Development and implementation of a scoring algorithm for a karaoke game

Master Thesis Proposal

Dongping Qiu
Supervisor: Zetterberg, Per
Examiner: Händel, Peter

School of Electrical Engineering, KTH
Stockholm, SWEDEN
2011-02-22
Last revised: 2011-03-14
Contents

1 Introduction .................................................. 3
   1.1 Background ............................................ 3
   1.2 Motivation ............................................. 3
   1.3 Problems in general .................................... 3
   1.4 Previous work ......................................... 3

2 Problem Definition ..................................... 4

3 Research Approach .................................... 4

4 Expected Results ....................................... 6

5 Document .................................................. 6

6 Time Plan .................................................. 7
1 Introduction

1.1 Background

Karaoke-based music game is a console game where player sings along with on-screen guidance and musical accompanies and finally receives a grade based on their pitch and timing. Pitch, also called fundamental frequency, determines the height of sound. In definition, pitch of periodic or nearly periodic signals is the inverse of period. In the game, console collects voice via microphone and synchronizes with reference melody. Sometimes players wear headphone which separates music from voice. The player sings in time and in tune with the lyrics as they roll at the bottom of the screen. Some games display the evolution of pitch as thick horizontal lines on a musical staff to show players the notes they produce. The game scores the performance based on how precise the song was performed and feeds a grade back.

1.2 Motivation

This thesis which is provided by [1] is supposed to create a karaoke-like game for the web. It is aimed to develop a tool for the automatic evaluation of sung voice. A monophonic melody as reference is stored in MIDI file, i.e. musical instrument digital interface. MIDI provides a common file format to store fundamental song units including sequence of notes and tempos. In musical notations, notes represent the pitch and relative duration of a sound. Another reason to choose MIDI is that it saves storage space compared with waveform formats. The idea of the project is to design a music transcription system of single-voice melodies. Such system accepts acoustic input and converts it into standard music notations. To transcribe melody automatically, notes must be recognized while the tempo sometimes also need to be detected. To score players performance, a quantitative evaluation is employed to calculate the difference between reference and transcribed notes.

1.3 Problems in general

Transcription of sing-voice melodies comprises two main steps. First, a track of pitch estimate is extracted from an audio waveform. The solutions are also called pitch determination algorithms (PDA). It is largely a matured subject while the study has a history of 40 years or even more. Second, a reliable conversion from the pitch track to musical notations is yet difficult. Voice is probably the instrument whose output least resembles a sequence of notes. For most musical instrument, the first step of conversion is to detect the onset, i.e. beginning of the note. Algorithms performs reasonably well with musical instruments but fail for sung voice. In the case of vocal sounds, the pitch changes rapidly and constantly. The onset can have an instantaneous pitch several tones away from the note eventually stablized upon. Even during the ”steady-state” of a note, vibrato can cause an oscillatory effect of pitch.

1.4 Previous work

A early work to transcribe sung-voice was proposed by [9]. After the note-onset was detected, note’s pitch was reported using a 300 milliseconds duration of steady-state where the distance between maxmum and minimum pitch was supposed to be within four half-tones. The note’s pitch was the middle value of pitch range. The note’s end or off-set is detected either by a falling off of amplitude below a threshold, or by the onset of another note. [8] set a threshold to categorize pitch errors into gross and fine errors. Only ”Stable pitch” inside a note boundary were kept to exclude gross errors while averaging remaining pitch over a given window to eliminate fine errors. The corresponding frequency to the bin with
The highest probability of a histogram is identified as the note of a frame. [5] proposed a stochastic approach using Hidden Markov Models (HMM). Three models for voiced, voiceless and silent frame are built upon six acoustic features, i.e., energy, delta energy, zero crossing, delta fundamental frequency, delta frequency energy, fundamental frequency and fundamental frequency error. The probability distribution of feature vectors was supposed as Gaussian Mixture Models (GMM). The proposal in [11] consisted of three parallel models: a pitch-trajectory model, a musicological model, and a duration model. The first detected the note via HMM. The second improved the transcription accuracy via estimate of key signature. The third adjusted the duration of note according to tempo.

In the past few years, several karaoke video games have been popular, like [2, 3, 4]. Most of them use pitch and rhythm to describe players’ voice while some apply sophisticated post-processing technologies. Also, they posed internet features where players can share and compete with other remote players.

2 Problem Definition

1. Eliminate the background music when microphone input is not pure vocal.
2. Select suitable algorithm to measure pitch.
3. Select musical features to describe acoustic characters of sung voice.
4. Build HMM note model to transcribe pitch estimate to note sequence.
5. Comparison between transcribed notes and MIDI reference.
6. Implement a friendly Graphic User Interface (GUI).

3 Research Approach

![Diagram](image)

**Figure 1**: Overview of the performance analysis system.

1. **Audio input**
   The sung voice of players is collected via microphone of console and converted to digital samples.
Usually the sampling frequency (Fs) is 44.1 KHz which produces a large amount of data. It is beneficial to down-sample original data by a factor of 4 (Fs = 11025 Hz) which does not generate aliasing effect.

II. Accompaniment
The audio waveform might be a mixture of voice, musical accompaniment and environmental noise. In this case it is necessary to remove the musical accompaniment. Recursive Least Square (RLS) algorithm is an efficient and noise insensitive method with low computational complexity.

III. Short frames
The sung voice is divided into short segments whose typical duration ranges between 20 and 50ms. The value is chosen because a sing frame require several pitch period. Adjacent frames overlap 50 percent which owns a smooth effect.

