

# Separating Voices in Polyphonic Music: A Contig Mapping Approach

Elaine Chew and Xiaodan Wu

University of Southern California,  
Viterbi School of Engineering, Integrated Media Systems Center,  
Epstein Department of Industrial and Systems Engineering,  
3715 McClintock Avenue GER240 MC:0193,  
Los Angeles, California, USA  
{echew, xiaodanw}@usc.edu

**Abstract.** Voice separation is a critical component of music information retrieval, music analysis and automated transcription systems. We present a contig mapping approach to voice separation based on perceptual principles. The algorithm runs in  $O(n^2)$  time, uses only pitch height and event boundaries, and requires no user-defined parameters. The method segments a piece into contigs according to voice count, then reconnects fragments in adjacent contigs using a shortest distance strategy. The order of connection is by distance from maximal voice contigs, where the voice ordering is known. This contig-mapping algorithm has been implemented in VoSA, a Java-based voice separation analyzer software. The algorithm performed well when applied to J. S. Bach's *Two- and Three-Part Inventions* and the forty-eight Fugues from the *Well-Tempered Clavier*. We report an overall average fragment consistency of 99.75%, correct fragment connection rate of 94.50% and average voice consistency of 88.98%, metrics which we propose to measure voice separation performance.

## 1 Introduction

This paper presents an algorithm that separates voices in polyphonic music using basic principles of music perception and proposes metrics for evaluating the correctness of the machine-generated solutions. Creating music with multiple voices that are relatively independent is a compositional technique that results in auditory pleasure and has been practised for centuries in western music. This has led to a library of compositional rules that facilitate auditory streaming and the perception of multiple voices dating as far back as Palestrina (1526-1594) and as recently as Huron (2001, see [7]). In this paper, we use knowledge of the perceptual principles of auditory streaming to create an  $O(n^2)$  contig mapping algorithm for separating polyphonic pieces into their component voices.

Distinct from audio source separation, voice separation is the determining of perceptible parts or voices from multiple concurrently sounding streams of music. The multiple streams can originate from the same source and also be of the

same timbre. The contig mapping approach described in this paper considers only pitch height and event boundaries, ignoring information on timbre and sound source. Prior researchers (such as [8], [11] and [2]) have not reported any significant testing on large corpora because of the lack of methods for quantitative evaluation of voice separation results. We propose three metrics for quantifying the goodness of voice separation results and test the contig mapping algorithm on Johann Sebastian Bach’s 15 *Two-Part Inventions*, 15 *Three-Part Inventions* and 48 Fugues from the *Well-Tempered Clavier*.

Computationally viable and robust methods for voice separation are critical to machine processing of music. Separating music into its component voices is necessary for notating music in separate staves according to voice or instrument, or in the same staff with stems up or down depending on voice [8]. Another application related to music transcription is pitch spelling, the assignment of letter names to numeric representations for pitches or pitch classes (see for example, [3], [4] and [10]). The spelling of any given pitch is based on its tonal context as well as accepted voice leading principles. Voice separation is a precursor to incorporating voice leading spelling rules to any pitch spelling method.

Many applications in music information retrieval require the matching of *monophonic* queries to *polyphonic*<sup>1</sup> (or *homophonic*) databases, for example, query by humming applications. While other approaches to matching single line queries to multi-line records exist (see for example [9]), one approach made possible by voice separation is to first separate each piece into its component voices prior to matching the melodic query to now single-line records. Hence, a robust voice separation algorithm will vastly improve the hit rate of matching melodic queries to polyphonic databases. Another computational problem relevant to music information retrieval is the automatic detection and categorization of music by meter. Metric structure is most obvious in the lower voices and methods for meter induction can be improved by voice separation techniques.

The final example of a voice separation application is that of expressive performance. One of the main tasks of the performer or conductor is to determine the main melody or motif in any musical segment. The notes in the segment to be highlighted is often played louder or even a little before the others that are notated simultaneously in the score [6]. At other times, different voices are sounded at different volume levels to produce a foreground and background effect. Hence, machine models for voice separation are also essential to knowledge-based approaches to generating expressive performances.

As shown above, voice separation is a valuable tool in music information retrieval, automated transcription and computer analysis of music. One of the

---

<sup>1</sup> In traditional music literature, there exists a clear distinction between *polyphony* and *homophony*. Polyphonic music is multi-voice music where the voices exhibit independence relative to one another. Homophonic music, although also consisting of multiple voices, has one primary lead voice while other voices act as accompaniment to the main melody. In contrast, *heterophonic* music (less well defined) is music with one primary melody, and all accompanying voices embellishing with variants of the main theme.

easiest approaches to voice separation is to split voices according to some set of non-overlapping pitch ranges. According to [8], this is the method adopted by most commercial sequencer software packages. Needless to say, this method of separating voices can produce highly inaccurate and unsightly (in the case of automatic transcription) results. Various researchers have proposed ways to improve on this primitive approach.

In [11], Temperley proposed a preference rule approach to voice separation, incorporating the following rules for assigning voices to piano-roll representation of music: 1. avoid large leaps in any one stream; 2. minimize the number of streams; 3. minimize long breaks in streams; 4. avoid having more than one stream occupy a single square; and, 5. maintain a single top voice. Rules 1 through 4 were tested on four of Bach’s fugues. Rule 5 was found to be necessary for handling classical music; rules 1 through 5 were tested on a few classical string quartets. The errors were analyzed in terms of the number of breaks, missed or incorrect collisions and misleads. Another rule-based approach was briefly described by Cambouropoulos in [2]. This method segments the input into beats then, within each beat, connects all onsets into streams by selecting the shortest path. The crossing of streams is disallowed and the number of streams is set to be equal to the number of notes in the largest chord.

