Pitch spelling: Investigating reductions of the search space

Aline Honingh*

*Music Informatics research group, Department of Computing, City University, London, UK

***Abstract*— Pitch spelling addresses the question of how to derive traditional score notation from pitch classes or MIDI numbers. In this paper, we motivate that the diatonic notes in a piece of music are easier to spell correctly than the non-diatonic notes. Then we investigate 1) whether the generally used method of calculating the proportion of correctly spelled notes to evaluate pitch spelling models can be replaced by a method that concentrates only on the non-diatonic pitches, and 2) if an extra evaluation measure to distinguish the incorrectly spelled diatonic notes from the incorrectly spelled non-diatonic notes would be useful. To this end, we calculate the typical percentage of pitch classes that correspond to diatonic notes and check whether those pitch classes do indeed refer to diatonic notes in a piece of music. We explore extensions of the diatonic set. Finally, a good performing pitch spelling algorithm is investigated to see what percentage of its incorrectly spelled notes are diatonic notes. It turns out that a substantial part of the incorrectly spelled notes consist of diatonic notes, which means that the standard evaluation measure of pitch spelling algorithms cannot be replaced by a measure that only concentrates on non-diatonic notes without losing important information. We propose instead that two evaluation measures could be added to the standard correctness rate to be able to give a more complete view of a pitch spelling model.**

I. Introduction

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 semi tones. 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. There has been an increasing interest in pitch spelling algorithms over the last decades, and various algorithms have been proposed [11], [18], [13], [14], [15], [16], [3], [5], [6], [8].

These proposed pitch spelling algorithms have been shown to give a high correctness rate, somewhere between 97% and 100% correctly spelled notes. One could think that all these pitch spelling algorithms work extremely well or wonder whether it is just very easy to get a good result in this field. If a part of the spelling process is indeed easy, it might be worth to focus on the more difficult part instead.

The correctness rate of a pitch spelling algorithm is generally indicated by the percentage of notes spelled correctly, a measure that has been named *note accuracy* [14]. It has however been proposed by Cambouropoulos [2] to measure the percentage of correctly spelled notes among the notes with accidentals instead, which implies that the notes without accidentals are not difficult to spell correctly. Meredith [14] writes in answer to this: "However, there is no guarantee that every mis-spelled note is a note that has an accidental in the original score".

Still, one could intuitively think that some notes are easier to spell correctly than others. In the field of pitch spelling it would then be interesting to concentrate on the most difficult notes. Let us consider a piece in C major. A pitch indicated by pitch class[1] 0 can indicate a C, $D\flat\flat$, or another enharmonic equivalent note, however in C major it would most likely indicate a C. For the same reason, pitch class 2 would most likely indicate a D (as opposed to e.g. a $C\sharp\sharp$ or $E\flat\flat$) in C major. We can thus understand that the notes comprising the scale of the key in which the music is written, are usually easy to spell. For a piece in C major, this approach is the same as that of Cambouropoulos [2] referred to above, since in this case all non-diatonic notes are the notes with accidentals.

The fact that some notes may be easy to spell does not yet mean that all these diatonic notes are necessarily spelled correctly, and thus we should find out how many diatonic notes are indeed spelled correctly. This will help us to investigate the question whether the generally used method of comparing pitch spelling models by their note accuracy can be replaced by measuring the percentage of notes spelled correctly among the non-diatonic notes, to so give a clearer picture of the problem and its search space.

This paper focuses on the following issues within the field of pitch spelling. First of all, in section II, we calculate the percentage of diatonic pitches in a piece of music. Knowing this, we could say something about the size of the area to focus on in pitch spelling. If the percentage of diatonic notes is high, this could explain the success of many pitch spelling algorithms. We define the diatonic pitch classes in a piece of music as the pitch classes or MIDI numbers that correspond to the

[1] Pitch classes represent the same information as MIDI pitch numbers, only under octave equivalence.

diatonic set of the (given) key of the piece. For example, to calculate the percentage of diatonic pitch classes in a piece of music in C major, we count the instances of the pitch classes $0, 2, 4, 5, 7, 9, 11$ corresponding to the notes C, D, E, F, G, A, B. However, the pitch class 0 corresponds also to the note $D\flat\flat$ or $B\sharp$, and so on. Therefore, we want to find out whether all these counted diatonic pitch classes really correspond to the diatonic notes of the key of the piece. If they do, the percentage of diatonic pitch classes corresponds to the percentage of diatonic notes in the piece of music. Furthermore, then a simple method of spelling diatonic notes on the basis of the key has been found. This is our next focus addressed in section II.

