Subject and counter-subject detection for analysis of the Well-Tempered Clavier fugues

Mathieu Giraud[1], Richard Groult[2], and Florence Levé[2]

[1] LIFL, CNRS, Université Lille 1 and INRIA Lille, France
[2] MIS, Université Picardie Jules Verne, Amiens, France

Abstract. Fugue analysis is a challenging problem. We propose an algorithm that detects subjects and counter-subjects in a symbolic score where all the voices are separated, determining the precise ends and the occurrence positions of these patterns. The algorithm is based on a diatonic similarity between pitch intervals combined with a strict length matching for all notes, except for the first and the last one. On the 24 fugues of the first book of Bach's *Well-Tempered Clavier*, the algorithm predicts 66% of the subjects with a musically relevant end, and finally retrieves 85% of the subject occurrences, with almost no false positive.

Keywords: symbolic music analysis, contrapuntal music, fugue analysis, repeating patterns

1 Introduction

Contrapuntal music is a polyphonic music where each individual line bears interest in its own. Bach fugues are a particularly consistent model of contrapuntal music. The fugues of Bach's *Well-Tempered Clavier* are composed of two to five voices, appearing successively, each of these voices sharing the same initial melodic material: a subject and, in most cases, a counter-subject. These patterns, played completely during the exposition, are then repeated all along the piece, either in their initial form or more often altered or transposed, building a complex harmonic network.

To analyze symbolic scores with contrapuntal music, one can use generic tools detecting repeating patterns or themes, possibly with approximate occurrences. Similarity between a pattern and several parts of a piece may be computed by the Mongeau-Sankoff algorithm [20] and its extensions or by other methods for approximate string matching [7, 8], allowing a given number of restricted mismatches. Several studies focus on finding *maximal repeating patterns*, limiting the search to *non-trivial* repeating patterns, that is discarding patterns that are a sub-pattern of a larger one with the same frequency [13, 14, 16, 17]. Other studies try to find musically significant *themes*, with algorithms considering the number of occurrences [25], but also the melodic contour or other features [18].

Some MIR studies already focused on contrapuntal music. The study [26] builds a tool to decide if a piece is a fugue or not, but no details are given on the

algorithm. The bachelor thesis [1] contains a first approach to analyze fugues, including voice separation. For sequence analysis, it proposes several heuristics to help the selection of repeating patterns inside the algorithms of [13] which maximizes the number of occurrences. The website [10] also produces an analysis of fugues, extracting sequences of some repeating patterns, but without precise formal analysis nor precise bounds.

One can take advantage of the apparently simple structure of a fugue: as the main theme – the subject – always begins at only one voice, this helps the analysis. But a good understanding of the fugue requires to find *where the subject exactly ends*. In this work, we start from a symbolic score which is already track-separated, and we propose an algorithm to sketch the plan of the fugue. The algorithm tries to retrieve the *subjects* and the *counter-subjects*, precisely determining the *ends* of such patterns. We tested several substitution functions to have a sensible and specific approximate matching. Our best results use a simple *diatonic similarity* between pitch intervals [4] combined with a strict length matching for all notes, except for the first and the last one.

The paper is organized as follows. Section 2 gives definitions and some background on fugues, Section 3 details the problem of the bounds of such patterns, Section 4 presents our algorithm, and Section 5 details the results on the 24 fugues of the first book of Bach's *Well-Tempered Clavier*. These results were evaluated against a reference musicological book [2]. The algorithm predicts two thirds of the subjects with a musically relevant end, and finally retrieves 85% of the subject occurrences, with almost no false positives.

2 Preliminaries

A *note* x is described by a triplet (p, o, ℓ), where p is the pitch, o the onset, and ℓ the length. The pitches can describe diatonic (based on note names) or semitone information. We consider ordered *sequence of notes* $x_1 \dots x_m$, that is $x_1 = (p_1, o_1, \ell_1), \dots, x_m = (p_m, o_m, \ell_m)$, where $1 \leq o_1 \leq o_2 \leq \dots \leq o_m$ (see Fig. 1). The sequence is *monophonic* if there are never two notes sounding at the same onset, that is, for every i with $1 \leq i < m$, $o_i + \ell_i \leq o_{i+1}$. To be able to match transposed patterns, we consider relative pitches, also called *intervals*: the interval sequence is defined as ${}^{\Delta}x_2 \dots {}^{\Delta}x_m$, where ${}^{\Delta}x_i = ({}^{\Delta}p_i, o_i, \ell_i)$ and ${}^{\Delta}p_i = p_i - p_{i-1}$.