In [8], Kilian and Hoos proposed a local optimization approach to voice separation. The piece was first partitioned into slices which can contain parts that overlap (in time) with other slices. Within each slice, the notes are then separated into voices by minimizing a cost function, which assigns penalty values for undesirable features such as, overlapping notes and large pitch intervals. One flexible feature of the Kilian and Hoos model is the ability to assign entire chords to one single voice. (The cost function penalizes chord tones that are spread too far apart.) The penalty values can be adjusted by the user to achieve different tradeoffs between the features. Their algorithm was tested on selected Bach *Chorales* and Chopin *Valses*, and Bartok’s *Mikrokosmos*, and was found to be sensitive to the choice of penalty function parameters. For the purpose of automated transcription, the user can change the parameter values until a satisfactory result is achieved.

Like Temperley, our goal is to produce a correct analysis rather than an appropriate one for transcription, as is the case for Kilian and Hoos. In this paper, we propose three metrics to measure the correctness of a voice separation solution. They are: the average fragment consistency, the correct fragment connection rate and the average voice consistency. These metrics allow the algorithm’s results to be quantified objectively. Unlike Kilian and Hoos’ local optimization approach, our method does not allow synchronous notes to be part of the same voice. On the other hand, the contig mapping approach exhibits high fragment consistency, the grouping of notes from the same voice into the same fragments.

Both Temperley’s preference rule approach as well as Kilian and Hoos’ local optimization approach can potentially incur prohibitive computational costs if all possible solutions were enumerated and evaluated. Temperley utilized dynamic programming while Kilian and Hoos used a heuristically-guided stochastic

local search procedure to avoid the exponential computational cost of exhaustive enumeration. In contrast, the contig mapping approach has an  $O(n^2)$  performance and does not require approximation methods to compute a solution.

Distinct from previous approaches, our method hinges on one important feature of polyphonic music that has been ignored by other researchers. Because voices tend not to cross, when all voices are present, one can be certain of the voice ordering and assignment. We use these maximal voice segments as pillars of certainty out of which each voice connects to other members of its stream. This method requires no pre-assigned parameters or rule definitions. The perceptual rules are incorporated into the mathematical model and the algorithm has a guaranteed worst case performance of  $O(n^2)$ .

Section 2 describes the perceptual principles and the concepts underlying the contig mapping approach, and introduces the contig mapping algorithm. Section 3 presents additional details of the computer implementation of the algorithm and describes the VoSA (Voice Separation Analyzer) software. Section 4 presents our evaluation techniques and computational results. Finally, Section 5 outlines our conclusions and future work.

## 2 The Contig Mapping Approach

This section presents the contig mapping approach and its underlying perceptual principles. Section 2.1 outlines the auditory perceptual principles relevant to our approach, and Section 2.2 extracts from the principles and rules the assumptions underlying the contig mapping algorithm. Section 2.3 describes the contig mapping algorithm, including the segmentation procedure and the fragment connection policy.

### 2.1 Perceptual Principles for Voice Leading

In this section, we highlight the perceptual principles that are relevant to the contig mapping approach. Because the goal of the rules of voice leading is to create two or more concurrent yet distinct parts or voices, the same rules result in optimal auditory streaming. In [7], Huron reviews the perceptual principles for the organizing of auditory stimuli into streams and derives the rules of voice leading from these principles and empirical evidence.

The first is the pitch proximity principle. In the review, Huron reports that Bregman and his colleagues have gathered strong evidence for the pre-eminence of pitch proximity over trajectory in stream organization [1]. He argues that “the coherence of an auditory stream is maintained by close pitch proximity in successive tones within the stream,” and that this principle holds true in the music across different cultures. Thus, in determining the connections between notes that are perceived to be from the same stream, proximity should be the guiding principle.

The second is the stream crossing principle. Humans have great difficulty in tracking streams of sounds that cross with respect to pitch. Huron reports

the results of Deutsch [5] who showed that concurrent ascending and descending streams of the same timbre are perceived to switch directions at the point of crossing<sup>2</sup> as shown in the diagram on the right in Figure 1. Hence, a guiding principle in connecting notes in the same stream is that the streams should not cross.



**Fig. 1.** Possible interpretations of crossing streams.

These perceptual principles lead to numerous traditional and non-traditional rules for writing polyphonic music with perceptibly distinct parts. The ones relevant related to the pitch proximity principle are (following Huron’s numbering system):

[D6.] Avoid Unisons Rule. *Avoid shared pitches between voices.*

D10. Common Tone Rule. *Pitch-classes common to successive sonorities are best retained as a single pitch that remains in the same voice.*

D11. Conjunct Movement Rule. *If a voice cannot retain the same pitch, it should preferably move by step.*

C3. Avoid Leaps Rule. *Avoid wide pitch leaps.*

D13. Nearest Chordal Tone Rule. *Parts should connect to the nearest chordal tone in the next sonority.*

[D18.] Oblique Approach to Fused Intervals Rule. *When approaching unisons, octaves, or fifths, it is best to retain the same pitch in one of the voices.*

[D19.] Avoid Disjunct Approach to Fused Intervals Rule. *If it is not possible to approach unisons, octaves and fifths by retaining the same pitch, step motion should be used.*

while D6, D14 and D15 are encapsulated in the stream crossing principle:

[D6.] Avoid Unisons Rule. *Avoid shared pitches between voices.*

D14. Part-Crossing Rule. *Avoid the crossing of parts with respect to pitch.*

---

<sup>2</sup> A simple and informal experiment conducted on March 4th in a class of 14 students showed that this result held true even when the ascending and descending streams were played using the rhythm of the Christmas carol “Joy to the World,” where the opening melody is essentially a descending scale embellished with temporal variation. This perceptual principle is so strong that it overrode the perception of the well-known melody.

