Automatic Lyrics Display System for Live Music Performances

Institution: University of Rochester

Authors: Karan Vombatkere *(kvombatk@u.rochester.edu)*

Bochen Li (bli23@ur.rochester.edu)

Research Mentor: Prof. Zhiyao Duan *(zhiyao.duan@rochester.edu)*

June – July 2016

Submission in response to

The IEEE Signal Processing Magazine

Call for Student Project Information

Abstract

Live musical performances (e.g., choruses, concerts, and operas) often require the display of lyrics for the convenience of the audience. This requires following the performance and controlling the lyrics display in real time. In practice, this is usually controlled by a staff member of the concert who has been very familiar with the performance. In this paper, we present our effort in implementing a computational system to automate this real-time lyric display process using music following techniques. The idea is to use a real-time Dynamic Time Warping (DTW) algorithm to align the musical performance with a reference recording (e.g., a rehearsal recording or a cover song for the same piece) in real time, and display the lyrics that have been pre-aligned with the reference recording. In this paper, we present a Java implementation of this system on a PC using a multithread audio recording and processing framework and a graphical user interface. Preliminary results show that the system can successfully deal with different performance speed and display lyrics appropriately.

Introduction

The primary objective of this project was to design and implement a computational system that can follow live music performances (e.g., choruses) in real time [1] and display pre-encoded lyrics for the audience. Our basic idea was to use a pre-recorded performance of the song as a reference for the alignment between the live performance and the lyrics. The reference recording could be rehearsals of the performer or a recording from other performers. The alignment between the reference recording and the lyrics has been manually encoded before the concert. During the concert, the system listens to the live music performance, and aligns it to the reference recording in real time. Figure 1 conceptually illustrates the system framework. Although manual work is still required in this system to align the reference recording with the lyrics prior to the concert, this manual work can be performed in a much more flexible fashion: it can be performed any time before the concert; it allows mistakes and revisions; it can be performed by performers themselves instead of specially hired staff members. In addition, once the alignment is done, it can be used for any performance of the same song, and by different performers.

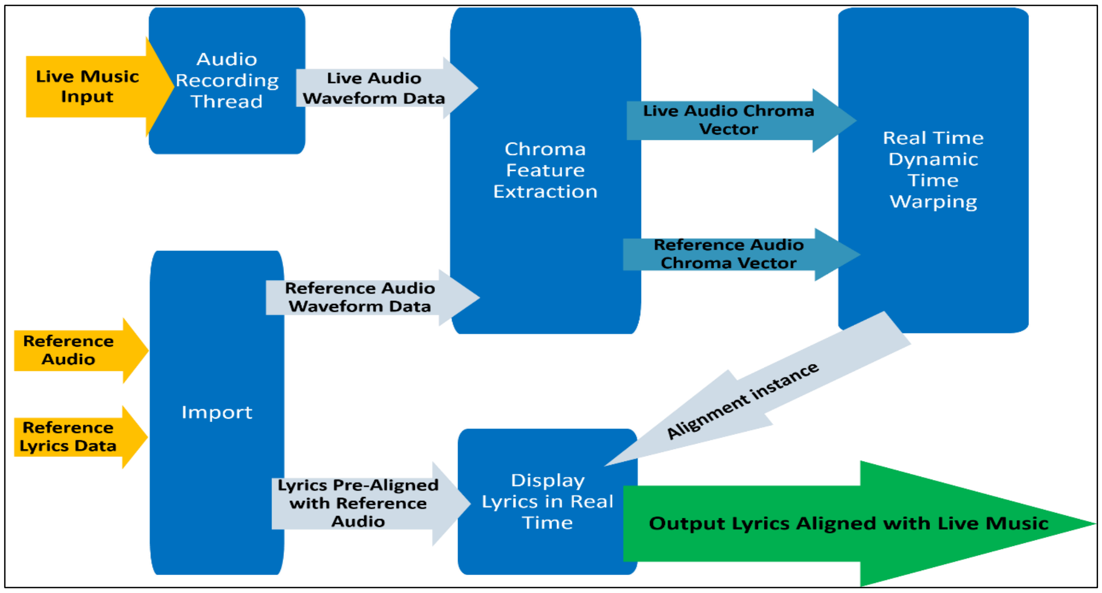

Live Music Input
Audio Recording Thread
Live Audio Waveform Data
Chroma Feature Extraction
Live Audio Chroma Vector
Real Time Dynamic Time Warping
Reference Audio Chroma Vector
Reference Audio Waveform Data
Reference Audio
Reference Lyrics Data
Import
Alignment instance
Lyrics Pre-Aligned with Reference Audio
Display Lyrics in Real Time
Output Lyrics Aligned with Live Music

Figure 1: Lyrics Display System Flowchart

Note that the reference recording and the live performance are not exactly the same, considering temporal dynamics, instrument arrangements, or environment noise. So audio fingerprinting technology [2] cannot be used here. However, the basic harmonic progression of the two recordings are similar. Therefore, we use the chroma feature to represent the audio data, which describes the relative energy of the audio signal at frequencies of the 12 pitch classes (harmonic content) and eliminates other factors such as timbre and intensity. Then the alignment is achieved through online Dynamic Time Warping (DTW). DTW can retrieve the optimal alignment between two pre-acquired temporal sequences by calculating the pair-wise distance in the feature domain. Different from the traditional one, the online DTW is able to output the alignment when one sequence is fully acquired (reference audio) while the other keeps coming (live recording) [3]. Then the system is able to align the lyrics with the live performance and display it at the right time. More specifically, the project contains three main tasks:

● **Real-time audio recording.** This is implemented in Java multi-threading, and the recording module runs in a separate thread, in parallel with the main algorithm thread.