We now introduce some notions about fugue analysis (see for example [2, 23] for a complete musicological analysis). These concepts are illustrated by an example on Fugue #2, which has a very regular construction.

A *fugue* is given by a set of *voices*, where each voice is a monophonic sequence of notes. In Bach's *Well-Tempered Clavier*, the fugues have between 2 and 5 voices, and Fugue #2 is made of 3 voices.

The fugue is built on a theme called *subject* (S). The three first *occurrences* of the subject in Fugue #2 are detailed in Fig. 2: the subject is *exposed* at one voice

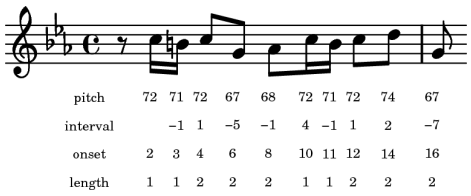


Fig. 1. A monophonic sequence of notes (start of Fugue #2, see Fig. 2), represented by (p, o, ℓ) or $({}^{\Delta}p, o, \ell)$ triplets. In this example, onsets and lengths are counted in sixteenths, and pitches and intervals are counted in semitones through the MIDI standard.

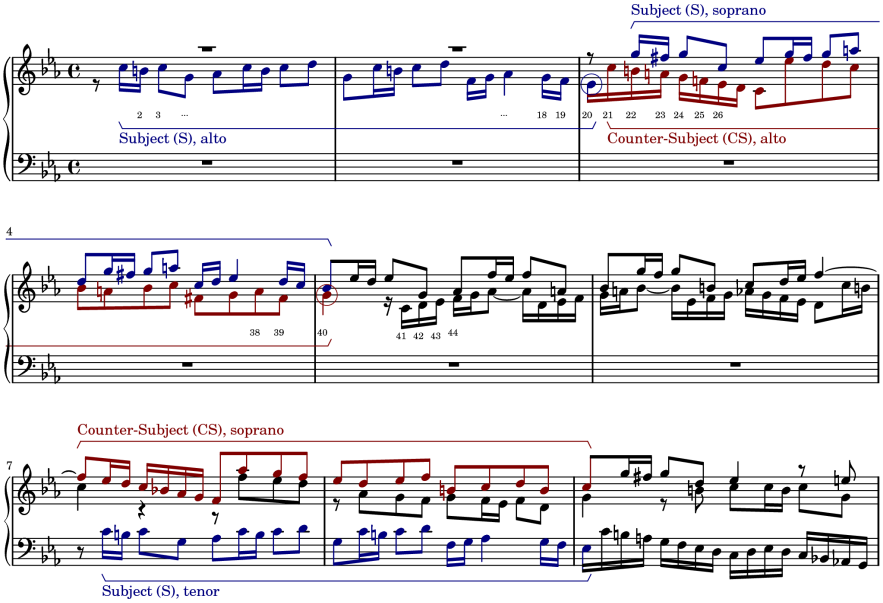
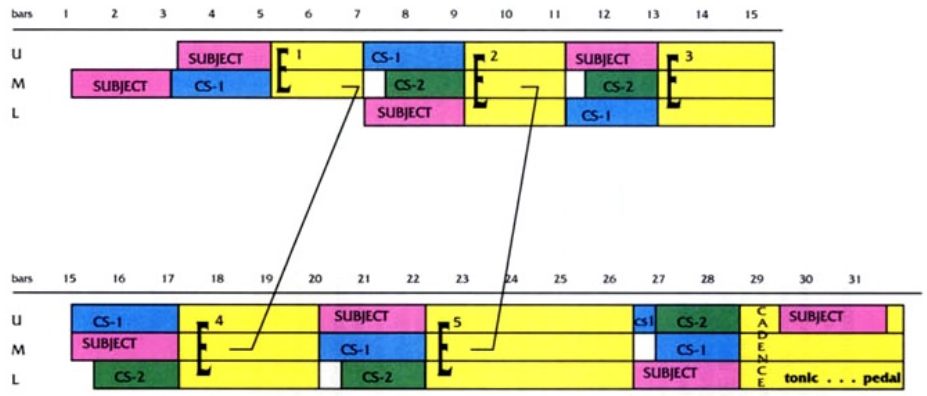


Fig. 2. Start of Fugue #2 in C minor (BWV 847).

(the alto), beginning by a C, until the second voice enters (the soprano, measure 3). The subject is then exposed at the second voice, but is now transposed to G. Meanwhile, the first voice continues with the first *counter-subject* (CS) that combines with the subject. Fig. 3 shows a sketch of the entire fugue. The fugue alternates between other instances of the subject together with counter-subjects (8 instances of S, 6 instances of CS, and 5 instances of the *second counter-subject* CS2) and development on these same patterns called *episodes* (E).