The definition of the diatonic set has been extended by some authors to include more pitches than just the ones from the major and minor diatonic scale. In section III we discuss what happens if we use this new definition and ask ourselves the questions that we have before.

In the rest of the paper we look at the result of a pitch spelling algorithm in the light of the above findings. Does a specially developed pitch spelling algorithm perform any better than the naive program that spells any pitches according to the diatonic context it has been given? To be able to have a close look on the notes in the music that are not spelled correctly, we concentrate on one particular pitch spelling model: the compactness model which has been proposed earlier by the author [8]. A comprehensive overview of this model is given in section IV. In section V it is investigated which percentage of the diatonic notes has been spelled correctly by this pitch spelling algorithm. Since our hypothesis was that the diatonic notes are the easiest to spell, we expect high percentages of correctly spelled notes here. It is musically interesting to look at the incorrectly spelled diatonic notes.

II. DIATONIC NOTES AND TRIVIAL SPELLINGS

Which percentage of the notes of a piece of music are diatonic notes? In a first attempt to answer this question, a program has been written that counts the number of pitch numbers that correspond to the diatonic notes in a piece of music. In fact, this program does not exactly answer the question above, but answers the question of which percentage of the pitch classes in a piece of music correspond to diatonic pitches. For a piece in C major, all the instances of the pitch numbers $0, 2, 4, 5, 7, 9, 11$ are counted since they correspond to the notes C, D, E, F, G, A, B, C. For a piece in a minor key, we decided to take into account the notes from the natural minor scale, as well as the notes from the melodic and harmonic minor scale. Thus, for C minor, the instances of the pitch numbers $0, 2, 3, 5, 7, 8, 9, 10, 11$ are counted since they correspond to the notes $C, D, E\flat, F, G, A\flat, A, B\flat, B$. The algorithm was applied to the preludes and fugues of the first book of the Well-tempered Clavier by Bach. The percentages of diatonic notes in each prelude and fugue are given in table I. We see that the percentages lie between 85% and 98% with an average of 92.69%, so we can conclude that a fairly large percentage of notes from these pieces consists of diatonic notes.

In the program, we did not take into account the modulations that might be involved in the music. So, if a piece of music is in C major, and it modulates to G major and then back to C major, the notes G, A, B, C, D, E from G major will be counted as diatonic notes according to the program, since these notes are also present in the key of C. Only the note $F\sharp$ (present in G major) is not counted as a diatonic note since it does not appear in the scale of C major. We can thus understand that we miss some diatonic notes in other keys because of some modulations in the piece, but this number turns out not to be so big as we see from table I.

We can now wonder if the search space for pitch spelling algorithms is really limited to the 7.31% of the notes in these pieces, since this is the average percentage of non-diatonic notes. Before answering this question, which we attempt to do in section V, it is first important to investigate if the pitches counted as diatonic pitches, do indeed refer to diatonic notes in the music. Therefore we calculate the percentage of diatonic notes that are spelled correctly using the assumption that in a certain key, every pitch class that is part of the diatonic set would be translated to the corresponding note in the diatonic set. That means that in C major, every pitch number 0 would be translated to a C, every 2 to a D, and so on. We will refer to this spelling as the *trivial spelling*. For every prelude and fugue the percentage of correct trivial spelling is given in table I. Be aware that these percentage of correct trivial spelling only address the spelling of the diatonic notes, not of all notes in the piece of music. Since we expected that this trivial spelling is usually the correct spelling, it is surprising to find that four preludes and fugues are not spelled 100% correctly. For example, in fugue no. 10 (bwv 855b), one of the diatonic notes was incorrectly spelled. The piece is in E minor, so the diatonic notes encompass $E, F\sharp, G, A, B, C, C\sharp, D, D\sharp$. The note in the prelude that was spelled incorrectly was pitch number 3 which should have been translated to an $E\flat$, but was spelled as $D\sharp$ by the program, since this note is within the set of diatonic notes. At the bar where the spelling error occurred, the music has modulated to G major (see [1]). The specific note (the $E\flat$) is part of a downward chromatic melody (the $E\flat$ leading to a D) which is the reason that the note is spelled as an $E\flat$ in the music. The other errors include multiple $B\flat$'s spelled as $A\sharp$ and a $F\sharp\sharp$ spelled as G in B minor, and a $D\sharp\sharp$ spelled as E in $G\sharp$ minor, which could all be explained in a similar way as above from the music.