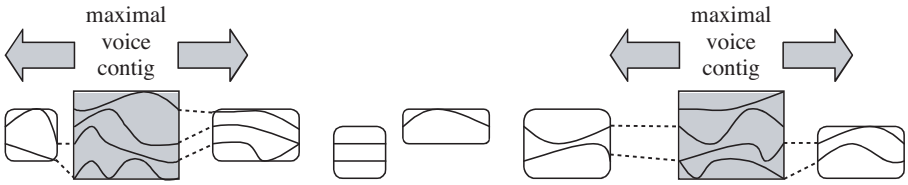
D15. Pitch Overlapping Rule. *Avoid “overlapped” parts in which a pitch in an ostensibly lower voice is higher than the subsequent pitch in an ostensibly higher voice.*

## 2.2 The Assumptions and Underlying Concept

For the purpose of the contig mapping algorithm, we translate the rules and perceptual principles detailed in Section 2.1 to the following assumptions:

1. By definition, each voice can only sound at most one note at any given time.
2. All the voices will sound synchronously at some time (we use this as a baseline count of the total number of voices present in the piece.)
3. *Pitch Proximity*: intervals are minimized between successive notes in the same stream or voice.
4. *Stream Crossing*: voices tend not to cross.

The contig mapping approach derives its method directly from these assumptions. Assumptions 1, 2 and 4 imply that, at certain segments of time, all voices will sound synchronously in a well-behaved manner. In these segments, which we call *maximal voice contigs*, we can be certain of the voice assignments for each note. Based on assumptions 3 and 4, we can use distance minimizing procedures to connect voices between segments. The maximal voice contigs seed the connection process: they act as the pillars out of which voice assignments grow at each iteration of our procedure.



**Fig. 2.** Minimum distance voice connections grow out from the maximal voice contigs

## 2.3 The Algorithm

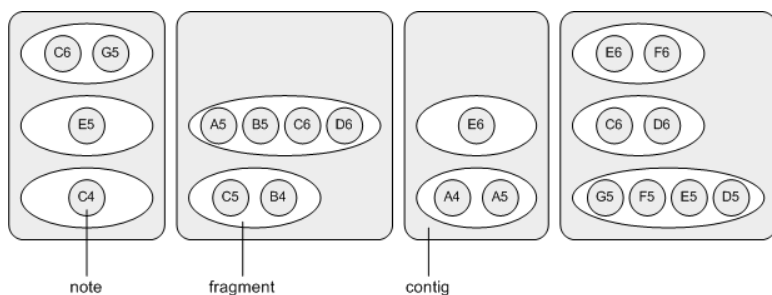
We have outlined the principles and concept behind our contig mapping approach in the previous sections. In this section, we shall provide the algorithmic details for its systematic implementation, including the procedures for segmentation and connection.

Before embarking on a description of the algorithm, we first introduce the terminology used in this section. A *note* is a musical entity with pitch and

duration properties. A *fragment* is a sequence of successive notes that belong to the same voice. A *contig*<sup>3</sup> is a collection of overlapping fragments such that the overlap depth (number of fragments present) at any time is constant. A *maximal voice contig* is a contig with the maximum number of voices present. Examples of a fragment, contig and maximal voice contig are shown in Figure 4, which corresponds to bars 24 and 25 of Bach’s Three-Part Invention (Sinfonia) No. 13 (shown in Figure 3.) In this case, both the first and last contigs are maximal voice contigs.



**Fig. 3.** Measures 24 and 25 of Bach’s Three-Part Invention No.13.

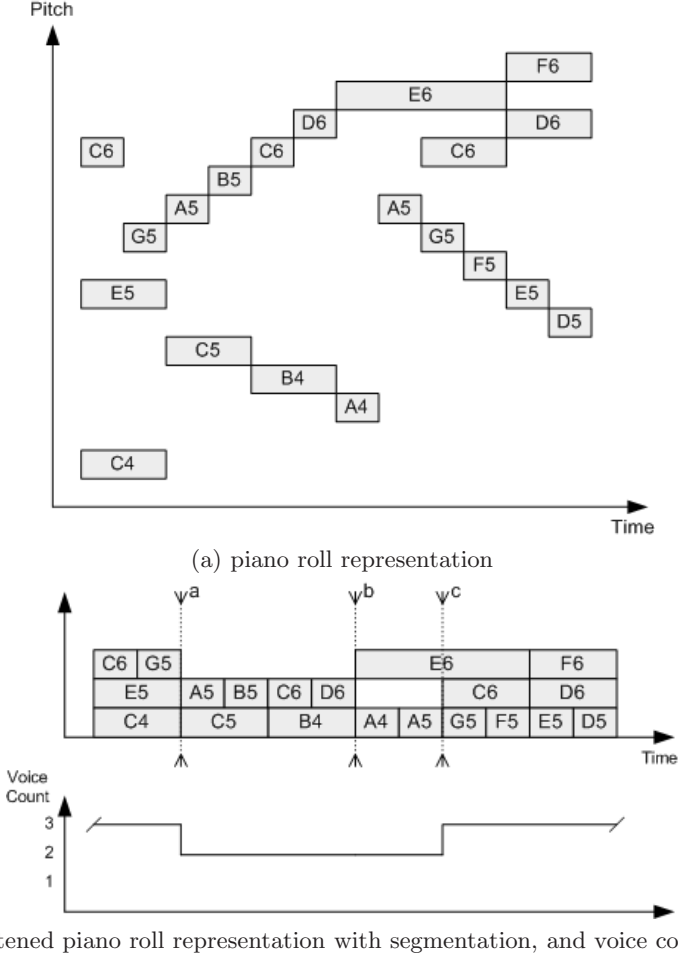


**Fig. 4.** Terminology

**Segmentation Procedure** The piece is segmented according to voice count. The segmentation procedure is best illustrated by example. The final outcome is a segmentation of the original piece into contigs such that the voice count remains constant within the contig. We return to the Bach Three-Part Invention example shown in Figure 3. Figure 5(a) shows a piano roll representation of the same

<sup>3</sup> The term *contig* is borrowed from the field of computational biology where, in DNA sequencing, the shotgun sequencing method utilizes computer algorithms to connect ordered sets of overlapping clones of DNA fragments in order to determine the DNA sequence.

excerpt. The lower half of Figure 5(b) charts the voice count at any given time while the upper half of the figure shows the flattened piano roll representation and the segmentation boundaries, marked as “a”, “b” and “c.” Boundaries a and c result from the change in voice counts, while boundary b is the results of the voice status change.



