

Separating Voices in Polyphonic Music: A Contig Mapping Approach,
by E. Chew, X Wu

The algorithm presented in this paper uses basic principles of music perception to separate voices in polyphonic music using contig mapping. Three metrics are proposed to evaluate performance and accuracy of the automatically separated voices. These metrics are: the average fragment consistency (AFC), the correct fragment connection rate (CFC), and the average voice consistency (AVC). In this method synchronous notes are not allowed to be part of the same voice and contig mapping shows high fragment consistency, the grouping of notes from the same voice into the same fragments.

The authors use the knowledge of the perceptual principals of auditory streaming to create an $O(n^2)$ contig mapping algorithm for separating polyphonic pieces into their component voices. The algorithm only considers pitch height and event boundaries and ignores information on timbre and sound source.

The easiest method for voice separation, adopted by most sequencer software packages, is split voices according to some set of non-overlapping pitch ranges, but it can produce highly inaccurate and unsightly results. Thus, many researchers have tried to improve it.

The contig mapping approach, used in this method, is based on several underlying perceptual principles. One of the principles is the “pitch proximity principle”, which means that “the coherence of an auditory stream is maintained by close pitch proximity in successive tones within the stream”. The other perceptual principle is the “stream crossing principle”, which is related to the human’s difficulty in tracking streams of sounds that cross with respect to pitch. The rules and principles are translated into some assumptions implying that at certain segments of time (called maximal voice contigs), all voices sound synchronously in a well-behaved manner and also that intervals between the successive notes in a stream or voice are minimized. Based on the assumptions, distance minimizing procedure can be used to connect voices between segments.

The following terms are used in description of the algorithm: *note* (a musical entity with pitch & duration); *fragment* (a sequence of successive notes belonging to the same voice); *contig* (a collection of overlapping fragments that the overlap depth at any time is constant); and finally, *maximal voice contig* (a contig with the maximum number of voices present).

The algorithm is based on a *segmentation procedure*, in which the piece is segmented according to voice count that remains constant within the contig, and a *connection policy*, which is applied to maximal voice contigs after the segmentation. To connect voice fragments across the neighboring contigs, the distance minimizing choice is selected. First for each maximal voice contig, fragments in the immediate neighboring are connected to those in the contig, and then, the second order neighbors are connected to the immediate neighbors and so on. This is done iteratively and the maximum number of iterations and maximum number of contigs will be n when the number of notes is n , which will result in $O(n^2)$ algorithm. The maximal voice contigs are like seeds of the process that at first the contigs that are closer to seeds get connected. This policy for grouping works most of the time but not always.

The system is implemented in a Java application called VoSA (Voice Separation Analyzer) and it is platform-independent and only accepts MIDI files as input. To reduce certain errors, the

Engineering Approaches to Music Perception & Cognition

Baharak Zali
8679928115

Homework # 6
Feb. 24, 2005

Review #2

data is quantized using a selective snapping procedure and then the notes of a piece are sorted by their onset times. Since some of the pieces are embellished by an ending chord, the last three contigs are discarded and they are not considered in counting of maximum number of voices.

To evaluate their system, authors needed an absolute answer to compare with their results. Thus they used a separated version of their input piece in which each voice was stored in a separate track.

To test the system the authors used 15 Two-Part Inventions, 15 Three-Part Inventions and 48 Fugues by Bach and they used a quantization threshold of 30 ms for pre-processing the MIDI file before contig mapping. Then the authors evaluated the average fragment consistency (overall percentage consistency –when all notes in the fragment belong to the same voice- over all fragments), the correct fragment connection rate (proportion of connections that are correctly assigned), and the average voice consistency (average of how well the notes in the piece are assigned to the same voice) of the voice separation results. Their evaluation showed 99.75% AFC, CFC of 94.50%, and AVC of 88.98%. AVC is smaller than the other two because each incorrect connection can cause a severe loss of voice consistency. In general, if the average size of fragments is higher, it would cause higher average voice consistency numbers.

The overall statistics showed that the contig mapping approach is a very accurate solution for the voice separation problem and it is promising. One of the paths for future works can be testing the system on a larger polyphonic corpus and also to extend it to homophonic music.