All these instances are not exact ones – the patterns can be transposed or altered in various ways. As an example, Fig. 4 shows the five complete occurrences of CS. For these occurrences, the patterns can be (diatonically) transposed, and the lengths are conserved except for the first and last note.




```
            1   2   3   4   5   6   7   8   9  10   1   2   3   4
soprano             S17-----       C19-----        S19-----
   alto     S19-----CS19----
  tenor                                S19-----        C19-----
            I       V                  I               III

5   6   7   8   9  20   1   2   3   4   5   6   7   8   9  30   1
CS18----            S19-----                               S19-----
S17-----            CS19----
                                                S19-----
V                   I                           I           I
```

Fig. 3. Analysis of Fugue #2 in C minor (BWV 847). Top: diagram summarizing the analysis by S. Bruhn, used with permission [2], [3, p. 80]. Bottom: output of the proposed algorithm, retrieving all occurrences of S (and their degrees) and all but one occurrences of CS. The numbers indicate the pitch intervals exactly matching (in a diatonic way) those of the patterns (out of 19 for S). The two `S17` occurrences correspond thus to approximate matches of the subject (tonal answers).

3 Where does the subject end?

A fundamental question concerns the precise *length* of the subject and of any other interesting pattern. The subject is heard alone at the beginning of the first voice, until the second voice enters. However, this end is generally not exactly at the start of the second voice.

Formally, let us suppose that the first voice is $x_1, x_2, ...$, and the second one is $y_1, y_2, ...$, with $x_i = (p_i, o_i, \ell_i)$ and $y_j = (p'_j, o'_j, \ell'_j)$. Let x_z be the last note of the first voice heard before or at the start of the second voice, that is $z = \max\{i \mid o_i \leq o'_1\}$. The end of the subject is roughly at x_z. Table 2, at the end of the paper, lists the exact values of g such that the *true* subject ends at x_{z+g}:

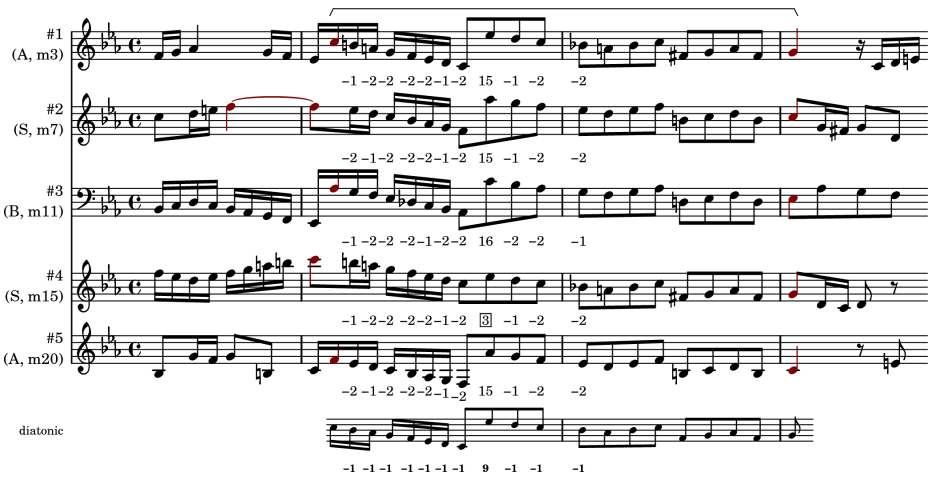


Fig. 4. The 5 complete occurrences of the first counter-subject into Fugue #2 in C minor (BWV 847). (Note that this counter-subject actually has a latter occurrence, split between two voices.) In these occurrences, all notes – except the first and the last ones – have exactly the same length. The values in the occurrences indicate the intervals, in number of semitones, inside the counter-subject. Only occurrences #2 and #5 have exactly the same intervals. The occurrence #4 is almost identical to occurrence #1, except that it lacks the octave jump (+3 instead of +15). Between groups {#1, #4}, {#2, #5}, and {#3}, the intervals are not exactly the same. However, all these intervals (except the lack of the octave jump in #4) are equal when one considers only diatonic information (bottom small staff): clef, key and alterations are here deliberately omitted, as semitone information is not considered.

in the first book of Bach's *Well-Tempered Clavier*, we notice that g is always between -8 and $+6$, and, in the majority of cases, between -4 and $+1$.

