**Play With Yourself - Simple Musical Accompaniment**

|  |  |  |
| --- | --- | --- |
| **Adar Guy** | **Colin Malloy** | **Hector Perez** |
| Author  adarguy@uvic.ca | Author  malloyc@uvic.ca | Author  hectorp@uvic.ca |

**ABSTRACT**

Play With Yourself is an accompaniment tool for musicians interested in making demo recordings or simply adding more texture to their solo music. The program allows a user to record a track with guitar or piano the program then generates a MIDI file based on extracted beat, tempo, and harmonic information. This MIDI file can then be used with virtual instruments in a DAW to easily create a basic harmonic and rhythmic accompaniment for the original audio file.

1. **INTRODUCTION**

The project to be developed is a music accompaniment system. The system is composed of two stages. The first is a detection stage that will extract tempo, beat and key information from the input audio file. The second is a generative stage. A musical accompaniment in MIDI format is built based on the extracted information acquired in the previous stage. The final goal is to create a reasonable rhythmic pattern and a simple harmonic accompaniment. The system will work take audio files as input and generate a corresponding MIDI file for the purpose of accompanying the original audio using virtual instruments in a digital audio workstation environment.

This software is intended to be used by average musicians in tandem with DAW software. This software will allow musicians to quickly and efficiently make basic demo tracks. The software generates a MIDI map to accompany the original audio input that can be used with virtual instruments in a DAW environment to augment the original music by creating rhythmic and harmonic accompaniment. The motivation was to create a useful tool for amateur musicians that has an immediate usefulness. Ideas also came from discussions in the Music Information Retrieval class that suggested extending existing projects that have not yet been implemented.

1. **Timeline**

Week 1: Feb 27 - Mar 5

* Test potential libraries for music information retrieval relating to the project. The team will learn how to use the libraries and determine their usefulness and application for the project.

Week 2 : Mar 6 - Mar 12

* Begin designing detection system. Specifically begin building pitch and key extraction tools.
* Begin testing these tools with sample data sets.

Week 3 : Mar 13 - Mar 19

* Continue working on the detection system.  
  Complete the pitch and key extraction tools.  
  Begin the beat and tempo extraction tools.
* Continue testing and fine-tuning tools with sample data sets.

Week 4 :Mar 20 - Mar 26

* Begin work on MIDI map generation that takes extracted tempo/beat information and creates MIDI rhythm accompaniment.

Week 5 : Mar 27 - Apr 2

* Continue work on MIDI map generation from tempo/beat information.
* Add harmonic accompaniment generation to MIDI map tools.

Week 6 : Apr 3- Apr 9

* Finish various components to create a basic working version.
* Until final due date improve and optimize various aspects of the program.

1. **Roles**

Responsibilities for our project include: Research, code implementation, testing the different stages of the project, and recording results from tests.

All of the team members will perform these activities, but more specific tasks that team members will complete will be :

* Adar Guy: Testing libraries and tools, assist in team set up. Maintenance of project framework (Shared resources, folders, GitHub repository). Code implementation.
* Colin Malloy: Research for rhythm generation. Recording results for the testing of the rhythm detection part of the detection stage. MIDI map generation.
* Hector Perez: Testing libraries and tools, assist in team set up. Research for better data sets to use, and alternative implementations for both detection and generation stages. Code implementation.

1. **Tools and Resources**
   1. **Libraries for Feature Extraction**

* Sound Analyse 0.1.1 - Nathan Whitehead  
  This is a Python Library for detecting pitch and loudness.
* Mir Eval 0.4 - Colin Raffel et al.  
  Mir Eval is a Python library designed for testing accuracy and correctness of MIR algorithms.
* Aubio  
  Aubio is a feature extraction library that can be used for both pitch and rhythm extraction. It is able to work in real time. Aubio is written in C, but it can be accessed from Python.
* Pymir - jsawruk  
  This is python library for music information retrieval.
* Librosa - Dan Ellis and other members at LabROSA.  
  This is a Python Library for Audio and Music analysis.
* VAMP plugins.  
  These are plugins for audio feature extraction. VamPy is a program that allows us to use VAMP in Python.
  1. **Sample Implementations.**
* Sonots - Naotoshi Seo.  
  Sonots does pitch detection in Matlab
* Frequency estimator - Endolith  
  Pitch detection Python.
* Python beat detector - Shunfu  
  Real time beat detection python implementation.
* Bpm detector - Scaperot  
  Bpm detector for a complete .wav file, based on G. Tzanetakis’ algorithm in Audio Analysis using the Discrete Wavelet Transform.
  1. **Related projects.**
* Beat This - Kileen Cheng  
  A beat synchronization project created at Rice University.

1. **Literature and Prior Work:**

From the literature, we were able to identify some options and previous solutions for key estimation and rhythm detection. Some articles are very specific in terms of the implementation. For key estimation, chroma vectors are used, as seen in class. A fourier transform with high resolution is taken of the signal and posteriorly the frequencies are ‘folded’ into a 12 pitch histogram.