(b) flattened piano roll representation with segmentation, and voice count plot

**Fig. 5.** Example: Bach's Three-Part Invention No.13, measures 24 and 25.

More formally, if  $v_t$  represents the voice count at time slice  $t$ , the boundary between time slices  $t - 1$  and  $t$  becomes a segmentation boundary if:

- either  $v_t \neq v_{t-1}$ ;
- or  $v_t = v_{t-1}$  but the voice status changes.



A voice status change is caused by held notes that cross over a segmentation boundary, and thus are suspended over an empty slot as shown in the segment (b,c) in Figure 5(b). The held note resulted in a status change across boundary b even though the voice count does not change. As a result, b becomes a segmentation boundary. Because the note E6 crosses the boundary c, this note will be cloned, marked as being a part of a longer note and duplicated in the contigs on either side of boundary c. The resulting segmentation is shown in the contig diagram in Figure 4.

**Connection Policy** After segmentation, the maximal voice contigs seed the connection process. They act as the centers out of which the connections to neighboring contigs grow. Because voices tend not to cross and maximal voice contigs contain all voices, the voice assignment for each note in a maximal voice contig is known with high certainty. Hence, all fragments in each maximal voice contig are ordered by pitch height and assigned voice numbers corresponding to their ordering. In connecting voice fragments across neighboring contigs, we select the distance minimizing choice. Connected fragments are assigned to the same voice, and the fragment assembly process grows out from the maximal voice contigs.

Because the number of voices is usually small<sup>4</sup>, we can enumerate all possible connection combinations and select the one with the lowest penalty. Suppose we wish to connect the fragments in two neighboring contigs, X and Y, where X is followed by Y (in time). Consider a note,  $q_X$ , that is the last one from a fragment in contig X and another,  $p_Y$ , that is a first note in a fragment in contig Y. The cost of connecting  $q_X$  to  $p_Y$ ,  $c(q_X, p_Y)$ , is assigned based on the following rules:

- if the two notes are segments of the same longer note,  $c(q_X, p_Y) = -2^{31}$ ;
- if one of the two notes is null or both,  $c(q_X, p_Y) = 2^{31}$ ;
- else,  $c(q_X, p_Y)$  is the absolute difference between the pitches of the two notes.

The first rule ensures that all long notes that were previously partitioned are re-connected at this stage. The second rule forces all connectible fragments to be assigned a partner whenever one exists. And the third rule ensures minimal distance assignments.

The connection sequence grows outward from the maximal voice contigs, which act as seeds for the connection procedure. First, fragments in the immediate neighbors are connected to those in each maximal voice contig (this first level connection is illustrated in Figure 2.) Then, the second order neighbors are connected to the immediate neighbors, and so on. The assembling procedure can be viewed as a crystallization process. The maximal voice contigs act as seeds

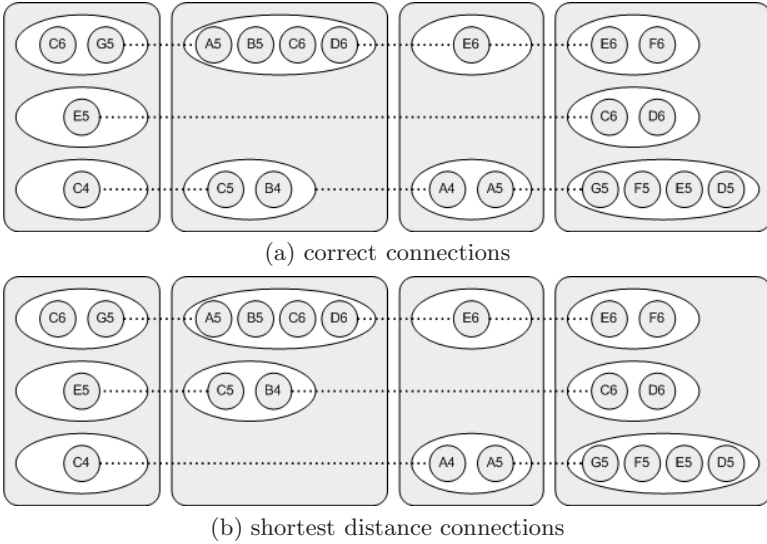
---

<sup>4</sup> According to Huron’s Principle of Limited Density [7], “If a composer intends to write music in which independent parts are easily distinguished, then the number of concurrent voices or parts ought to be kept to three or fewer.” Typically, the number of voices range from two to four, and occasionally, five or six voices are utilized. However, in the latter cases, the human ear cannot distinguish more than three or four concurrent voices at any given time.

for the process, and the contigs closer to these seeds will be connected first. The procedure ends when all contigs (or fragments in every contig) are connected.

In a piece with  $n$  notes, there can be at most  $n$  contigs. At each iteration, at least one (and at most  $n$ ) neighboring contig(s) is connected to a growing section centered around a maximal voice contig. There are at most  $n$  such iterations, hence the worst case complexity is  $O(n^2)$ .

The shortest distance connection policy produces correct groupings in the vast majority of cases. However, it is useful to note that sometimes the policy may not generate the correct solution. See, for example, the connection solutions presented in Figure 6. In the figure, dotted lines link fragments that are grouped into the same voice. The correct solution is shown in Figure 6(a) while the shortest distance solution is given in Figure 6(b). The algorithm assigns the lower fragment in the second contig to the incorrect voice. These erroneous connections are visually presented in Figure 8(b) as the four “X”’s on the left hand side. Because of the robustness of the maximal contig approach, this one incorrect assignment will not affect the majority of the notes, which are correctly grouped together according to voice.