For example, in the Fugue #2, the subject has 20 notes, ending on alto note x_{20} (the first sixteenth of the third measure, E♭, first circled note on Fig. 2), that is 2 notes before the start of the soprano voice ($g = -2$). This can be deduced from many observations in this third measure:

- metrically, the phrase ends on a strong beat;
- harmonically, the five preceding notes "F G A♭ G F" suggest a 9th dominant chord, which resolves on the E♭ suggesting the C minor tonic;
- moreover, the subject ends with a succession of sixteenths with small intervals, whereas the following note x_{21} (C) belongs to CS with the line of falling sixteenths.

Determining the precise end of the subject is thus an essential step in the analysis of the fugue: it will help to localize the counter-subject and build the structure with all occurrences of these patterns, but also to understand the rhythm, the harmony and the phraseology of the whole piece.

We could use generic algorithms to predict the subject end. For example, the "stream segment detection" described in [24] considers melody, pitch and rhythm informations. Many different features are also discussed in [18] for theme extraction. However, in the following, we will show that a simple algorithm, only based on similarities, is able to detect precisely most of the subject ends.

4 Algorithm

Starting from track-separated data, we propose here to detect the subject as a repeating pattern finishing approximatively at the start of the second voice, under a substitution function considering a diatonic similarity for pitch intervals, and enforcing length equalities of all notes except the first one and the last one.

The similarity score between a pattern and the rest of the fugue piece can be computed via dynamic programming by the Mongeau-Sankoff equation [20]. The alignment can then be retrieved through backtracking in the dynamic programming table.

As almost all the content of a fugue is somewhat derived from a subject or some counter-subject, any part will match a part of the subject or of another base pattern within a given threshold. Here, we will use very conservative settings – only substitution errors, and strict length requirements – to have as few false positives as possible, still keeping a high recognition rate.

Subject identification. To precisely find the end the subject, we thus want to test patterns finishing at notes x_{z+g}, where $g \in [g_{\min}, g_{\max}] = [8, +6]$. Each one of these candidates is matched against all the voices. In this process, we use a substitution cost function able to match the first and the last notes of the subject independently of their lengths.

Let $S(a, b)$ be the best number of matched intervals when aligning the start $x_1 \dots x_a$ of a pattern (the subject) against a part of a given voice finishing at y_b, and $S_f(a, b)$ the best number of matched intervals when aligning a complete pattern (the complete candidate subject) $x_1 \dots x_a$ against the same part. These tables S and S_f may be computed by the following dynamic programming equation:

$$\begin{cases} S(1, b) = 0 \\ \forall a \geq 2, S(a, b) = S(a-1, b-1) + \delta({}^{\Delta}x_a, {}^{\Delta}y_b) & \text{(match, substitution)} \\ \forall a \geq 2, S_f(a, b) = S(a-1, b-1) + \delta_f({}^{\Delta}x_a, {}^{\Delta}y_b) & \text{(finishing)} \end{cases}$$

The substitution functions δ and δ_f are detailed on Fig. 5: δ checks pitch intervals and lengths, whereas δ_f only considers pitch intervals. The length of the first notes (x_1 and y_1) is neither checked, as the algorithm actually compares ${}^{\Delta}x_2 \dots {}^{\Delta}x_a$ against ${}^{\Delta}y_2 \dots {}^{\Delta}y_b$. Finally, notice that these equations only use substitution operations, but can be extended to consider other edit operations.

$$\delta((^{\Delta}p, o, \ell), (^{\Delta}p', o', \ell')) = \begin{cases} +1 & \text{if} \quad {}^{\Delta}p \approx {}^{\Delta}p' \quad \text{and} \quad \ell = \ell' \\ 0 & \text{if} \quad {}^{\Delta}p \not\approx {}^{\Delta}p' \quad \text{and} \quad \ell = \ell' \\ -\infty & \text{otherwise} \end{cases}$$

$$\delta_f((^{\Delta}p, o, \ell), (^{\Delta}p', o', \ell')) = \begin{cases} +1 & \text{if} \quad {}^{\Delta}p \approx {}^{\Delta}p' \\ 0 & \text{otherwise} \end{cases}$$

Fig. 5. Substitution operations between intervals. The actual comparison of length ($\ell = \ell'$) also checks the equality of the rests that may be immediately before the compared notes. The relation $\approx$ is a similarity relation on pitch intervals.

