Dancing Nao Robot

Milan Pogadl, *232964*, Marius Kaiser, *337180*, Khaliun Jargal, *205745*, and Anderson Machado, *397026*

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

I. Introduction

THE goal of this project was to make a robot dance to any kind of music the algorithm 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 are 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 to it [2].

We used RhythmExtractor2013 for the Beat Detection, Intensity to get the mood of the music. We tried to use Danceability to decide if the given sound file represents music, but this algorithm has some problems with noise and in some cases speech was as danceable as certain songs. On the other hand a metronom song had a nearly perfect danceability.

B. Key Frames

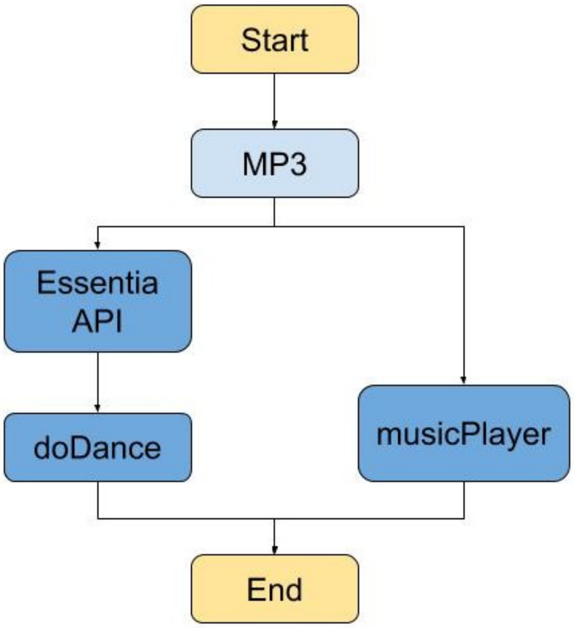
The NAOqi library ALMotion provides methods which allow complex movements of the NAO. It contains commands allowing to manipulate joint stiffness, joint angles, and a higher level API allowing us to control the moves.

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 the API from NAOqi. Next, angles of Nao joints are gained through getAngles function to create a 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 and test it at home with a virtual nao robot. 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. The result of the Intensity algorithm determines could be used to decide between relaxed or more aggresive dancing movements, but we didn't finish enough different dance moves in the end to use it. Then we decide which predefined set of dance moves is used. Some dance moves had to be combined with a stand move to properly initialize them. Now the music starts and the robot begins to dance to the chosen music file. The speed of the dance movements is adapted to the current beat duration of the song determined by RhythmExtractor2013 algorithm. RhythmExtractor2013 gives us also the position of each beat so that we can syncronize each dance move to the next beat position. The Nao v6 proved to be superior to former models for dancing.

D. Project Setup

Github Repository of the project
Make sure you are using Python 2.X, because the Nao Framework is based on it.
Install Essentia and pydub:

```
pip install essentia
pip install pydub
```

This works under Linux but under Windows you have to compile Essentia manualy. Make sure to disable autonomous life of the nao, otherwise the robot will fall down.
Then run in the motion dir of the repository:

```
python naorecord.py ——song /path/to/song/
```

II. Conclusion

At the beginning of our project, we wanted to do the music analysis in real-time but latency and slow reaction to tempo changes made this difficult.
The project could be further expanded with an ability to read new dance moves from .json files. This dance moves could be anotated with the genre, mood, intsity and beats per minute ranges for which they are designed. But designing such dance moves would of course take some time. With enough dance moves the robot could react to different genres of music through musical genre classification[5] and adjust his dance moves to the genre. The robots lights could be used for the expression of different moods. With that the robot could be used in disco as an obvious application of our program.

Acknowledgment

The authors would like to thank the Universitat Pompeu Fabra for developing Essentia and making it available to the public as an open source 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 Gulati1, Perfecto Herrera, Oscar Mayor, Gerard Roma1, Justin Salamon, Jose Zapata1and Xavier Serra1,
ESSENTIA: AN AUDIO ANALYSIS LIBRARY FOR MUSICINFORMATION 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.

[5] Boix Luna, Pau,
Automatic Movie Genre Classification Based On Musical Descriptors,
Universitat Pompeu Fabra, Barcelona, Spain, 2014.