**Fig. 6.** Connection solutions for Bach’s Three-Part Invention No.13, measures 24 and 25.

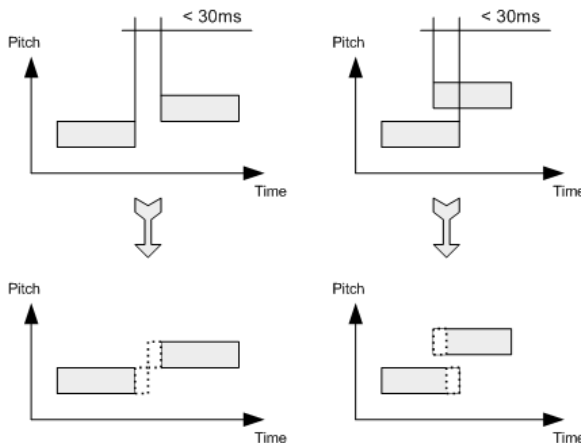
### 3 Implementation

The contig mapping approach to voice separation has been implemented in a Java application called VoSA, the Voice Separation Analyzer. The platform-

independent application was developed under Java jdk1.4.2 and runs on Windows, Mac OS and Unix machines. Its graphical user interface allows the user to visualize and evaluate the results of the voice separation algorithm. The current version of VoSA takes only MIDI input. It also has the capacity to export voice separated pieces in MIDI format and evaluation results in comma separated value (CSV) format. In this section, we present the implementation strategies not covered in the previous section’s explanation of the algorithm, and describe VoSA’s graphical user interface.

### 3.1 Quantization

Because performance artifacts and rounding errors produce overlapping notes from the same voice or gaps between successive notes, we use a selective snapping procedure to quantize the data. Since we are not concerned with beat onset irregularities, quantization only needs to occur at the boundaries with ambiguous note overlaps or gaps between note boundaries. Unlike the usual quantizing procedure of snapping the observed note boundaries to the closest unit grid, the selective snapping will only be invoked when the time difference between any two note boundaries is less than a given threshold (we used 30ms). Figure 7 shows the selective snapping quantization procedure. After quantization, the notes of the piece are stored as an ordered list sorted by onset times.



**Fig. 7.** The selective snapping quantization procedure.

### 3.2 Treatment of Ending Chordal Embellishments

In the library of contrapuntal pieces we tested, many of the polyphonic compositions have endings that are embellished with chords consisting of more notes

than the number of voices in the pieces. These ending chords serve as statements of finality but also masquerade as maximal voice contigs, causing VoSA to overestimate the number of voices in the piece and also to grow the one maximal voice contig from right to left, a highly suboptimal process. To facilitate the search for the “true” maximal voice contigs, we exclude the last three contigs to compute the maximum number of voices, and eliminate all voice fragments with an index greater than the maximum voice count. These discarded fragments (a small fraction of the total notes in the piece) will not be counted during the evaluation process.

### 3.3 User Interface

VoSA provides a graphical user interface for the user to analyze the performance of the voice separation algorithm. This graphical user interface is shown in Figure 8. The upper part of the Figure 8(a) shows the piano roll representation and the segmentation of Bach’s Three-Part Invention No.13. In the lower part of Figure 8, a graph charts the voice count at each point in time. The vertical lines in the piano roll graph shows the segmentation boundaries indexed by the contig numbers.

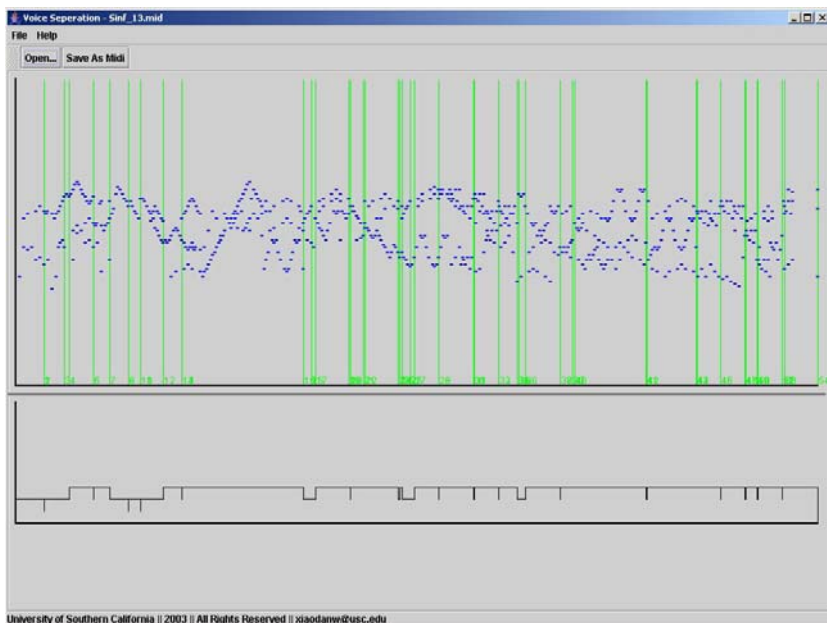
The latest version of VoSA, VoSA 3, incorporates zoom-in and zoom-out capabilities, colors voice assignments by voice, and marks the erroneous connections by a red “X.” Figure 8(b) shows a screenshot of a zoomed-in analysis of the results of voice separation for Bach’s Three-Part Invention No.13. The red X’s mark the points at which connections were incorrectly assigned.

## 4 Computational Results

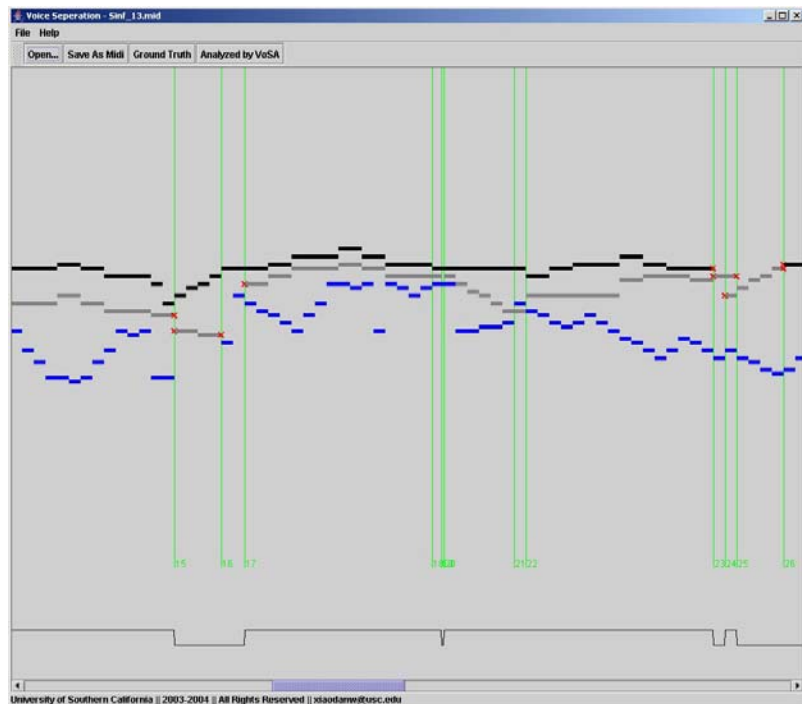
This section presents the contig mapping algorithm’s voice separation results when applied to polyphonic music by J. S. Bach, namely his *Two- and Three-Part Inventions* and Fugues from the *Well-Tempered Clavier*. Section 4.1 describes the test corpus and the acquisition of voice separation solutions. Section 4.2 lays out the evaluation procedures and Section 4.3 presents the evaluation statistics for our test corpus.