Despite the few errors in the (trivial) spelling of the diatonic notes, the percentage of diatonic notes in a piece of music is very high (92.69% on average) and the percentage of correct trivial spelling does not significantly differ from 100%. This suggests that the most interesting part of the pitches to be spelled by a pitch spelling algorithm is the approximately 7.3% of the notes representing the non-diatonic notes. However, be aware that the process of spelling the diatonic notes is not a trivial process although we have used the term 'trivial spelling'. In spelling the diatonic notes above we used the key information to know which note name to assign to a pitch class, but key information is not present in the input file for a pitch spelling algorithm. However, a lot of research has been done in the area of key finding [4], [10], [20], [12], [17] and for example the research by Chew [4] shows us that the correct keys of all fugue subjects of book I of the Well-tempered Clavier can be found with the proposed algorithm. Moreover, she shows that the program on average only needs 3.75 pitch events to correctly determine the key. From this we may conclude that determining the key of a piece of music can be fairly well done, and combining a key-finding algorithm with the trivial spelling procedure that was illustrated above, provides us with an intuitive pitch spelling algorithm for the diatonic pitches.

III. Extending the diatonic set

Even if we leave modulations out for a moment and consider a piece of music in one and the same key, one could argue that the notes from a piece of music in one key do not just come from one scale, as we saw in the example above where an instance of a $E\flat$ was found in the key of G major. Some authors have tried to formalize the idea that the key contains more notes than just the scale of the tonic [19], [11]. Longuet-Higgins [11] states that "a note is regarded as belonging to a given key if its sharpness relative to the tonic lies in the range -5 to $+6$ inclusive". The sharpness of a tone refers to the digit that is attached to the note in fifth ordering, starting with $C = 0$. Thus according to Longuet-Higgins [11] the key of C should include the notes $D\flat, A\flat, E\flat, B\flat, F, C, G, D, A, E, B, F\sharp$. Using this 'new' key-content, we could do the same as we did before. We could calculate the percentage of diatonic pitches in each piece of music using this new set of diatonic notes. However, it may be clear that it is not necessary to calculate these percentages, because if we count the instances of the pitch numbers $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$ corresponding to the notes $D\flat, A\flat, E\flat, B\flat, F, C, G, D, A, E, B, F\sharp$, we will end up counting all notes of the piece. The percentages of diatonic notes would all read 100%, due to the fact that a note is defined for all 12 semitones in the octave. It is however interesting to see how many pitches would be spelled correctly if we would translate the pitch numbers $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$ into the notes $C, D\flat, D, E\flat, E, F, F\sharp, G, A\flat, A, B\flat, B$ respectively, which would be the 'trivial' spelling according this new key context proposed by Longuet-Higgins. These percentages are given in the 5th column of table I under the heading LH. It can be seen from table I that the percentages of correctness of this new trivial spelling are fairly high (95.28% on average), and therefore this method seems to be a good intuitive approach to spell the pitches of a piece of music, given a certain key. Since this intuitive method seems to work already quite well,

	F#	C#	G#	D#	A#		
G	D	A	E	B	F#	C#	
Eb	Bb	F	C	G	D	A	E
Cb	Gb	Db	Ab	Eb	Bb	F	C
	Ebb	Bbb	Fb	Cb	Gb	Db	

	6	1	8	3	10		
7	2	9	4	11	6	1	
3	10	5	0	7	2	9	4
11	6	1	8	3	10	5	0
	2	9	4	11	6	1	

Fig. 1. Tone space or Euler-lattice constructed from note names and pitch numbers.

we are interested to compare these percentages with the results of a real pitch speling model.

