Jonathan Mooser
Review of Separating Voices in Polyphonic Music:A Contig Mapping Approach
By Elaine Chew and Xiaodan Wu
February 28, 2005

Chew and Wu's paper, *Separating Voices in Polyphonic Music:A Contig Mapping Approach*, describes a method for extracting individual voices from a piece of polyphonic music. It takes a piano roll representation, such as MIDI, as its input and outputs two or more subsets of the input, each of which represents an individual voice. The algorithm is based on a set of logical assumptions about how voices are likely to behave, and distills those assumptions into a few simple, concise steps. Results are analyzed for some 78 Bach fugues with explanations as to why the system failed in a few places.

Chew cites a few previously published works that enumerate sets of assumptions about voice separation. Those assumptions are then collective reduced to four basic rules. For example, we assume that within a given voice each successive note will be relatively close to the last. We also assume that voices tend not to cross. Together, these rules form the basis of the algorithm.

The author points out that, in contrast to the previous methods using stochastic or dynamic programming methods to find the best fit, this algorithm will do an exhaustive search over a large set of possible combinations. The resulting time complexity is $O(n^2)$, an important result that indicates that as larger inputs are introduced, the system will remain tractable.

The algorithm begins by breaking the input into a three level hierarchy. At the lowest level are individual notes. A series of notes known to be part of the same voice make a "fragment." Multiple fragments playing at the same time are then called a "contig." With a few specific exceptions, a boundary between contigs exists whenever the number of voices playing simultaneously changes. The result of this segmentation is a series of contigs from the start of the piece to the end. The bulk of the algorithm is dedicated to joining the fragments of neighboring contigs in accordance with the rules.

The rules for joining the fragments in neighboring contigs are straightforward. If two notes are part of the same longer note, they will definitely be joined. Other than that, the algorithm basically searches all possible pairings and finds the one that minimizes the sum of differences between joined notes. I'm not sure here if "difference" means difference in frequency or number of half-steps on the scale. Another point of confusion is a rule that excludes the joining of null notes without defining the term "null."

The resulting separation is tested on a large collection of Bach Fugues including the *Two-Part Inventions,* the *Three-Part Inventions*, and the *Well-Tempered Clavier*. As ground truth, the author cites the MuseData collection that provides both complete and individual voice representations of music. When the output of the program matches the individual voices in the Musedata database, we can say that the separation is correct. The analysis uses three very clearly defined statistical measures to objectively test the results. *Average fragment consistency* measures how many notes within a given fragment, on average, belong to the same voice. *Correct Fragment Connection* is an indication of how often fragments are correctly joined. Finally, *Average voice consistency*, is like *average fragment consistency*, but looks at the consistency over the length of an entire voice.

The results are impressive. The first two measures report close to 100% accuracy in all but a few cases. The results for *average voice consistency* are slightly worse, which can be explained by the fact that one error in the middle of a piece can make the whole rest of the voice wrong. Still, most of the results were above 90%. In general, most of the errors can be attributed to the fact that voices do, occasionally cross, violating one of the assumptions.

In the future, it may be useful to test the system on music written by different composers. If there is something unique to Bach's style that lends itself to this algorithm, the reported results may be overly optimistic. Overall, however, Chew makes a powerful case for its effectiveness. Moreover, it is explained thoroughly and clearly enough to be implemented and tested by others.