### 4.1 Test Data and Ground Truth

We test the contig mapping algorithm using Johann Sebastian Bach’s (1685-1750) 48 Fugues from his *Well-Tempered Clavier* (BWV 846-893), his *Two-Part Inventions* (BWV 772-786) and his *Three-Part Inventions* (BWV 787-801), also known as *Sinfonias*. As noted by Temperley in [11], “the correct ‘contrapuntal analysis’ for a piece is often not entirely clear. . . . One case where the correct contrapuntal analysis is explicit is Bach fugues (and similar pieces by other composers). In that case, the separate voices of the piece are usually clearly indicated by being confined to particular staves and notated with either upward or downward stems.”



(a) main screen showing segmentation and voice count



(b) the error locator screen showing voice assignments and erroneous connections (X)

**Fig. 8.** Screenshots of VoSA, the Voice Separation Analyzer

To facilitate evaluation of the voice separation procedure, we first need the ground truth, the correct assignment. An advantage of using Bach’s fugues and his two- and three-part inventions is that many MIDI renditions of these pieces exist that have been sequenced such that each voice is in a separate track. For comparison against our results, we use such track separated MIDI files. The fugues were obtained from the MuseData repository, [www.musedata.org](http://www.musedata.org), and the two- and three-part inventions from The Midi Archive at [archive.cs.uu.nl/pub/MIDI](http://archive.cs.uu.nl/pub/MIDI). We used the scores from Virtual Sheet Music, [www.virtualsheetmusic.com](http://www.virtualsheetmusic.com), for checking the voice assignments manually.

## 4.2 Evaluation Method

We use three main statistics to quantify the performance of the algorithm, namely, the *average fragment consistency*, the *correct fragment connection rate* and the *average voice consistency*. The evaluation process in VoSA records all the errors in the results and shows them visually as demonstrated in Figure 8(b). The GUI in VoSA allows the user to compare the voice assignments to the ground truth.

The *average fragment consistency* measures the overall percentage consistency over all fragments. A fragment is considered consistent if all notes in the fragment belong to the same voice. The percentage consistency of a fragment is the highest proportion of notes assigned to the same voice. This number shows the accuracy of the segmentation and fragment generation procedure. Formally, if  $V$  is the set of all voice indices,  $F$  the set of all fragments and  $vN(note)$  the true voice assignment for  $note$ , then the percentage consistency of fragment  $f$  is defined as:

$$FC(f) = \frac{100}{\|f\|} \max_{v \in V} \{\| \text{note in } f : vN(\text{note}) = v \|\},$$

where  $\|f\|$  represents the cardinality of  $f$ , the number of notes in fragment  $f$ . The average fragment consistency is given by:

$$AFC = \frac{1}{\|F\|} \sum_{f \in F} FC(f). \quad (1)$$

The *correct fragment connection rate* measures the proportion of connections that are correctly assigned. The correctness of each connection is evaluated by comparing it to the ground truth obtained a track-separated MIDI file as described in Section 4.1. To describe the mathematical formula for this quantity, we first define  $C$  to be the set of all pairs of connected fragments,  $\{(f, g) : f, g \in F \text{ and } f \text{ is connected to } g\}$  and  $vF(f)$  to be the true voice assignment for fragment  $f$ . In the case of 100% fragment consistency,  $vF(f)$  is the true voice assignment of all notes in fragment  $f$ . When a fragment has less than 100% consistency,  $vF(f)$  is the voice to which the majority of the notes in  $f$  belong. More formally,  $vF(f) = \arg \max_{v \in V} \{\| \text{note in } f : vN(\text{note}) = v \|\}$ . The correct fragment connection rate is then given by the equation:

$$CFC = \frac{100}{\|C\|} \|\{(f, g) \in C : vF(f) = vF(g)\}\|. \quad (2)$$

Finally, the *average voice consistency* measures how well the notes in the piece have been properly assigned to their appropriate voices. This quantity measures, on average, the proportion of notes from the same voice that have been assigned by the algorithm to the same voice. Again, we begin with two definitions: let  $vA(note)$  be the algorithm-assigned voice for  $note$  and  $S(v)$  be the set of notes assigned to voice  $v$ ,  $\{note : vA(note) = v\}$ . The voice consistency is defined as

$$VC(v) = \frac{100}{\|S(v)\|} \max_{u \in V} \|\{note \in S(v) : vN(note) = u\}\|,$$

and the average voice consistency is given by:

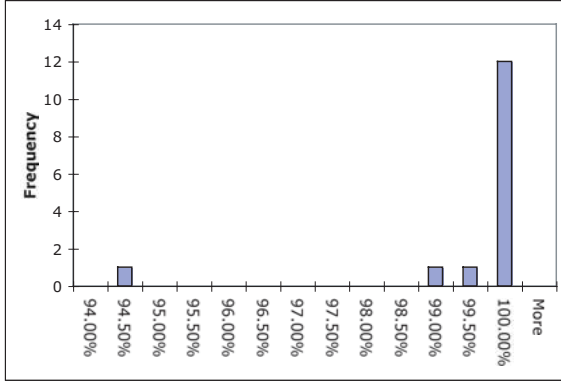
$$AVC = \frac{1}{\|V\|} \sum_{v \in V} VC(v). \quad (3)$$