As in [13], we only compute once each table (for a given voice), then we scan the table S_f to find the occurrences: given a sequence x and a threshold τ, the candidate finishing at x_{z+g} occurs in the sequence y if for some position i in the text, $S_f(z+g, i) \geq \tau$. The best candidate is selected on the total number of matched intervals in all occurrences. We call x_s its last note, so the subject is defined to be $x_1 ... x_s$. The whole algorithm is in $O(mn)$, where $m = z + g_{\max}$.

For example, on the Fugue #2, the algorithm correctly selects the note x_{20} as the end of the subject (see Table 1).

$z+g$	14	15	16	17	18	19	**20**	21	22	23	24	25	26	27	28
g	-8	-7	-6	-5	-4	-3	$\mathbf{-2}$	-1	0	$+1$	$+2$	$+3$	$+4$	$+5$	$+6$
occ.	8	8	8	8	8	8	**8**	3	3	3	3	2	2	2	2
score	100	108	116	124	132	140	**148**	59	61	63	65	48	50	52	54

Table 1. Occurrences and scores when matching all candidate subjects in Fugue #2. The score is the sum of the $S_f(z+g, i)$ values at least equal to τ: it is the total number of intervals exactly matched on all occurrences. Here this end corresponds to the "non-trivial maximal-length repeating pattern" for most occurrences, but it is not always the case.

Interval similarities and diatonic matching. The equation Fig. 5 needs to have a similarity relation $\approx$ on pitch intervals. Between a strict pitch equality and very relaxed "up/down" classes defining the contour of some melody [9], some intermediary interval classes may be defined as "step/leap intervals" [5] or "quantized partially overlapping intervals" (QPI) [15].

We propose here to use a similarity on *diatonic pitches*. Such a pitch representation is often mentioned [21, 22] and was studied in [4, 6, 12]. A diatonic model is very relevant for tonal music: it is sensible enough to allow mode changes, while remaining specific enough – a scale will always match only a scale. For example, with diatonic similarity, all occurrences but one of the counter-subject on the Fig. 4 can be retrieved exactly, and the occurrence #4 with only one substitution.

Counter-subjects identification. The same method as for subject identification is used to retrieve the first counter-subject. The first occurrence of the counter-subject starts right after the subject (at x_{s+1}), and its length is approximatively equal to the length of the subject. We thus have a rough end of the counter-subject at x_w, where $w = \max\{i \mid o_i - o_{s+1} \leq o_s - o_1\}$, and the same procedure refines the bound to find the good last note x_{cs}, where again $cs - w$ is in a given interval. To prevent detection of non-relevant patterns, the counter-subject is marked as not detected if the above procedure leads to more occurrences than the subject occurrences.

5 Results and discussion

We tested the algorithm of the previous section on the 24 fugues of the first book of Bach's *Well-Tempered Clavier*, starting from Humdrum files where the voices are separated, available for academic purposes at `http://kern.humdrum.org/`. The pitches were encoded according to two frameworks: MIDI encoding, and Base40 encoding [11]. While the first one only counts semitones, the second one allows to discriminate enharmonic pitches, thus allowing a precise diatonic match as described in the previous section.

We ran the algorithm on the 24 fugues[3], and manually checked all results and occurrences. Results (with diatonic similarity) are summarized on Table 2. We fixed a minimum threshold of $\tau = 0.9z - 3$, where z is the number of notes defined in Section 3.

Subject lengths. We searched for end of subjects in the range $[g_{\min}, g_{\max}] = [-8, +6]$, that are the observed values. In 16 of the 24 fugues, the algorithm retrieves precisely the ends of the subjects. To our knowledge, this is the first algorithm able to correctly detect the ends of the subject: In [1], the subjects found are said to be "missing or including an extra 1 to 4 notes", and the ends of the subjects on [10] are also very approximate.

Fugue #8 shows why the proposed algorithm does not always find the correct length of the subject. In this fugue, a subject of length 9 notes is found instead of 13 notes: there are several truncated occurrences of the subject, and the algorithm chooses the end that provides the best match throughout the piece (Fig. 6).

The algorithm already considers the last note in a special way (and the former notes can be handled through substitution errors in the pitch intervals). It is possible to adapt the matching to be even more relaxed towards the end of the pattern, but we did not see a global improvement in the detection of subject lengths.

False positives. There are very few false positives among the subjects found (specificity of 90%), even when the length of the subject is badly predicted. The false positives appear in only two fugues:

[3] A part of the output of the algorithm is shown at the bottom of Fig. 3, and the full output on all the 24 fugues is available at `http://www.lifl.fr/~giraud/fugues`.

