Bach10 Dataset

----A Versatile Polyphonic Music Dataset

Zhiyao Duan and Bryan Pardo

zhiyaoduan00@gmail.com, pardo@northwestern.edu

Interactive Audio Lab
EECS Department
Northwestern University
Evanston, IL 60208, USA

1. Introduction

This is a polyphonic music dataset which can be used for versatile research problems, such as Multi-pitch Estimation and Tracking, Audio-score Alignment, Source Separation, etc. This dataset consists of the audio recordings of each part and the ensemble of ten pieces of four-part J.S. Bach chorales, as well as their MIDI scores, the ground-truth alignment between the audio and the score, the ground-truth pitch values of each part and the ground-truth notes of each piece. The audio recordings of the four parts (Soprano, Alto, Tenor and Bass) of each piece are performed by violin, clarinet, saxophone and bassoon, respectively.

2. Download

This dataset can be freely downloaded at http:// www.cs.northwestern.edu/~zdu459/Bach10. Before downloading, the user will be asked to fill a simple form to help us keep track of the usage of this dataset.

3. How to Cite

If you use the dataset in a work of your own that you wish to publish, please cite one of the following papers:

- Multi-pitch Estimation & Tracking

a) Zhiyao Duan, Bryan Pardo and Changshui Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio Speech Language Process.*, vol. 18, no. 8, pp. 2121-2133, 2010.

- Audio-score Alignment, Source Separation

b) Zhiyao Duan and Bryan Pardo, "Soundprism: an online system for score-informed source separation of music audio," *IEEE Journal of Selected Topics in Signal Process.*, vol. 5, no. 6, pp. 1205-1215, 2011.

4. Content

- Bach10_Dataset_Description.pdf: this document itself
- Each folder corresponds to one piece.
    - *.mid: the MIDI score of the piece, downloaded from Internet
    - *.wav: the audio of each individual part and the ensemble of the piece
    - GTF0s_*.mat: ground-truth pitch values (in MIDI number) of each part of the audio recording in each time frame. This is a matrix. Each row corresponds to a part (From top to bottom: Violin, Clarinet, Saxophone and Bassoon). Each column corresponds to a frame.
    - GTNotes_*.mat: ground-truth notes (pitch value is in MIDI number) of each part of the audio recording. This is a cell array. Each cell is again a cell array, which stores the notes of a part (From top to bottom: Violin, Clarinet, Saxophone and Bassoon). Inside the cell array of a part, each cell corresponds to a note, which is represented as a matrix of two rows. The first row is time (in frame number) and the second row is pitch (in MIDI number). Time is contiguous in each note. Onset and offset of this note correspond to the first column and the last column, respectively.
    - *.txt: the modified MIREX format of the ground-truth alignment between audio and MIDI. It is a data array of four columns. Each row corresponds to a MIDI note. From left to right, columns correspond to
        - onset time in audio (in ms)
        - onset time in MIDI (in ms)
        - pitch in MIDI (in MIDI number)
        - channel number of this note (violin=1, clarinet=2, saxophone=3, bassoon=4)

        Note: the original MIREX format only has the first three columns.
    - *.asl: another format of the ground-truth alignment between audio and MIDI. It is a data array of four columns. Each row corresponds to an audio frame. From left to right, columns correspond to
        - audio frame index
        - audio frame center time (in ms)

    - score beat number (in beat)
    - score time (in ms)

- Code: the folder of accompanying code
    - plotNote.m: a function to plot the notes of a part, e.g.
      ```
      load 'GTNotes_01-AchGottundHerr.mat';
      plotNote(GTNote, 1);
      ```
    - genMorePoly.m: a script to extend this dataset to have more polyphonic pieces. It creates all combinations of different parts of each quartet, so that solos, duets and trios are generated.
    - ErrorRate_FrameLevel.m: a function to evaluate frame-level multi-pitch estimation results. It associates with *-GTF0s.mat.
    - ErrorRate_NoteLevel.m: a function to evaluate note-level multi-pitch estimation & tracking results. I associates with *-GTNotes.mat.
    - EvalScoFo_ASL.m: a function to evaluate audio-score alignment results. It associates with *.asl.