### 4.3 Results

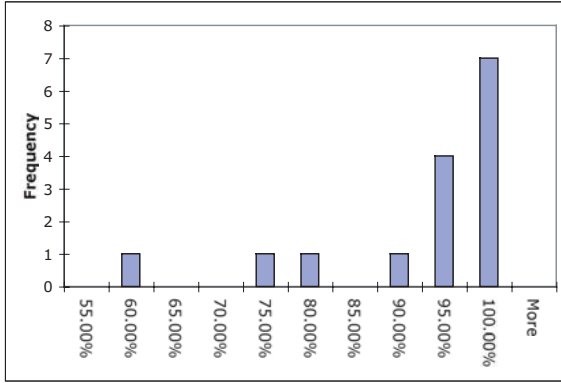
The contig mapping algorithm was tested on the 15 *Two-Part Inventions* (BWV 772-786), the 15 *Three-Part Inventions* (BWV 787-801) and the 48 Fugues from the *Well-Tempered Clavier* (BWV 846-893) by Johann Sebastian Bach (1685-1750). For each test sample, we used a quantization threshold of 30ms to pre-process the MIDI data before separating the voices using the contig mapping algorithm. We then evaluated the average fragment consistency (AFC), the correct fragment connection rate (CFC) and the average voice consistency (AVC) of the voice separation result. The distributions of these values for each test set – Two- and Three-Part Inventions and Fugues – are summarized in Figures 9, 10 and 11 respectively. The summary statistics are reported in Table 1.

The overall average fragment consistency (AFC) for the test corpus was 99.75%, that is to say, all notes in the same fragment are almost certain to be from the same voice. The overall correct fragment connection (CFC) rate was 94.50% indicating that the likelihood of connecting each fragment correctly to its contiguous strand is high. And, the overall average voice consistency (AVC) was 88.98%. Recall that this number reflects the proportion of notes in the same stream that were correctly assigned to the same voice by the algorithm. This number is lower than the AFC or CFC because each incorrect connection can result in a severe loss of voice consistency.

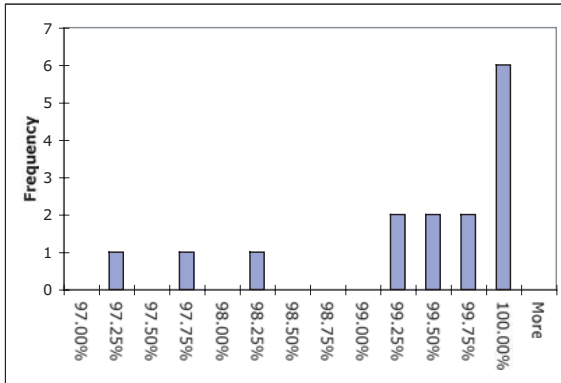
In general, higher average fragment sizes are correlated with higher average voice consistency numbers. This is not surprising considering that the average fragment consistency is extremely high. We found three possible sources for error in the contig mapping approach. The connection policy minimizes pitch distance. Even though this is generally the case, sometimes the shortest distance connection does not produce the correct result. On rare occasions, voices do cross, producing connection distances that are not minimal. Unintentional gaps between notes in the MIDI file that are not properly quantized can also lead to higher rates of error.



(a) average fragment consistency histogram (average AFC = 99.46%)



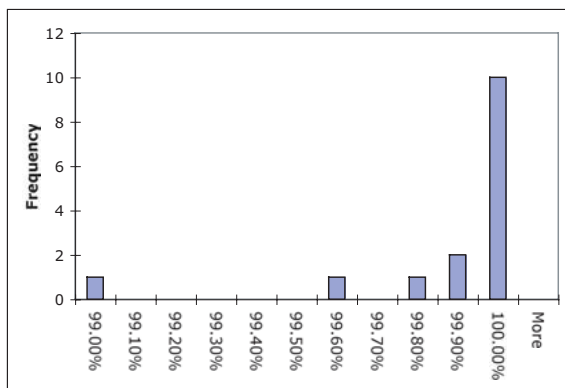
(b) average correct fragment connection histogram (average CFC = 91.47%)



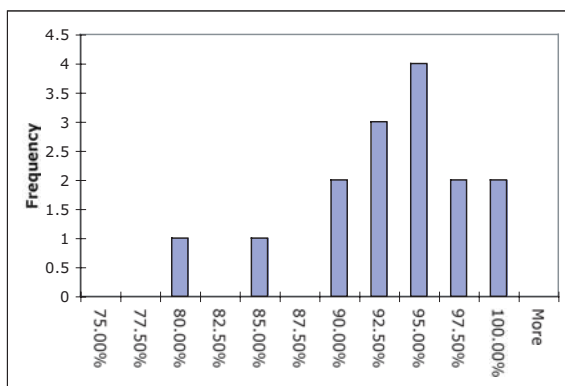
(c) average voice consistency histogram (average AVC = 99.29%)

**Fig. 9.** Voice separation results for Bach's Two-Part Inventions.

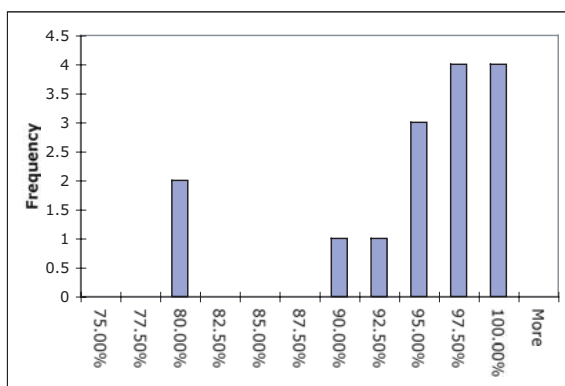




(a) average fragment consistency histogram (average AFC = 99.80%)

