rule approach to music analysis which covers many problems, the pitch spelling model consists of preference rules that are specifically designed for the pitch spelling problem. The only other pitch spelling model which has a degree of generalizability is Chew and Chen’s 2005 model (which extended Chew’s 2000 and 2001 model). This pitch spelling model is based on the Spiral Array model which has been used in many musical applications, such as key finding (Chuan & Chew, 2005) and segmentation (Chew, 2002; 2006).

CONCLUSIONS

In this paper, a pitch spelling model based on the principle of compactness in the Euler lattice has been described. The model was evaluated on four different grounds: spelling accuracy, parsimony, cognitive plausibility, and generalizability. The spelling accuracy was tested on the first book of Bach's Well-Tempered Clavier and had a spelling accuracy of 99.21%. Although this result is comparable with other pitch spelling models, the fact that the discussed algorithm has a high degree of parsimony since it is based on only one principle, together with the fact that the model is generalizable to other domains makes the compactness model promising. The cognitive plausibility of the compactness model has been discussed in a qualitative way. The cognitive plausibility of the model has been discussed in three stages: the input, the model and the output. Although a high cognitive plausibility of a model does not mean that the human brain works in the same way, it does mean that there is a hypothesis of how this particular human cognitive process works. Furthermore, in the case of pitch spelling, a high spelling accuracy combined with a high cognitive plausibility makes the hypothesis even stronger.

ACKNOWLEDGMENTS

The author wishes to thank Rens Bod, Jens Wissmann and Tillman Weyde for useful discussions and comments. Thanks to David Meredith for helpful comments on a previous version of this paper, and to Elaine Chew and Emiliós Cambouropoulos for their thorough reviews and many useful suggestions.

Address for correspondence:

Aline K. Honingh
Music Informatics Research Group
Department of Computing
City University London
Northampton Square, London EC1V OHB, UK
e-mail: Aline.Honingh.1@soi.city.ac.uk

• REFERENCES

- Balzano, G. J. (1980). The group theoretical description of 12-fold and microtonal pitch systems. *Computer Music Journal*, 4 (4), 66-84.
- Bod, R. (2002a). Memory-based models of melodic analysis: Challenging the Gestalt principles. *Journal of New Music Research*, 31 (1), 27-37.
- Bod, R. (2002b). A unified model of structural organization in language and music. *Journal of Artificial Intelligence Research*, 17, 289-308.
- Cambouropoulos, E. (2001). Automatic pitch spelling: From numbers to sharps and flats. In *Proceedings of the VIII Brazilian Symposium on Computer Music*. Fortaleza, Brasil.
- Cambouropoulos, E. (2003). Pitch spelling, a computational model. *Music Perception*, 20 (4), 411-29.
- Chater, N. (1999). The search for simplicity: A fundamental cognitive principle? *The Quarterly Journal of Experimental Psychology*, 52 (A) (2), 273-302.
- Chew, E. (2000). *Towards a mathematical model of tonality*. Ph.D. thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge.
- Chew, E. (2001). Modeling tonality: Applications to music cognition. In J. D. Moore, & K. Stenning (eds) *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society, CogSci2001*, (pp. 206-11). Edinburgh, Scotland, UK.
- Chew, E. (2002). The spiral array: An algorithm for determining key boundaries. In C. Anagnostopoulou, M. Ferrand, & A. Smaill (eds) *Music and Artificial Intelligence — Second International Conference, ICMAI 2002*, (pp. 18-31). Edinburgh, Scotland, UK.
- Chew, E. (2006). Slicing it all ways: Mathematical models for tonal induction, approximation and segmentation using the spiral array. *INFORMS Journal on Computing*, 18 (3).
- Chew, E., & Chen, Y.-C. (2003a). Determining context-defining windows: Pitch spelling using the spiral array. In *Proceedings of the 4th International Conference for Music Information Retrieval, ISMIR*.
- Chew, E., & Chen, Y.-C. (2003b). Mapping midi to the spiral array: Disambiguating pitch spellings. In H. K. Bhargava, & N. Ye (eds) *Computational Modeling and Problem Solving in the Networked World, Proceedings of the 8th INFORMS Computer Society Conference, ICS2003*, vol. 21 of *OR/CS Interfaces Series*, (pp. 259-75). Kluwer Academic Publishers.
- Chew, E., & Chen, Y.-C. (2005). Pitch spelling using the spiral array. *Computer Music Journal*, 29 (2), 61-76.
- Chuan, C.-H., & Chew, E. (2005). Polyphonic audio key-finding using the spiral array CEG algorithm. In *International Conference on Multimedia and Expo (ICME)*, (pp. 21-24). Amsterdam, Netherlands.
- Collard, R., Vos, P., & Leeuwenberg, E. (1981). What melody tells about metre in music. *Zeitschrift für Psychologie*, 189, 25-33.
- Fokker, A. D. (1949). *Just Intonation*. The Hague: Martinus Nijhoff.
- Helmholtz, H. (1954/1863). *On the Sensations of Tone*, (second English ed). Dover.
- Honing, H. (2006). Computational modeling of music cognition: a case study on model selection. *Music Perception*, 23 (5), 365-76.
- Honingh, A. K. (2003). Group theoretic description of just intonation. In G. Buzzanca *et al.* (eds)

