

Dancing Nao Robot

Milan Pogadl, , Marius Kaiser, , Khaliun Jargal, , and Jane Doe,

Abstract—How to analyse music to make a robot dance to it.

I. INTRODUCTION

THE goal of this project was to make a robot to anything the algrithm can identify as music. We used Essentia for Music Information Retrieval and a predefined set of dancing moves.

September 10, 2019

A. Essentia

Finding the right music analysis framework was a bit difficult, because there a lot of frameworks for this in Python. But we found a research paper[1] that compared the many different music analysis libraries and Essentia was the best library with Python bindings. Essentia is used in many diffenrent applications like: freesound.org, KORG, Fretello and Crypt of the NecroDancer [3].

To use Essentia you have to load the music with one of the different loaders and then you can apply one of the many algorithms of the library on it [2].

We used RhythmExtractor2013 for the Beat Detection, Intensity to get the mood of the music and Danceability to decide if the given sound file represents music.

B. Key Frames

The NAOqi library ALMotion provides methods which allow complex movements of the NAO. It contains commands allowing to manipulate joint stiffness, joint angles, and a higher level API allowing us to control the moves.

The following methods are important for the implementation of our movements angleInterpolation(names, angleLists, timeLists, isAbsolute) and getAngles(names, useSensors).

For basic postures, we implemented it using API from NAOqi. Next, angles of Nao joints are gained through getAngles function to create particular movement.

Creating working key frames for the dance moves of the robot proved to be more difficult, than we expected. Therefore, a complex choreography for the Nao has been built in this project using the timeline functions of the Choregraphe software. Choregraphe is a multi-platform desktop application which allows us to create our dance, test them at home and monitor Nao through programming. Every move had to be tested directly with the robot to check its stability so that the robot would not fall down while dancing. We had to optimize every move for a variable speed derived from the beats per minute of the music.

C. Dance Generation

First the sound file is loaded with an Essentia loader. Then we analyze the Track with the RhythmExtractor2013, Intensity and Danceability algorithms. We use the outcome of the Danceability algorithm to decide if the sound file represents danceable music. The result of the Intensity algorithm determines if the use of relaxed or more aggressive dancing movements. Then combined this moves get combined to form a choreography. The speed of the movements is adapted to the beats per minute of the song determined by RhythmExtractor2013 algorithm. RhythmExtractor2013 gives us also the position of each beat so that we can time some moves to a beat position. Then the music starts and the robot gets it movement commands to dance to the music.

II. CONCLUSION

It might be even possible with a faster Nao CPU to do the music analysis in real time.

APPENDIX A BEAT DETECTION

Appendix one text goes here.

ACKNOWLEDGMENT

The authors would like to thank the Universitat Pompeu Fabra for developing Essentia and making available for the public as an opensource library.

REFERENCES

- [1] David Moffat, David Ronan, Joshua D. Reiss, *EVALUATION OF AUDIO FEATURE EXTRACTION TOOLBOXES*, Center for Digital Music, Queen Mary University of London, 2015.
- [2] Dmitry Bogdanov, Nicolas Wack, Emilia Gomez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, Jose Zapata and Xavier Serra, *ESSENTIA: AN AUDIO ANALYSIS LIBRARY FOR MUSIC INFORMATION RETRIEVAL*, Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain, 2013.
- [3] Dmitry Bogdanov, *Music Feature Analysis with Essentia*, Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain, 2019.
- [4] Akshay Paranjape, *Tools for Feature Extraction - exploring and using essentia library*, 2017.