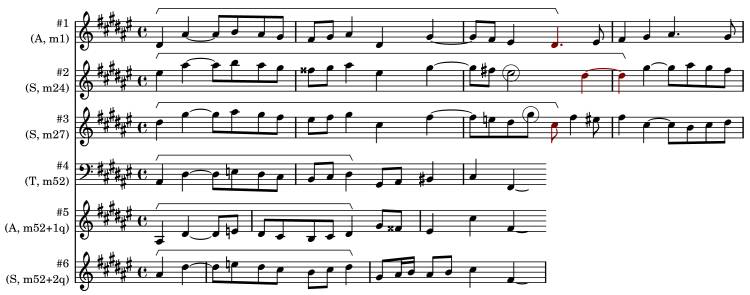


Fig. 6. Some subject occurrences in Fugue #8 in D# minor. The occurrence #1 is the first one, and is similar to 16 occurrences, sometimes with diatonic transpositions. In the occurrence #2, the last but one note of the subject (circled E) has not the same length than in the other occurrences (and this is forbidden by our substitution function δ). In the occurrence #3, a supplementary note (circled G) is inserted before the end of the subject, again preventing the detection if the true length of the subject is considered. Moreover, the occurrences #3, #4 and #5 are truncated to the head of the subject, and lead to a false detection of subject length.

- in Fugue #19, the 5 false positives correspond to 4 extended subjects [2], and one almost complete subject.
- in Fugue #5, the length of the subject is wrongly selected to the first 9 notes (8 first thirty-second notes and a final note), and this head of the subject matches the 11 true occurences, but also 24 false positives.

False negatives. The algorithm correctly retrieves about 85% of the subject occurrences. The false negatives are occurrences that are too much altered: insertions, deletions, or too many substitutions compared to the threshold.

Inverted and augmented subjects. In some fugues, the subject appears upside down (all its intervals are reversed) or augmented (all lengths are doubled). Once the subject is known, the same matching algorithm can thus be applied to the inversion or the augmentation of the subject. This method never produced a false positive, and was able to recover 72% (26/36) of the complete inverted and augmented subjects reported in [2].

Counter-subjects. Counter-subjects were detected with the same algorithm within the range $[g_{\min}, g_{\max}] = [-2, +4]$. In 40% of the fugues, the algorithm correctly detects the exact length of the CS or the absence of a CS.

In 9 fugues, the algorithm predicts the absence of CS. This was expected for Fugues #1, #8 (no CS), #15 (the CS occurs completely only once) #19 (late exposition of CS) and #20 (there is no real "characteristic and independent counter-subject" according to [2]). As in the case of the subjects, there are false negatives due to the bad recogntion of altered patterns. Moreover, when the subject is badly detected, the detection of the counter-subject end fails in the majority of the cases.

The algorithm retrieves correctly about the half of the CS occurrences, with more than 80% specificity.

Pitch interval similarities. We compared the diatonic matching against a simple exact matching on MIDI semitones, possibly adapting the error threshold. As expected, diatonic similarity has a better performance, because such a relaxed similarity is able to match approximate occurrences as the counter-subjects shown on Fig. 4.

Starting from MIDI pitches, an idea could be thus to use pitch spelling methods as [19]: such methods are almost perfect and provide the diatonic spelling of some pitches. However, we also tested a pseudo-diatonic matching on semitone information – considering as similar the intervals that differ from at most 1 semitone. The results (not shown) are very similar to those with true diatonic matching.

Other edit operations. Finally, we also tested other edit operations. The equations of Fig. 5 consider only substitutions, and can be simply extended to include the full Mongeau-Sankoff edit operations [20]. For instance, using insertions and allowing rhythm substitutions will, starting from the true subject, retrieve the occurrences #2 and #3 in Fig. 6. However, in the general case, insertions or deletions destroy the measure, leading to bad results on the predicted subject lengths.

More musical operations (fragmentation, consolidation), with fine-tuned costs, give a slight advantage in some of the 24 fugues, but this has not been reported here to keep the simplicity of the algorithm.

6 Conclusions

A complete fugue analysis tool should use any available information, including pattern repetition, harmonic analysis and phrasing considerations.

In this work, we focused only on pattern repetition. Our simple algorithm, based on the total number of matched intervals in all occurrences of patterns, allows to find precise ends of subjects and first counter-subjects in the majority of cases. This model considers a unique substitution operation with a diatonic similarity, enforcing the equality of lengths for all notes except the first and the last ones.

Extensions could include a study on the second counter-subject and on other inferred patterns. Combined with other techniques, this algorithm could lead to a more complete automatic fugue analysis pipeline.