● **Algorithm implementation.** This is the core algorithm to align the live audio stream with the reference audio using online DTW technique.

● **Graphic User Interface (GUI).** This is the main interface that the user interacts with and is used to display the lyrics.

Development Process and Implementation

While the actual development process occurred in a parallel manner, it is convenient to illustrate the system functionality in this 3 stage implementation.

1. Multithread framework for Audio Recording and Processing:

In order to successfully record and process audio data in real-time (without noise and loss of data) it was necessary to implement a Multithread framework in Java. This was done by initializing two separate threads for Audio Recording and Audio Processing functions, which run simultaneously.

A shared audio buffer (4096 bytes) is synchronously accessed by both threads and was configured so as to enable the recording thread to write audio data into the buffer, while the processing thread reads the data from the buffer. Note that this process occurs synchronously in a frame-by-frame manner so as to stream audio in real-time.

The Java Sound API was used to program the necessary functionality of the threads as well as to set up the audio format and buffer. A sampling rate of 44.1 kHz was used and the audio sampling was configured for single-channel audio data. Figure 2 below shows a schematic of the multithread framework and the manner in which the shared byte buffer is accessed by the two different threads.

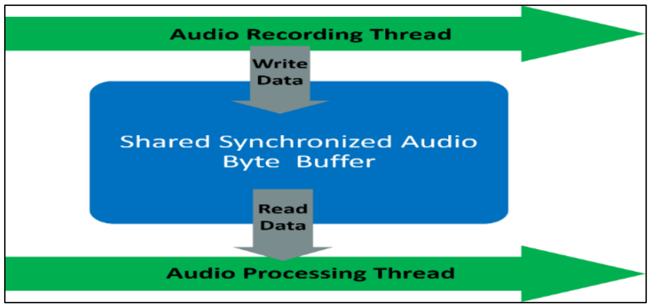


Figure 2: Multithread Framework Configuration

Once the recording and processing framework was set up, the next stage of the development process was to build the algorithmic structure to process the data. This implementation is detailed in the next section.

2. Real Time Algorithm Implementation:

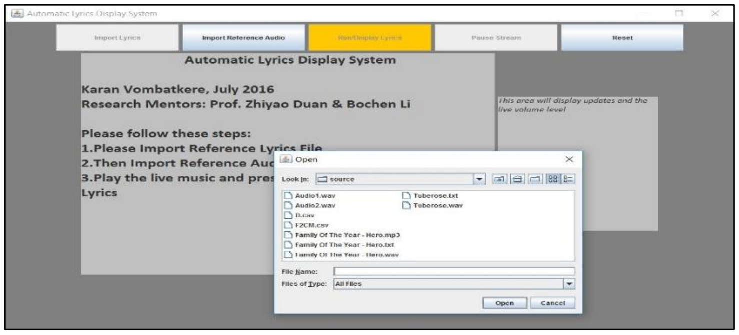
The first step in processing the audio data was to calculate chroma vectors for each frame (~50ms) of data so as to extract the pitch data. A chroma vector, which uses 12 bins to represent the relative energy of the audio in the

12 semitones of a musical octave, represents the harmonic content of the audio. The chroma vector is pre-calculated for reference audio data, since this is used as an input into the DTW algorithm.

The Dynamic Time Warping algorithm measures the similarities between two temporal sequences by calculating the least "distance" path of alignment between the two sequences. For our real time implementation, the algorithm uses two chroma vectors, corresponding to the live and reference audio, as the time series inputs and finds the best alignment for each frame. Note that while the algorithm has access to the entire reference audio chromagram, it can only access the live audio chroma vector a frame at a time, as they are recorded in real-time. The alignment is automatically synced with a manually notated instance for the onset of each lyric sentence.

3. Graphic User Interface (GUI) Design:

The GUI design consists of a simple window with buttons for basic functionality of importing the reference lyrics and audio data as well as to play and pause the stream. There is a text region created to display the lyrics in real time and to display updates and the live volume of the data being captured. The Java Swing Graphics API was used to design this graphic interface.



Automatic Lyrics Display System

Karan Vombatkere, July 2016
Research Mentors: Prof. Zhiyao Duan & Bochen Li

Please follow these steps:
1.Please Import Reference Lyrics File
2.Then Import Reference Auc
3.Play the live music and pres
Lyrics

This area will display updates and the live volume level

Figure 3: Graphic User Interface

Conclusions and Future Work

This automatic lyric display system was implemented successfully and tested to work correctly with several songs and lyrics. Some of the possible steps that could be taken to further improve and refine the performance of the system are highlighted below:

- There is scope for improvement and sophistication of the Graphic User Interface.
- The real-time Dynamic Time Warping algorithm can be further optimized and refined.
- The code could be packaged as a stand-alone software, which can be executed by any user.
- The system must be tested and refined for use during live concerts

References

[1] Bochen Li, Zhiyao Duan. Score Following for Piano Performances with Sustain-pedal Effects, International Society for Music Information Retrieval, 2015.

[2] Cano Pedro, Batlle Eloi, Kalker Ton, Haitsma Jaap. A review of audio fingerprinting. Journal of VLSI signal processing systems for signal, image and video technology, 2005, 41(3): 271-284.

[3] Simon Dixon. Live Tracking of Musical Performance Using Online Time Warping. International Conference on Digital Audio Effects, 2005.