5. Data Generation

The MIDI files of the ten pieces were downloaded from the Internet.

For the audio recordings, the four parts (Soprano, Alto, Tenor and Bass) of each piece were performed by violin, clarinet, saxophone and bassoon, respectively. Each musician's part was recorded in isolation while the musician listened to the recordings of others through a headphone.

The ground-truth pitch values and ground-truth notes of the audio recording of each piece were first generated for each individual part, and then combined with other parts. The audio of the mixture and each individual part were segmented into frames of 46ms length and 10ms hop. The first window was centered at 23ms from the beginning. For each frame of the mixture signal, if its RMS value was less than 0.075, then the frame was thought of unvoiced and no ground-pitch of any individual part was detected. Otherwise, a robust single pitch detection algorithm called YIN [1] was performed in the frame of each individual part to detect the ground-truth pitch value of that part in that frame. Slight manual corrections were held afterwards to fix some apparent errors. Ground-truth notes were then formed by connecting ground-truth pitches in adjacent frames manually.

The ground-truth alignment between audio and MIDI were obtained through human annotation. We built software to record and modify human tapped beats. A musician tapped beats using a computer keyboard when listening to the audio file. In this way, we obtained a ground-truth alignment between audio beat times and MIDI beat times. Then, in generating the *.txt alignment file, we linearly interpolate from the beat time alignment for each note in the MIDI

file. In generating the *.asl alignment file, we linearly interpolate from the beat time alignment for each audio frame.

6. Dataset Extension

Besides the original quartets, one can generate more audio recordings of different polyphonies, by exploring the combinations of different parts of each piece. For each piece, the maximum number of audio recordings that can be generated is 15, containing four monophonic parts, six duets, four trios and one quartet. Although the temporal dynamics of these new recordings are the same as the original one, they can be used to test algorithms in different polyphonies and instrumentations.

One can use genMorePoly.m to do this extension. This script will generate the audio recordings, MIDI files, as well as all ground-truth files corresponding to each combination of the parts.

7. Example Usage

- Multi-pitch Estimation & Tracking
    - Input: audio recording of the mixture
    - Ground-truth files: *-GTF0s.mat and *-GTNotes.mat
    - Evaluation code: ErrorRate_FrameLevel.m and ErrorRate_NoteLevel.m
- Audio-Score Alignment
    - Input: audio recording and MIDI of the mixture
    - Ground-truth files: *.txt and *.asl
    - Evaluation code: EvalScoFo_ASL.m for the *.asl ground-truth file. Refer to [2] for the *.txt ground-truth file.
- Source Separation
    - Input: audio recording of the mixture
    - Ground-truth files: audio recording of each part
    - Evaluation code: BSS_EVAL [3]
- Score-informed Source Separation
    - Input: audio recording and MIDI of the mixture
    - Ground-truth files: audio recording of each part
    - Evaluation code: BSS_EVAL [3]

8. Reference

[1] A. de Cheveigné and H. Kawahara, “Yin, a fundamental frequency estimator for speech and music,” *J. Acoust. Soc. Amer.*, vol. 111, pp. 1917–1930, 2002.

[2] A. Cont, D. Schwarz, N. Schnell, and C. Raphael, “Evaluation of realtime audio-to-score alignment,” in Proc. Int. Conf. Music Inf. Retrieval (ISMIR), 2007, pp. 315–316.

[3] E. Vincent, R. Gribonval, and C. Févotte, “Performance measurement in blind audio source separation,” IEEE Trans. Audio, Speech, Lang. Process., vol. 14, no. 4, pp. 1462–1469, Jul. 2006.