IV. The compactness pitch spelling algorithm

We will now give a comprehensive overview of the compactness pitch spelling model proposed by Honingh [8], since this pitch spelling algorithm will be used further in this paper, to compare its note accuracy with the results of the trivial spelling methods described above. The space of pitches that is used in this method is known under the name of Euler-lattice and can be represented in several forms [7], [9]. In fig. 1 the Euler-lattices 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 fig. 1, only part of the space is shown.

Projecting the tone space of pitch classes onto the tone space of note names by mapping the number 0 onto the note name C, it becomes clear that pitch class 1 indicates $C\sharp$ or $D\flat$, pitch class 2 indicates D or $E\flat\flat$, etc. This projection immediately shows the problem of pitch spelling: when should pitch class 1 be translated as $C\sharp$ and when as $D\flat$? These kind of problems hold for all pitch classes 0 to 11, as may be clear from fig. 1.

The pitch spelling problem is now attacked using the property of compactness[2]. The compactness of a set of points in the lattice is defined here as the sum of the Euclidean distances between all pairs of points. That means that, the lower this value for compactness, the more

[2]Some other pitch spelling algorithms make also use of a related concept to compactness [6], [18].

TABLE I

THE PRELUDES AND FUGUES FROM BOOK I OF THE WELL TEMPERED CLAVIER BY BACH, EXPLORED USING VARIOUS MEASURES DESCRIBED IN THE TEXT.

WTC no.	key	% of diatonic notes	% of trivial correct spelling	LH	% of correct spelling using compactness algorithm ($n = 6$)	% correct spelling of diatonic notes with compactness algorithm ($n = 6$)
1a	C ma	93.08	100	99.27	99.45	100
1b	C ma	91.77	100	97.26	99.45	100
2a	C mi	94.23	100	100	99.63	99.71
2b	C mi	95.74	100	100	99.20	99.17
3a	$C\sharp$ ma	91.23	100	97.90	99.13	99.73
3b	$C\sharp$ ma	91.34	100	97.16	99.43	100
4a	$C\sharp$ mi	95.90	100	100	97.72	98.42
4b	$C\sharp$ mi	94.43	100	99.77	99.69	99.92
5a	D ma	85.93	100	97.35	98.61	99.68
5b	D ma	96.11	100	99.48	100	100
6a	D mi	92.35	100	99.87	97.32	98.62
6b	D mi	91.61	100	99.86	98.46	98.63
7a	$E\flat$ ma	91.21	100	97.73	99.79	100
7b	$E\flat$ ma	91.54	100	95.82	98.98	99.88
8a	$E\flat$ mi	92.36	100	99.71	98.24	99.05
8b	$D\sharp$ mi	94.85	100	100	98.98	99.54
9a	E ma	86.94	100	97.39	99.05	100
9b	E ma	94.54	100	98.36	99.86	99.45
10a	E mi	94.60	100	100	99.13	99.59
10b	E mi	89.63	99.86	98.27	99.01	99.80
11a	F ma	88.29	100	93.18	99.48	100
11b	F ma	93.55	100	96.55	99.40	99.15
12a	F mi	93.25	100	99.80	98.41	99.00
12b	F mi	92.13	100	99.62	98.32	99.45
13a	$F\sharp$ ma	91.04	100	96.02	99.50	100
13b	$F\sharp$ ma	94.14	100	98.36	99.77	100
14a	$F\sharp$ mi	97.68	100	100	99.01	99.66
14b	$F\sharp$ mi	92.94	100	99.73	98.76	98.67
15a	G ma	86.33	100	94.73	91.43	90.84
15b	G ma	90.18	100	96.98	99.35	99.93
16a	G mi	93.63	100	100	98.88	99.20
16b	G mi	95.18	100	100	97.99	98.87
17a	$A\flat$ ma	95.16	100	100	100	100
17b	$A\flat$ ma	92.98	100	97.28	99.55	99.76
18a	$G\sharp$ mi	94.58	100	99.82	99.10	99.62
18b	$G\sharp$ mi	92.23	99.86	99.00	99.75	99.86
19a	A ma	92.37	100	98.18	99.34	99.82
19b	A ma	92.15	100	97.44	99.23	99.81
20a	A mi	92.76	100	99.67	96.88	98.23
20b	A mi	93.51	100	99.83	98.95	99.19
21a	$B\flat$ ma	91.61	100	98.26	100	100
21b	$B\flat$ ma	92.28	100	96.41	99.58	99.89
22a	$B\flat$ mi	96.52	100	100	99.10	99.20
22b	$B\flat$ mi	93.99	100	99.04	99.32	99.56
23a	B ma	91.85	100	97.60	99.76	100
23b	B ma	93.42	100	98.78	99.88	100
24a	B mi	95.69	99.71	99.72	99.03	99.27
24b	B mi	90.29	99.81	98.49	98.10	99.32
average		92.69	100	95.28	98.98	99.43