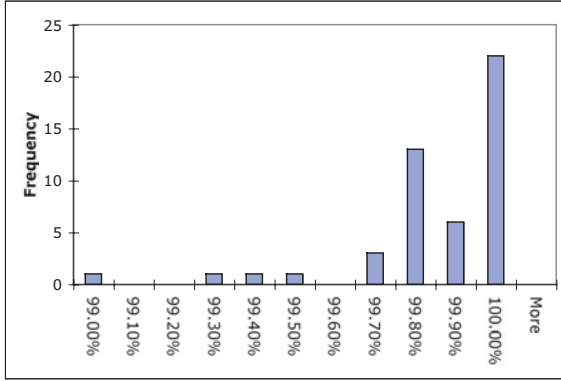


(b) average correct fragment connection histogram (average CFC = 92.27%)

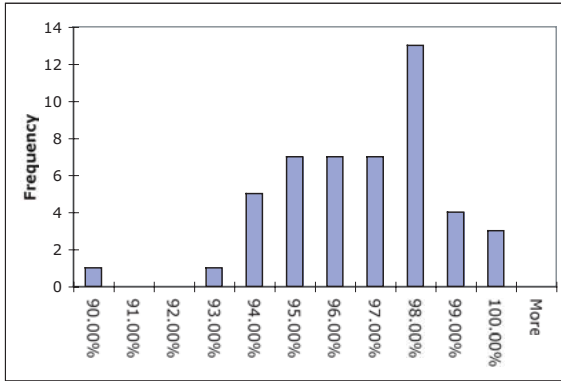


(c) average voice consistency histogram (average AVC = 93.35%)

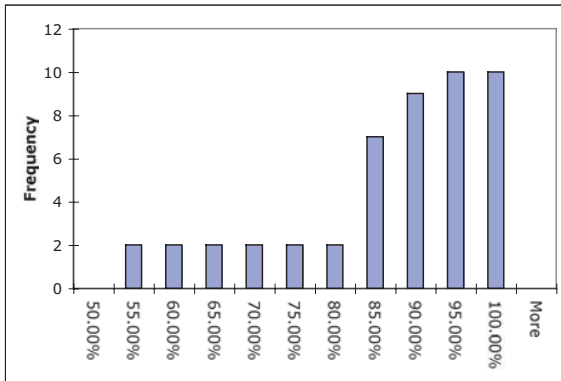
**Fig. 10.** Voice separation results for Bach's Three-Part Inventions.



(a) average fragment consistency histogram (average AFC = 99.83%)



(b) average correct fragment connection histogram (average CFC = 96.15%)



(c) average voice consistency histogram (average AVC = 84.39%)

**Fig. 11.** Voice separation results for Bach's 48 Fugues from the *Well-Tempered Clavier*.

**Table 1.** Summary statistics (average numbers) for voice separation experiments

MIDI input	no. of fragments per piece	average fragment size	no. of contigs per piece	average AFC (%)	average CFC (%)	average AVC (%)
Two-Part Inventions	46.67	18.26	32.60	99.46	91.47	99.29
Three-Part Inventions	194.67	4.28	82.33	99.80	92.27	93.35
WTC Fugues	581.81	3.05	226.50	99.83	96.15	84.39
OVERALL	404.45	6.21	161.49	99.75	94.50	88.98

## 5 Conclusions and Future Work

In this paper, we described a contig mapping approach to voice separation and three metrics for evaluating its voice separation results. The algorithm has been implemented in a voice separation analyzer application software called VoSA. We used VoSA to compute and analyze the voice separation results when the algorithm is applied to Bach’s Two- and Three-Part Inventions and Fugues. Our experiments and evaluations are the first of this scope for the testing of a voice separation algorithm. The overall statistics are promising, showing that the contig mapping approach presents a computationally viable and highly accurate solution to the voice separation problem. Future work includes the testing of the algorithm on a larger polyphonic corpus, and extending the method to homophonic music.

## 6 Acknowledgements

We acknowledge the kind assistance of Fabrizio Ferrari and Laura Caldera for giving us access to Virtual Sheet Music’s repository of classical sheet music.

The research has been funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152, and by a National Science Foundation Information Technology Research Grant No. ITR-0219912. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

## References

1. Bregman, A.: Auditory Scene Analysis: The Perceptual Organization of Sound. The MIT Press, Cambridge Massachusetts (1990) 417–442
2. Cambouropoulos, E.: From MIDI to Traditional Musical Notation. In Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis, July 30 - Aug 3, Austin, Texas (2000)

3. Cambouropoulos, E.: Pitch Spelling: A Computational Model. *Music Perception*. **20**(4) (2003) 411–429
4. Chew, E., Chen, Y.-C.: Determining Context-Defining Windows: Pitch Spelling Using the Spiral Array. In *Proceedings of the 4th International Conference on Music Information Retrieval*. (2003)
5. Deutsch, D.: Two-channel Listening to Musical Scales. *Journal of the Acoustical Society of America* **57** (1975) 1156–1160
6. Goebel, W.: Melody Lead in Piano Performance: Expressive Device or Artifact? *Journal of the Acoustical Society of America* **110**(1) (2001) 563–572
7. Huron, D.: Tone and Voice: A Derivation of the Rules of Voice-leading from Perceptual Principles. *Music Perception*. **19**(1) (2001) 1–64
8. Kilian, J., Hoos, H.: Voice Separation - A Local Optimization Approach. In *Proceedings of the 3rd International Conference on Music Information Retrieval*. (2002) 39–46
9. Lemström, K., Tarhio, J.: Detecting monophonic patterns within polyphonic sources. In *Content-Based Multimedia Information Access Conference Proceedings (RIAO 2000)*, Paris (2000) 1251–1279
10. Meredith, D.: Pitch Spelling Algorithms. In *Proceedings of the Fifth Triennial ESCOM Conference*. Hanover University of Music and Drama, Germany (2003) 204–207
11. Temperley, D.: *The Cognition of Basic Musical Structures*. The MIT Press, Cambridge Massachusetts (2001) 85–114