Track-separated data. The current algorithm works on track-separated data. Starting from plain MIDI files, we could use voice separating algorithms. Although it would be a challenging problem to adapt our algorithm to directly treat standard polyphonic MIDI files, we first want to improve the current approach to complete our comprehension of any fugue.

Studying other fugues. Finally, it would be interesting to study the efficiency of our algorithm on other fugues than Bach's Well-Tempered Clavier, keeping in mind some pratical limitations (availability of track-separated files, ground truth). As far as the fugues keep the strict structure with a clear subject exposition, we are confident that our algorithm should give good results. As an example, the website `http://www.lifl.fr/~giraud/fugues` shows the output on the fugue in Mozart's *Adagio and Fugue in C minor*, K 546. We plan to further experiment it on other baroque or classical fugues, or on more recent corpus such as the Shostakovitch preludes and fugues (op. 87).

References

1. Lisa Browles. Creating a tool to analyse contrapuntal music. Bachelor Dissertation, University of Bristol, 2005.
2. Siglind Bruhn. *J. S. Bach's Well-Tempered Clavier. In-depth Analysis and Interpretation.* 1993. ISBN 962-580-017-4, 962-580-018-2, 962-580-019-0, 962-580-020-4. Available online at `http://www-personal.umich.edu/~siglind/text.htm`.
3. Siglind Bruhn. *J. S. Bachs Wohltemperiertes Klavier, Analyse und Gestaltung.* Edition Gorz, 2006. ISBN 3-938095-05-9.
4. Emilios Cambouropoulos. A general pitch interval representation: Theory and applications. *Journal of New Music Research*, 25(3):231–251, 1996.
5. Emilios Cambouropoulos, Maxime Crochemore, Costas S. Iliopoulos, Manal Mohamed, and Marie-France Sagot. A pattern extraction algorithm for abstract melodic representations that allow partial overlapping of intervallic categories. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2005)*, pages 167–174, 2005.
6. Emilios Cambouropoulos and Costas Tsougras. Influence of musical similarity on melodic segmentation: Representations and algorithms. In *Sound and Music Computing (SMC 04)*, 2004.
7. Raphaël Clifford and Costas S. Iliopoulos. Approximate string matching for music analysis. *Soft. Comput.*, 8(9):597–603, 2004.
8. T. Crawford, C. Iliopoulos, and R. Raman. String matching techniques for musical similarity and melodic recognition. *Computing in Musicology*, 11:71–100, 1998.
9. Asif Ghias, Jonathan Logan, David Chamberlin, and Brian C. Smith. Query by humming: musical information retrieval in an audio database. In *ACM Multimedia*, pages 231–236, 1995.
10. J. Hakenberg. The Pirate Fugues. `http://www.hakenberg.de/music/music.htm`.
11. Walter B. Hewlett. A base-40 number-line representation of musical pitch notation. *Musikometrika*, 4(1-14), 1992.
12. Yuzuru Hiraga. Structural recognition of music by pattern matching. In *International Computer Music Conference (ICMC 97)*, 1997.
13. J. L. Hsu, C. C. Liu, and A. Chen. Efficient repeating pattern finding in music databases. In *International Conference on Information and Knowledge Management (CIKM 1998)*, 1998.
14. Ioannis Karydis, Alexandros Nanopoulos, and Yannis Manolopoulos. Finding maximum-length repeating patterns in music databases. *Multimedia Tools Appl.*, 32:49–71, 2007.

15. Kjell Lemström and Pauli Laine. Musical information retrieval using musical parameters. In *International Computer Music Conference (ICMC '98)*, pages 341–348, 1998.
16. Chih-Chin Liu, Jia-Lien Hsu, and Arbee L.P. Chen. Efficient theme and non-trivial repeating pattern discovering in music databases. In *15th International Conference on Data Engineering (ICDE 99)*, pages 14–21, 1999.
17. Yu lung Lo and Chun yu Chen. Fault tolerant non-trivial repeating pattern discovering for music data. In *IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAR 2006)*, pages 130–135, 2006.
18. Colin Meek and William P Birmingham. Automatic thematic extractor. *Journal of Intelligent Information Systems*, 21(1):9–33, 2003.
19. David Meredith. Pitch spelling algorithms. In *5th Triennal ESOM Conference*, 2003.
20. Marcel Mongeau and David Sankoff. Comparaison of musical sequences. *Computer and the Humanities*, 24:161–175, 1990.
21. Keith S. Orpen and David Huron. Measurement of similarity in music: A quantitative approach for non-parametric representations. *Computers in Music Research*, 4:1–44, 1992.
22. Sami Perttu. Combinatorial pattern matching in musical sequences. Master Thesis, University of Helsinki, 2000.
23. Ebenezer Prout. *Analysis of J.S. Bach's forty-eight fugues (Das Wohltemperirte Clavier)*. E. Ashdown, London, 1910.
24. Dimitrios Rafailidis, Alexandros Nanopoulos, Yannis Manolopoulos, and Emilios Cambouropoulos. Detection of stream segments in symbolic musical data. In *Int. Society for Music Information Retrieval Conf. (ISMIR 2008)*, 2008.
25. Lloyd Smith and Richard Medina. Discovering themes by exact pattern matching. In *Int. Symposium for Music Information Retrieval (ISMIR 2001)*, pages 31–32, 2001.
26. Pei-Hsuan Weng and Arbee L. P. Chen. Automatic musical form analysis. In *International Conference on Digital Archive Technologies (ICDAT 2005)*, 2005.