compact the set is. The model we will 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

In the pitch class tone space there is always more than one set with the same shape and therefore the same compactness. These sets correspond to sets in the note name space that have been transposed a diminished second up or down from each other.

The two pitch spelling 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. 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

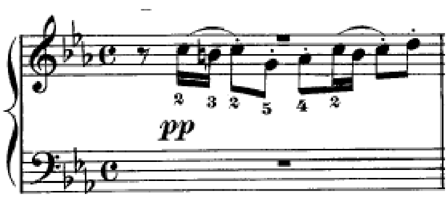

Fig. 2. First bar from Fugue II from Bach's Well-tempered Clavier book I.

4	11	6	1	8	3	10	5	0
0	7	2	9	4	11	6	1	8
8	3	10	5	0	7	2	9	4
4	11	6	1	8	3	10	5	0
0	7	2	9	4	11	6	1	8

E	B	F#	C#	G#	D#	A#	E#	B#
C	G	D	A	E	B	F#	C#	G#
Ab	Eb	Bb	F	C	G	D	A	E
Fb	Cb	Gb	Db	Ab	Eb	Bb	F	C
Dbb	Abb	Ebb	Bbb	Fb	Cb	Gb	Db	Ab

Fig. 3. Encoding of first bar from Fugue II from Bach's Well-tempered Clavier.

of accidentals is chosen. For the sets thereafter, we would want to choose the set which is closest in key (number of accidentals) to the preceding set. However, if the music is segmented in very small sets, and there is a sudden change of key, this may not work properly. Therefore, the average is calculated between the number of accidentals in the previous set and the sets before that set.

In fig. 3 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 fig. 2. 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 fig. 3. This set, $C, B, C, G, A\flat, C, B, C, D$ indeed represents the correct notes from the first bar of the fugue.

To spell a whole piece of music, each MIDI file is segmented in sets each consisting of n notes, and each set is spelled according to the process described above. 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.

To illustrate the performance of the compactness algorithm, we show here the performance on the preludes and fugues of Bach's Well-tempered Clavier. Results are given in table II for n ranging from 1 to 7, where n is the number of notes in the set being considered. For $n = 1$, the algorithm reduces to rule no. 2, 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 with around 30%. Since the algorithm requires time exponential in n, $n = 7$ is the practical limit here[3]. From table II it can be seen that the best performance occurs at $n = 7$. The worst performance is (if $n = 1$ is not taken into account) for $n = 3$.

TABLE II

RESULTS FOR THE COMPACTNESS PITCH SPELLING ALGORITHM, 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** %

V. PITCH SPELLING PERFORMANCE ON DIATONIC NOTES

We compare the note accuracy of the compactness algorithm with the percentage of diatonic notes in a piece, in order to see if the pitch spelling algorithm spelled more notes correctly than only the diatonic notes. The percentage of correct spelling according to the compactness algorithm with $n = 6$ can be found in table I. Comparing these percentages with the percentages of diatonic notes, we can see that the compactness algorithm spells indeed more notes correctly (98.98% on average) than just the diatonic notes (92.69% on average). Furthermore, the pitch spelling algorithm does also perform better than the intuitive spelling method using the proposed extension of the diatonic scale.