IV. Voice/ Unvoiced
This process is to detect the presence of voiced, voiceless and silent frames. If the sound is excited by nearly periodic flow, the frame is voiced. If the sound is excited by noise-like turbulent flow, the frame is voiceless. Otherwise, the frame is silence. An unvoiced frame means that it is not voiced, i.e. it can be voiceless or silent. Pitch only exists in voiced frames. In other words, the unvoiced frames are not selected for pitch detection. There are two main categories of algorithm. One is simple threshold analyzers using a small number of parameters. The other type is incorporated into PDA.

V. Basic extractor
Several musical features are measured in this part where pitch is the primary one. The estimation of the pitch can be subdivided into three steps [7]: pre-processing, extraction of the rough estimate and post-processing for error correction and temporal smoothing of the pitch track. The aim of pre-processing is to suppress the formant structure or enhance the energy in the region of possible pitch candidates. The essence of PDA is to detect periodic pattern in the signal. PDAs are generally classified into two main categories: spectral domain and time domain pitch detection. Recently a YIN algorithm is proposed by [6] which is a modified auto-correlation method. The YIN algorithm is based on average magnitude-difference function and extends with several modifications to improve the pitch estimation accuracy. However, all pitch trackers share common problems, e.g. octave error, glitches. Octave error occur when pitch estimate is twice or one half of the true frequency. Glitch means sudden abnormal peaks of pitch contour. These problems are caused by insufficient algorithm performance, natural fluctuations of voice and even special singing skills. These errors are corrected or compensated during post processing. Besides pitch, other musical features are measured as parameters of note model which is stated as below.

VI. Note detection
In this step, pitch estimates are grouped into musical note events. A basic and simple method has two steps: 1) to segment the note events, i.e. detect the start position and duration, according to musical feature, e.g. amplitude or pitch variation, 2) smooth the pitch estimates inside a note event into an integer MIDI number. The drawback of this idea lies in its inaccuracy. The stochastic nature of scattering pitch estimates motivates the use of probabilistic methods. A note model based on the theory of HMM is applied which provides a dynamic and flexible interpretation of notes. Various features including energy, zero crossing, pitch, voicing, meter, accent, spectral coefficients and spectral flatness were applied in previous works. A vector of musical features is called observation. In most conditions, the observation is assigned a multi-dimensional mixture Gaussian distribution. The note model expresses the statistical behavior of musical features within a note. [10] proves that a tree-state note model, i.e. attack, sustain and release, with two
Gaussian components provides low error rate. To train the note model, an acoustic database is used to create large number of note events. Note alignment can be carried out through decoding. Viterbi algorithm, a technique widely used, finds the state-sequence that most likely generates the complete sequence of observations. However, this method is of global optimality which is not the choice of real-time alignment.

VII. Scoring
A simple evaluation criterion is defined by the ratio between the number of correctly transcribed frame and the number of voiced frames in reference. A frame is considered to be correctly transcribed if the transcribed note equals to the reference note in that frame.

4 Expected Results
As results the parameters and thresholds for voice determination are needed. To illustrate the efficiency of PDA, a collection of folk songs in MIDI format is used. Graph of comparison between pitch estimate and MIDI note are necessary. To test the performance of PDA, error rate is defined by the ratio between number of erroneous pitch estimate and number of frames assigned a note. As for the note alignment, musical feature sets of note model, number of note states and number of GMM components are needed. Comparison among different values is tabulated. Graph of comparison between transcribed note and reference is necessary. To evaluate the performance, an acoustic database of singing is used to train and test the note model.

5 Document
Thesis Plan Describe why the thesis has been initiated and how the thesis will be carried through.
Final Report Present background, theory, methods and results of thesis work.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>Motivation, problem statement, requirement, expected results, thesis outline</td>
</tr>
<tr>
<td>Background</td>
<td>Description of music terminology, previous researches, existing solutions and systems</td>
</tr>
<tr>
<td>Research Methods</td>
<td>Illustration, evaluation and comparison among various algorithms</td>
</tr>
<tr>
<td>Implementation</td>
<td>System architecture, programming environment and graphical interfaces</td>
</tr>
<tr>
<td>Results</td>
<td>Experiment setup, system evaluation and results</td>
</tr>
<tr>
<td>Conclusion</td>
<td>Summary of the thesis work, how to improve in the future work</td>
</tr>
</tbody>
</table>

Table 1: Outline of final report
6 Time Plan

<table>
<thead>
<tr>
<th>Work title</th>
<th>Time</th>
<th>Activities</th>
<th>Weeks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Literature research</td>
<td>2010, Nov. 15 - 19</td>
<td>Search article and existing systems.</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2010, Nov. 22 - 26</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jan. 10 - 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jan. 17 - 21</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jan. 24 - 28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thesis proposal</td>
<td>Jan. 31 - Feb. 4</td>
<td>Write thesis proposal.</td>
<td>1</td>
</tr>
<tr>
<td>Pitch detection</td>
<td>Feb. 14 - 18</td>
<td>Extract rough pitch estimate from a voice frame.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Evaluate the performance of PDA with collection of MIDI files.</td>
<td></td>
</tr>
<tr>
<td>Post-processing</td>
<td>Feb. 21 - 25</td>
<td>Compensate octave error, tuning, estimate glitches of pitch estimate.</td>
<td>1</td>
</tr>
<tr>
<td>Background music cancellation</td>
<td>Feb. 28 - 4</td>
<td>Synchronize and cancel musical accompaniment.</td>
<td>1</td>
</tr>
<tr>
<td>Off-line test</td>
<td>Mar. 7 - 11</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Note model</td>
<td>Mar. 14 - 18</td>
<td>Select musical feature set to build note model.</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Mar. 21 - 25</td>
<td>Train and test the model with acoustic database.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mar. 28 - 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Off-line test</td>
<td>Apr. 4 - 8</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Continue but unspecified

Table 2: Time plan
References

[1] Introduction to master thesis project.  


   In practice, 1:0.


   Signal processing for melody transcription.  