#	BWV	tonality	voices	S				CS				remarks
				s	g	s'	occ.	cs	g	cs'	occ.	
1	846	C major	4	14	−2	14	21/23					
2	847	C minor	3	20	−2	20	8/8	40	0	40	5/6	
3	848	C# major	3	17	−5	17	12/12	44	0	**42**	7/11	wrong CS
4	849	C# minor	5	5	0	5	14/29	19	+4	19	2/2	
5	850	D major	4	13	−2	**9**	**35**/11	19	0	**15**	8/9	wrong S, S: 24 FP wrong CS, CS: 4 FP
6	851	D minor	3	12	0	12	11/11 $3^i/5^i$	29	0	**33**	2/3	wrong CS
7	852	Eb major	3	16	−8	16	9/9	40	0		0/6	no CS found
8	853	D# minor	3	13	0	**9**	18/19 $7^i/7^i$ (+ 2^i) $3^a/3^a$					wrong S
9	854	E major	3	6	−6	**18**	10/12	22	0		0/⋆	wrong S
10	855	E minor	2	26	+1	26	8/8	36	−2		0/7	no CS found
11	856	F major	3	15	−4	15	10/14	34	0	34	3/5	CS: 1 FP
12	857	F minor	4	11	−3	**10**	10/10	37	0	37	4/8	wrong S, good CS end
13	858	F# major	3	16	−1	16	7/8	41	0	41	2/4	
14	859	F# minor	4	18	0	18	6/7 $2^i/2^i$	44	0	**38**	5/6	wrong CS
15	860	G major	3	31	0	31	4/10 $2^i/3^i$	65	0		0/1	no CS found
16	861	G minor	4	11	0	11	14/16	22	0	22	3/10	
17	862	Ab major	4	7	0	7	15/15			**23**	3/0	wrong CS, CS: 3 FP
18	863	G# minor	4	15	−2	15	12/12	30	0	30	5/7	
19	864	A major	3	13	+6	**11**	**12**/8	21	0		0/2	wrong S, S: 5 FP
20	865	A minor	4	31	0	31	14/14 $5^i/14^i$	44	−12		0/3	no CS found
21	866	Bb major	3	38	0	38	8/8	65	0	65	7/7	
22	867	Bb minor	5	6	0	**10**	11/21			**16**	2/0	wrong S wrong CS, CS: 2 FP
23	868	B major	4	14	0	**13**	10/10 $2^i/2^i$	34	+1	**31**	3/4	wrong S, wrong CS
24	869	B minor	4	21	0	**19**	11/13	45	0		0/3	wrong S, no CS found
							288/306 (29 FP) (85% occ.) $23^i/33^i$ $3^a/3^a$				61/104 (10 FP) (49% occ.)	

^{i} : inverted subject – ^{a} : augmented subject

#7: The correct end for the CS is detected, but the actual CS begins after a small codetta.
#8: The correct end for the CS is detected, but the actual CS begins after a small codetta. Moreover, two incomplete inverted subject (also noted [2]) are detected on measure 54.
#9: The values for CS (⋆) are not counted in the total, as the CS is presented in a segmented form in almost all measures of the fugue [**?**].

Table 2. Results of the proposed algorithm on the 24 fugues of the first book of Bach's *Well-Tempered Clavier*. We take as a truth the analysis of [2], keeping here only the complete occurrence of each pattern. The columns "occ" lists the number of occurrences of Subjects and Counter-Subjects. The values s and cs indicate the index of the note ending the true subject and the counter-subject, whereas s' and cs' are the values predicted by the algorithm. See Section 3 for a definition of g. All false positives (FP) are counted in the remarks.