It is not yet clear if all diatonic notes are spelled correctly according to this algorithm. Therefore, in column 7 of table I we have calculated the percentages of correctly spelled diatonic notes using the compactness algorithm with $n = 6$. Since these percentages clearly do not all read 100%, the pitch spelling algorithm did not spell all diatoninc notes correctly. This is a surprizing result, since our hypothesis was that the diatoninc notes are the ones that are most easy to spell. This pitch spelling algorithm performs worse on diatonic notes than the naive 'trivial spelling' approach combined with a key finding program does, as described at the end of section II. In table III the

[3]Some approaches of dynamical programming have already been incorporated in the algorithm. Current investigations try to optimize the algorithm further.

average values of these percentages calculated with other values for n can be found as well. For higher values of n, more diatonic notes are spelled correctly, although the results do not show a linear correlation.

From these results, we don't know yet which part of all errors made by the pitch spelling algorithm, are incorrectly spelled diatonic notes, and which part are incorrectly spelled non-diatonic notes. We have therefore calculated the percentages of diatonic notes among the incorrectly spelled notes by the compactness algorithm. This last experiment has been rerun a couple of times, for different values of n. In table III the average results of these experiments can be found for different n. The observation that the majority of the incorrectly spelled pitches consist of diatonic pitches can be labeled as surprising, given our expectation that diatonic notes are easier to spell than non-diatonic notes. We observe that the percentages roughly decrease as n increases, so the set of incorrectly spelled notes consists more and more of non-diatonic notes as n increases. We also know that the overall performance of the algorithm increases with n (see table II), so we can conclude that the better the performance, the more the amount of incorrectly spelled notes consist of non-diatonic notes, which are the notes that are most difficult to spell, according to our hypothesis.

TABLE III

AVERAGE PERCENTAGES OF (A) CORRECTLY SPELLED DIATONIC NOTES, AND (B) PITCH CLASSES CORRESPONDING TO DIATONIC NOTES AMONG THE TOTAL NUMBER OF INCORRECTLY SPELLED NOTES.

n	a: percentage correct spelling of diatonic pitches	b: percentage diatonic pitches among incorrectly spelled notes
2	97.14 %	77.32 %
3	96.97 %	78.31 %
4	99.23 %	59.20 %
5	98.98 %	66.78 %
6	99.43 %	51.90 %

The incorrectly spelled diatonic notes are due to the character of the pitch spelling algorithm. To give one example, in the fourth prelude in $C\sharp$ minor, the program spells a $B\sharp$ as a C because the 6-note set in which the note appears describes a more compact region in the Euler lattice if spelled as $C\sharp, C, C\sharp, A, E, F\sharp$ than if spelled as $C\sharp, B\sharp, C\sharp, A, E, F\sharp$.

The fact that the compactness algorithm has a low note accuracy for the diatonic notes does not mean that it is a bad pitch spelling algorithm; it is clearly not, since it can compete with other proposed pitch spelling algorithms [8]. It does mean that there might be an easy way to improve the overall performance of this algorithm by separating the diatonic notes from the non-diatonic notes.

VI. CONCLUDING REMARKS

In this paper we raised the question whether the search space for pitch spelling algorithms could be limited, so as to arrive at a better understanding of the pitch spelling problem and to get a more meaningful evaluation measure of pitch spelling algorithms. We motivated that diatonic notes are usually easier to be spelled than non-diatonic notes. However, even the so-called trivial spelling of the diatonic notes did not always read 100%. Extending the definition of the diatonic set, we saw that an intuitive method of pitch spelling was created, if implemented together with a key-finding algotithm.

We have seen that not all diatonic notes are spelled correctly by the compactness pitch spelling algorithm. In fact, the majority of the errors made by the pitch spelling program consisted of diatonic notes. The fact that the compactness pitch spelling algorithm has problems to spell some diatonic notes does not mean that this is the case for other pitch spelling algorithms as well. Therefore it would be interesting to include other pitch spelling algorithms in this survey which will be a direction of future research.

We can conclude that the generally used method of comparing pitch spelling models by their note accuracy, cannot be replaced by measuring the proportion of notes spelled correctly among the non-diatonic notes without losing a lot of information. However, to give a more complete evaluation of pitch spelling methods, we propose that this measure plus the information of the percentage of diatonic notes among the incorrectly spelled notes should be added to the standard evaluation method of pitch spelling algorithms. In this way a number of different measures is given to give the reader a more complete picture of the performance of the pitch spelling algorithm. Furthermore, it could then be much clearer to see what different pitch spelling algorithms could 'learn' from each other. If one algorithm would for example spell virtually all diatonic pitches correct, and another algorithm all non-diatonic pitches, these two algorithms could serve as complements of each other.

