Evaluation

Used corpora: BASS-dB*, IS, ISMIRgenre*, RWC*

Our own corpus, the Instrument Separation (IS) corpus contains:
- Binaural recordings
- Recordings with reference tracks
- Module (MOD) files, decomposed into individual tracks

Subjective evaluation criteria:
- Separation performance, $S$ (0-5)
- Remaining signal quality, $O$ (0-5)

Objective error measure: signal to noise ratio (SNR)

Baseline (BL), produces sum and difference channel (L+R, L-R)

Results

- HSBSS separation performance 4x higher than baseline
- Baseline quality score almost as high as HSBSS
- DTM and HSBSS almost equal in terms of dB SNR
- Best separation values per title have 4x higher SNR than the mean values
- HSBSS also has objectively 4x higher SNR performance than baseline

**Future Work**

- Implement missing tone clustering part
- Move concept to frequency domain, better decorrelation expected
- Redesign initialization procedure

**Histogram Based BSS**

- Use a time-shift disambiguation heuristic to minimize cluster spreading
- Detect number of clusters automatically
- Find new features (dimensions) for the histogram

**Iterative Template Matching**

- Find better non-sparseness cost function

**Direct Template Matching**

- Find template occurrences
- Needs to be at least sample-accurate
- Can be done using fast correlation
- Peaks in correlation result are filtered
- To have values above some threshold
- To be further apart than some minimum interval
- To occur in both audio channels at approximately the same time

**Resynthesis**

- Render each template at its offsets
- Not implemented now: templates belonging to the same instrument have to be grouped together before rendering

**Onset Detection**

- Find onset time to separate tone from concurrent playing instruments

**Steering Vector**

- Let onsets happen at each time sample
- Onsets become weights, therefore onset vector becomes steering vector
- Good solutions contain only few non-zero weights

**Algorithm**

- Iterate over synthesizing and adaptation steps until convergence
- Synthesizing
  - Convolve steering vectors with templates
- Adaptation
  - Minimize synthesizing error by adapting steering vector and template
  - Use resilient backpropagation (RPROP)
  - Use additional non-sparseness error function for steering vector to encourage sparse solutions

**Refinements**

- Calculate templates exactly, use RPROP only for steering vector
- Work on upsampled input for less high frequency damping

---

**Goals and Applications**

Main Goal: Given audio data containing music, separate all playing Instruments

Sub Goals

- Template Based Approach
  - Find all instrument onsets
  - Find all tones
  - Synthesize each instrument into an audio file, using its onsets and tones
- Feature Histogram Based Approach
  - Find all clusters in the feature histogram
  - Classify each frequency in the Input according to its cluster
  - Synthesize the frequencies of each class into a separate audio file

Applications: remixing, editing, denoising, automatic transcription

---

**Concept**

The C++ implementation of the three algorithms was created along with this thesis as a collection of programs collectively called INEX for Instrument Extraction.