- Proceedings of the 3rd International Conference Understanding and Creating Music*, vol. 3. Caserta, Italy.
- Honingh, A. K. (2006a). Convexity and compactness as models for the preferred intonation of chords. In *Proceedings of the Ninth International Conference on Music Perception and Cognition (ICMPC 9)*. Bologna, 22-26 August.
- Honingh, A. K. (2006b). *The Origin and Well-Formedness of Tonal Pitch Structures*. Ph.D. thesis, University of Amsterdam, The Netherlands.
- Honingh, A. K. (2007a). Automatic modulation finding using convex sets of notes. In *Proceedings of Mathematics and Computation in Music (MCM2007)*. Berlin, Germany May 18-20.
- Honingh, A. K. (2007b). Pitch spelling: Investigating reduction of the search space. In *Proceedings of the 4th Sound and Music Computing Conference (SMC07)*. Lefkada, Greece, 11-13 July.
- Honingh, A. K., & Bod, R. (2005). Convexity and the well-formedness of musical objects. *Journal of New Music Research*, 34 (3), 293-303.
- Jackendoff, R. (1991). Musical parsing and musical affect. *Music Perception*, 9, 199-230.
- Krumhansl, C. L. (1990). *Cognitive Foundations of Musical Pitch*. Oxford Psychology Series No. 17. Oxford University Press.
- Lerdahl, F., & Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press.
- Longuet-Higgins, H. C. (1962a). Letter to a musical friend. *Music Review*, 23, 244-48.
- Longuet-Higgins, H. C. (1962b). Second letter to a musical friend. *Music Review*, 23, 271-80.
- Longuet-Higgins, H. C. (1987). The perception of melodies. In H. C. Longuet-Higgins (ed) *Mental Processes: Studies in Cognitive Science*, (pp. 105-129). London: British Psychological Society/MIT Press. Published earlier as Longuet-Higgins 1976.
- Longuet-Higgins, H. C., & Steedman, M. (1987/1971). On interpreting Bach. In H. C. Longuet-Higgins (ed) *Mental Processes: Studies in Cognitive Science*, (pp. 82-104). British Psychological Society/MIT Press. Published earlier as Longuet-Higgins and Steedman (1971).
- Meredith, D. (2003). Pitch spelling algorithms. In *Proceedings of the Fifth Triennial ESCOM Conference*, (pp. 204-07). Hanover University of Music and Drama, Germany.
- Meredith, D. (2005). Comparing pitch spelling algorithms on a large corpus of tonal music. In U. K. Wiil (ed) *Computer Music Modeling and Retrieval: Second International Symposium, CMMR*, (pp. 173-92). Springer, Berlin.
- Meredith, D. (2006). The ps13 pitch spelling algorithm. *Journal of New Music Research*, 35 (2), 121-59.
- Meredith, D. (2007a). *Computing Pitch Names in Tonal Music: A Comparative Analysis of Pitch Spelling Algorithms*. D. Phil. dissertation. Faculty of Music, University of Oxford.
- Meredith, D. (2007b). Optimizing Chew and Chen's pitch spelling algorithm. *Computer Music Journal*, 31 (2).
- Riemann, H. (1914). Ideen zu einer Lehre von den Tonvorstellungen. *Jahrbuch der Musikbibliothek Peters*, 21/22, 1-26. Leipzig.
- Shepard, R. N. (1982). Geometrical approximations to the structure of musical pitch. *Psychological Review*, 89, 305-33.
- Temperley, D. (2001). *The Cognition of Basic Musical Structures*. The MIT Press.
- Theodoru, G., & Raphael, C. (2007). Pitch spelling with conditionally independent voices. In *the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, (pp. 201-06). Vienna, Austria.

- Vidyamurthy, G., & Chakrapani, J. (1992). Cognition of tonal centers: A fuzzy approach. *Computer Music Journal*, 16 (2), 45-50.
- Wertheimer, M. (1923). Untersuchungen zur Lehre von der Gestalt. *Psychologische Forschung*, 4, 301-50.