ACKNOWLEDGMENT

The author wishes to thank Darrell Conklin for the suggestion of the initial idea for this paper and some useful discussions, and David Meredith for helpful comments.

REFERENCES

[1] Siglind Bruhn. *J.S. Bach's Well-Tempered Clavier: In-depth Analysis and Interpretation*. Mainer International Ltd, Hong Kong, 1993. Transcription for the Web published 2002-2003.

[2] Emilios Cambouropoulos. Automatic pitch spelling: From numbers to sharps and flats. In *Proceedings of the VIII Brazilian Symposium on Computer Music*, Fortaleza, Brasil, 2001.

[3] Emilios Cambouropoulos. Pitch spelling, a computational model. *Music Perception*, 20(4):411–429, 2003.

[4] Elaine Chew. Modeling tonality: Applications to music cognition. In Johanna D. Moore and Keith Stenning, editors, *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society, CogSci2001*, pages 206–211, Edinburgh, Scotland, UK, 2001.

[5] Elaine Chew and Yun-Ching Chen. Determining context-defining windows: Pitch spelling using the spiral array. In *Proceedings of the 4th International Conference for Music Information Retrieval, ISMIR*, 2003.

[6] Elaine Chew and Yun-Ching Chen. Pitch spelling using the spiral array. *Computer Music Journal*, 29(2):61–76, 2005.

[7] Aline K. Honingh. Group theoretic description of just intonation. In G. Buzzanca and et al., editors, *Proceedings of the 3rd International Conference Understanding and Creating Music*, volume 3, Caserta, Italy, 2003.

[8] Aline K. Honingh. Pitch spelling using compactness. In *Proceedings of the Ninth International Conference on Music Perception and Cognition (ICMPC 9)*, Bologna, 22-26 August, 2006.

[9] Aline K. Honingh and Rens Bod. Convexity and the well-formedness of musical objects. *Journal of New Music Research*, 34(3):293–303, 2005.

[10] Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford Psychology Series No. 17. Oxford University Press, 1990.

[11] Hugh Christopher Longuet-Higgins. The perception of melodies. In Hugh Christopher Longuet-Higgins, editor, *Mental Processes: Studies in Cognitive Science*, pages 105–129. British Psychological Society/MIT Press, London, 1987. Published earlier as Longuet-Higgins 1976.

[12] Hugh Christopher Longuet-Higgins and Mark Steedman. On interpreting Bach. In Hugh Christopher Longuet-Higgins, editor, *Mental Processes: Studies in Cognitive Science*, pages 82–104. British Psychological Society/MIT Press, 1987/1971. Published earlier as Longuet-Higgins and Steedman (1971).

[13] David Meredith. Pitch spelling algorithms. In *Proceedings of the Fifth Triennial ESCOM Conference*, pages 204–207, Hanover University of Music and Drama, Germany, 2003.

[14] David Meredith. *Computing Pitch Names in Tonal Music: A Comparative Analysis of Pitch Spelling Algorithms*. D.Phil. dissertation. Faculty of Music, University of Oxford, 2006.

[15] David Meredith. The ps13 pitch spelling algorithm. *Journal of New Music Research*, 35(2):121–159, 2006.

[16] David Meredith. Optimizing Chew and Chen's pitch spelling algorithm. *Computer Music Journal*, 31(2), 2007.

[17] David Temperley. What's key for key? the krumhansl-schmuckler key-finding algorithm reconsidered. *Music Perception*, 17(1):65–100, 1999.

[18] David Temperley. *The Cognition of Basic Musical Structures*. The MIT Press, 2001.

[19] Jan Van de Craats. *De fis van Euler: Een nieuwe visie op de muziek van Schubert, Beethoven, Mozart en Bach*. Aramith Uitgevers, Bloemendaal, 1989.

[20] P. G. Vos and E. W. van Geenen. A parallel-processing key-finding model. *Music Perception*, 14(2):185–224, 1996.