## Voice Separation-A Local Optimisation Approach

Jurgen Kilian
Department of Computer Science
Darmstadt University of technology

Holger H.Hoos
Department of Computer Science
University of British Columbia

**Presented by Arun Chidambaram**

## Introduction

- Authors
- Voice Separation for Computer based Transcription of music
- Stochastic Local search method
- Differs by Allowing chords in individual voices
- Controlled by parameters
- Interactive optimization (real time)
- Goal-to create an algorithm to find a range of voice separations

## Existing Approaches

- Split Point Separation
- Rule Based Approaches

## Preliminaries

- Assumptions -list of notes sorted by time positions of their respective onset times
- Feature Vector $m_i=(a_i,d_i,p_i)$ ,$v_i$,$c_i$ ,offset($m_i$)
- Onset($m_p$)≤ Onset($m_q$)
  Onset($m_p$)= Onset($m_q$)
- Overlap($m_p$,$m_q$)
- Partition input note M into slices

## Algorithm

- Given i/p pieces M from locally optimized separations for slices of M
- local optimization based on C which assesses the quality of Si

- Given i/p piece M and max no of voices nVoices
- Segment M into slices $y_1, y_2 \ldots y_n$
- Cost Optimized voice separation for the first slice y1 is computed
- Iteratively extend this till $y_n$ when Complete voice separation for M is reached
- Problematic for unquantified i/p data (resolved by adjusting durations or onset times)

## Cost Function

Used for assessing and optimizing the quality of voice separation $S_i$ of a slice $y_i$ given previous slices is a weighted sum of terms that penalizes individual, undesirable features

$C(S_i,S) = P_{pitch}C_{pitch}(S_i,S) + P_{gap}C_{gap}(S_i,S) + P_{chord}C_{chord}(S_i) + P_{ovl}C_{ovl}(S_i,S)$

Where S denotes partial voice separation

$C_{pitch}$, $C_{gap}$ penalize large pitch intervals, gaps between successive notes in a voice

$C_{chord}$ penalizes chords with large pitch intervals between highest and lowest note ,as well as irregular notes containing notes with different onset times and durations

$C_{ovl}$ penalizes overlaps between successive notes in the same voice

## Penalty terms

- Pitch Distances Penalty $C_{pitch}$
- Gap Distance Penalty $C_{gap}$
- Chord Distance Penalty $C_{chord}$
- Overlap Distance Penalty $C_{ovl}$

## Cost Optimized Slice Separation

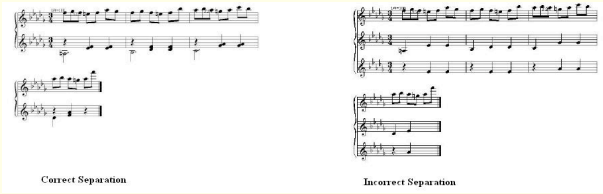
- Based on Cost function C and separation of slices $y_{1\ldots n}$ ,A stochastic local search approach for finding cost optimized voice separation $S_i$ for $y_i$
- Series of randomized iterative improvement steps starting with $S_i := S_i^o$ is performed during which each of which one note is assigned to reassigned to a different voice
- When Assigned with lower cost than the best assignment ,the current assignment and cost are memorized
- Terminated when no improvement can be achieved for a _ no of steps
- Initial separation $S_i^o$ for slice $S_i$ is obtained by assigning all notes of $y_i$ to the first voice. Notes with equal onset times are combined as chords or distributes all notes in $y_i$ into voices .Better results were obtained for the former case
- Two separations $S_i$ and $S_i^l$ are neighbors if both are valid separations of $y_i$ that differ in voice or chord assignment of exactly one note in $y_i$
- Selection of actual search step is performed randomly with probability ,neighboring selection with minimal cost is selected ,otherwise a neighbor is uniformly at random
- The stochastic local search in this case finds close to optimal separations

## Implementation

- Implemented in the Current version of midi2gmn
- MIDI file is given as i/p
- i/p data is quantized or unquantized
- Parameters set in Initialization file (fermata.ini)

## Results

- Tested on different kinds on music
- Highly dependant on Parameters




- Inaccurate results in some cases

## Conclusion

- Differs by detecting chords
- Different type of voice separations based on parameters
- Applied both quantized/unquantized input

Future Work

- Improved by fine tuning cost function
- GUI, Individual parameter setting

*Thank You*