- **La compactividad en el entramado de Euler:
un modelo parsimonioso de deletreo de alturas**

La compactividad y convexidad han revelado representar importantes principios en música, reflejando una noción de consonancia en escalas y acordes, y han sido aplicadas con éxito a problemas bien conocidos de investigación musical. En este trabajo, se aplica la noción de compactividad al problema del deletreo de alturas. El deletreo de alturas dirige la cuestión de cómo derivar la notación tradición de las partituras de doce clases de alturas o MIDI. Este trabajo propone un algoritmo de deletreo de altura que está basado en un único principio: la compactividad en el entramado de Euler. Generalmente, la bondad de un modelo de deletreo de altura se mide en términos de su precisión en el deletreo. En este trabajo, nos concentramos también en la parsimonia, plausibilidad cognitiva y generalización del modelo. La precisión de deletreo del algoritmo fue evaluada en el primer libro del *Clave Bien Temperado* de Bach, y tuvo una ratio de éxito del 99,21%. Se ofrece una discusión cualitativa de la plausibilidad cognitiva del modelo, su parsimonia y su generalización.

- **Compattezza nel metodo di Eulero: un modello semplice
di rappresentazione grafica dell'altezza**

E' stato dimostrato che la compattezza e la convessità sono importanti principi nella musica che riflettono la nozione di consonanza nelle scale e negli accordi oltre ad essere stati applicati con successo ai problemi noti da parte della ricerca musicologica. In questo lavoro il concetto di compattezza è applicato al problema della scrittura dell'altezza. La scrittura dell'altezza pone la questione di come ottenere la notazione tradizionale della partitura dalle classi di altezza di dodici note o MIDI. Questo saggio propone un algoritmo della scrittura dell'altezza basato su un unico principio: la compattezza nel metodo di Eulero. Generalmente, la bontà di un modello di scrittura dell'altezza è misurata in termini di accuratezza della scrittura stessa. In questo saggio ci concentriamo anche sulla semplicità, l'attendibilità cognitiva e la "generalizzabilità" di questo modello. L'accuratezza della scrittura dell'algoritmo è stata calcolata sul primo libro del *Clavicembalo ben temperato* di Bach con un indice di successo pari al 99.21%. Argomentiamo dal punto di vista qualitativo l'attendibilità, la semplicità e la "generalizzabilità" del modello.

- **Compacité dans le maillage d'Euler : un modèle parcimonieux
de transcription de hauteur tonale**

On a montré que la compacité et la convexité représentent deux principes importants de la musique, qui reflètent une notion de consonance dans les gammes et les accords, et ont été appliquées avec succès à des problèmes bien connus de la recherche musicale. Dans cet article, la notion de compacité est appliquée au problème de la transcription de tonalité. La transcription de tonalité pose la question

de comment dériver une notation musicale traditionnelle de classes de hauteurs tonales de 12 tons ou MIDI. Nous proposons ici un algorithme de transcription de tonalité basé sur un principe unique : la compacité dans le maillage d'Euler. En général, un bon modèle de transcription de tonalité est évalué selon sa justesse de transcription. Dans cet article, nous nous centrons également sur le caractère parcimonieux, la plausibilité cognitive et la possibilité de généralisation du modèle. La justesse de transcription de l'algorithme a été évaluée sur le premier livre du *Clavier bien tempéré* de Bach avec un taux de succès de 99,21 %. On termine par une discussion qualitative sur la plausibilité cognitive, le caractère parcimonieux et la possibilité de généralisation du modèle.

- **Kompaktheit im Eulergitter: Ein sparsames Modell der Tonbeschreibung**

Es wurde gezeigt, dass Kompaktheit und Konvexität wichtige Prinzipien in der Musik darstellen. Sie spiegeln eine Idee von Konsonanz in Skalen und Akkorden wider und wurden erfolgreich bei wohlbekannten Problemen der Musikforschung angewendet. In diesem Aufsatz wird die Idee der Kompaktheit auf das Problem der Tonhöhenbeschreibung angewendet. Die Beschreibung betrifft die Frage, wie sich die traditionelle Notierung von Zwölftonklassen oder MIDI ableiten lässt. Dieser Aufsatz schlägt einen Algorithmus zur Tonhöhenschreibweise vor, der nur auf einem Prinzip beruht: der Kompaktheit im Eulerschen Gitter. Insgesamt wird die Güte eines Modells der Tonhöhenschreibweise hinsichtlich seiner Genauigkeit gemessen. In diesem Aufsatz konzentrieren wir uns ebenso auf die Sparsamkeit, kognitive Plausibilität und Generalisierbarkeit des Modells. Die Genauigkeit des Algorithmus wurde mit dem ersten Band von Bachs Wohltemperierten Klavier evaluiert. Die Erfolgsrate lag bei 99,21 %. Eine qualitative Diskussion der kognitiven Plausibilität des Modells sowie seiner Sparsamkeit und seiner Generalisierbarkeit wird geboten.