For rhythm estimation, there are many different methods. Goto and Muraoka developed two different systems that work depending on whether the input signal is drumless or not. The drumless beat detection relies on probabilities of chord changes without actually detecting what the chord names are. However, this research was based on MIDI files. There was research done by the same authors in regards to audio signals containing drums. This system worked with acoustic audio signals, so the approach to finding the beats differed, and a complex multiple- hypothesis approach was used to determine the best candidate for the rhythm.

Other methods for beat detection use autocorrelation, binary trees or trellis trees (related to Hidden Markov Models). Timing nets (which are a form of neural network) are used in a self adjusting beat detector by R. Harper and Ed Jernigan. Nodes (which represent hypothesis of where the next beat is) that are more likely to be correct predictions for the beat are chosen as the current beat for the audio signal. The nodes take in ‘spike trains’ and try to find coincidences between their train and the one found in the audio signal.

An MIT research project is mentioned across the literature - it is called Machine Rhythm. This system performs polyphonic midi rhythm detection.

Another rhythm detection alternative came from E.D. Scheirer, and is performed on acoustic multi-timbral audio. This approach uses resonators to try to achieve phase-locking with the signal. The signal’s frequencies are divided into subbands and then phase-locking is attempted. After this happens, the results are combined and the tempo of the signal is extracted.

Peter Desain and Henkjan Honing used expectancy curves to generate an overall expectancy curve for the rhythm which displays spikes for onsets of beats. They repeated the process of creating an expectancy curve for equal time length sections of the audio and multiplied them by each other to generate a curve that is more representative of the average.

1. **REFERENCES**
2. A. Driesse: “Real-time Tempo Tracking Using Rules to Analyse Rhythmic Qualities” *Proceedings of the 1991 International Computer Music Conference, 578 - 581.*
3. A. Shenoy, and Y. Wang: “Key, Chord, and Rhythm Tracking of Popular Music Recordings” *MIT Press Journals, Computer Music Journal.*
4. A.T. Cemgil: “Polyphonic Pitch Identification and Bayesian Inference” *International Computer Music Conference.*
5. C. Harte, and M. Sandler: “Chromagram: Automatic chord identification using a quantised chromagram,” *In proceedings of AES 118th Convention, Barcelona, 2005*.
6. D. Rosenthal: “Machine Rhythm: Computer Emulation of Human Rhythm Perception” *MIT Media Lab, Ph.D Thesis Paper, (1992)*
7. D. Rosenthal, M. Goto, and Y. Muraoka: “Rhythm Tracking Using Multiple Hypotheses” *Proc. of the 1994 International Computer Music Conference, International Computer Music Association, San Francisco, 1994, 85 - 87.*
8. E. Scheirer: “Tempo and Beat Analysis of Acoustic Musical Signals” *Journal of the Acoustic Society of America, 588 - 601*
9. G. Loy: “Musicians Make a Standard: The MIDI Phenomenon” *Computer Music Journal. Vol. 9, No. 4 (Winter, 1985), 8 - 26, MIT Press.*
10. G. Peeters: “Chromagram: Automatic chord identification using a quantised chromagram,” *In proceedings of ISMIR conference, 2006, 115 - 120.*
11. K. Abdullah-Al-Mamun, F. Sarker, and G. Muhammad: “A High Resolution Pitch Detection Algorithm Based on AMDF and ACF” *University of Dallas Press, Citing Papers.*
12. M. Alghoniemy, and A.H. Tewfik: “Rhythm and Periodicity Detection in Polyphonic Music” *Institute of Electrical and Electronics Engineers (IEEE), MMSP 1999: 185 - 190.*
13. M. Davy, and S.J. Godsill in Bayesian: “Bayesian Harmonic Models for Musical Signal Analysis” *Bayesian Statistics 7, University of Montreal, 2003.*
14. M. Goto: “An Audio-based Real-time Beat Tracking System for Music With or Without Drum-Sounds” *Journal of New Music Research, Vol. 30, No. 2, July 2001*
15. M. Goto, and Y. Muraoka: “Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions” *Speech Communication 27(3-4): 311 - 335 (1999).*
16. N. Griffith: “Modelling the acquisition and representation of musical tonality as a function of pitch-use through self-organising artificial neural networks” *University of Exeter, UK, 1993.*
17. P.E. Allen, and R.B. Dannenberg: “Tracking Musical Beats in Real Time” *Proceedings of the 1990 International Computer Music Conference, 140-143.*
18. P. Cariani: “Neural Timing Nets” *Neural Network, 2001d, 14, 737 - 753*
19. P. Desain, and H. Honing: “Advanced Issues in Beat Induction Modeling: Syncopation, Tempo, and Timing” *Proceedings of the 1994 International Computer Music Conference, 92 - 94.*
20. R. Harper, and M. Jernigan: “Self-adjusting Beat Detection in Prediction in Music” *Institute of Electrical and Electronics Engineers (IEEE), ICASSP (4) 2004: 245 - 248.*
21. S.A. Raczynski, E. Vincent, and Shigeki Sagayama: “Advanced Issues in Beat Induction Modeling: Syncopation, Tempo, and Timing” *Proceedings of the 1994 International Computer Music Conference, 92 - 94.*