26.05.2019

Project in Natural Language Processing
Analysis of midi files

Markus Niebisch
I6123084

Prior Work

In this project I was mainly inspired by the TEDx „Pop Music is Stuck on Repeat“ of Colin Morris. In this talk, he compares lyrics of different songs across the decades. He destinguishes their compressibility and derives from that, their complexity. Also, he creates similarity matrices of the lyrics within a song. In this project I want to do the same, but use the melodies instead. Additionally, I create new melodies by creating n-gram models out a set of melodies.

Data

As input I used all singles by the British band The Beatles. I included the Songs 'How far we've come' by Matchbox 20, „Pompeij“ by Bastille, 'Sandstorm' by Darude and 'Ruby' by Kaiser Chiefs as well. These have been chosen as The Beatles are famous for their influence on pop music in general. Sandstorm has been chosen, because of its simplicity. The other three songs, because I had recently to play them at the Battle of the Studies in Maastricht and therefore they are familiar to me.

For every song the lead channel has been extracted and stored into a new midi file.

Model

Complexity Measure Zip

As complexity measure the compression algorithm Zip using Winrar with compression settings 'best' is used. As Zip is still a very good compression even by modern standards this seems fitting to compare the complexity of different songs.

Visualisation Similarity Matrix

A similarity matrix compares every value of an array with every other value of an array. So every row stands for a different moment in time in a track and every column stands for a different time in a song. On the top left the first note is compared with the first note. On the top right the last note is compared with the first. This matrix is always symmetric. If we choose a black and white output an entry is colored black if the respective notes are the same. If we choose a coloured output, the hue is defined by the 11 half notes of an octave. Brightness is defined by the difference of the octave of the notes.

Melody generation

For the melody generation n-grams are created. One time for the pitch of the notes. One time for the length of the notes. Pauses are considered as notes with pitch 'negative 1'. These are converted into a cumulative distribution function, such that for every pattern an output value is generated.
Most important for the implementation is the fact, that if one creates a hashmap of lists in Java, it works for storing and retrieving pattern counts, as Lists of Integer compare to actual content. Not like Arrays of Integer, where the identity, i.e. memory location, of the array is compared.

Results and Analysis

Complexity

The songs differ greatly in their compressibility. Sometimes because one song is very redundant as Sandstorm in the one end. The infamous 'Strawbery Fields Forever' has the lowest compressibility. 'How far we've come' seems to be more complex than it is. This is due the midi file only representing the first verse and chorus. (see Fig 1,14)

Visualisation Similarity Matrix

It can be seen that for every song a different pattern appears. For deeper analysis the time is missing. Especially Sandstorm is notable for its many equally spaced black areas. (see Fig 2) In Sgt. Peppers one can clearly make out the different parts of the song. The first and the second verse appear the same, big breaks in between and the 'interlude' has a complete different pattern. (see Fig 8,9) Yesterday shows the classical AABABAB pattern. This is a pattern where the first verse is repeated twice before the chorus. (See Fig 10,11). In I Feel Fine, which shows the lead guitar the different melodies which are repeated throughout the song are clearly visible. (See Fig 4,5).

Melody generation

The melodies which are created often seem chaotic to a certain degree, but not completely off. This is due to the separation of the rhythm and the pitch during the generation. (See Fig 12,13) The implementation is reasonably fast. Laplace smoothing has been omitted due to memory constraints and limited time.

Future research

A music generator, where one paints into the similarity matrix, could be interesting. For this it should be noted, that actually the first line and column have already determined the whole song. Whatever change is done inside the image, it changes other parts of the image as well.

The visualisation right now technically only works with monophonic melodies. The program does still work, but the results are not fully reliable.

The melody generation needs a more complex model. This could be done by using normalised pitch values and by linking pitch with the length of a note. Also a song could be created by first creating a theme for verse, one for the chorus. These themes can be repeated in modified form like inversion. Also one could predetermine the key of the song and decide if the song should be in Major or in Minor (or one of the exotic modes).

References:

https://en.wikipedia.org/wiki/Key_(music)
https://en.wikipedia.org/wiki/Mode_(music)
https://www.csie.ntu.edu.tw/~r92092/ref/midi/
https://youtu.be/_tjFwcmHy5M
https://stackoverflow.com/questions/3850688/reading-midi-files-in-java
https://stackoverflow.com/questions/29480842/sample-without-replacement-in-java-with-probabilities
https://riptutorial.com/java/example/621/play-a-midi-file

Appendix:

Name	CompressionR
Darude_-_Sandstorm6bpm_120.mid	0,048355539
Beatles_I_Feel_Fine1bpm_120.mid	0,174331345
TheBeatles_Ticket to Ride0bpm_120.mid	0,2000496401
TheBeatles_Ticket to Ride1bpm_120.mid	0,2020352445
Help K4bpm_120.mid	0,208
Hey Jude K3bpm_120.mid	0,2213247173
We can work it out K0bpm_120.mid	0,2276390008
TheBeatles_A Hard Day's Night0bpm_120.mid	0,2289551261
Beatles_We_Can_Work_It_Out0bpm_120.mid	0,2324738114
Beatles_Let_It_Be3bpm_120.mid	0,2435163355
TheBeatles_Get Back0bpm_120.mid	0,245821727
Beatles_Come_Together2bpm_120.mid	0,2461267606
Beatles_She_Loves_You0bpm_120.mid	0,2465548232
Beatles_She_Loves_You1bpm_120.mid	0,2468544038
Beatles_Baby_Its_You1bpm_120.mid	0,2719019519
Beatles_Baby_Its_You3bpm_120.mid	0,2834080717
Beatles_Love_Me_Do0bpm_120.mid	0,2841623785
Beatles_Sgt_Pepper3bpm_120.mid	0,2867540029
Beatles_Aint_She_Sweet4bpm_120.mid	0,2939876216
Beatles_I_Want_To_Hold_Your_Hand4bpm_120.mid	0,2967153285
Beatles_Cant_Buy_Me_Love1bpm_120.mid	0,2995969548
Beatles_Eleanor_Rigby3bpm_120.mid	0,3146825397
Beatles_With_A_Little_Help_From_My_Friends3bpm_120.mid	0,3216136195
Beatles_From_Me_To_You3bpm_120.mid	0,3463268366
Beatles_Yesterday0bpm_120.mid	0,4044414536
Help K5bpm_120.mid	0,4168157424
Hello Goodbye K3bpm_120.mid	0,4197713021
Beatles_Love_Me_Do1bpm_120.mid	0,4307511737
R1bpm_120.mid	0,4721502591
Beatles_We_Can_Work_It_Out1bpm_120.mid	0,4758961681
Day tripper K3bpm_120.mid	0,4867724868
Pompeij-Bastille2bpm_120.mid	0,495
Beatles_Penny_Lane1bpm_120.mid	0,5350089767
Yesterday 2 k3bpm_120.mid	0,5371318822
Beatles_Please_Please_Me3bpm_120.mid	0,5668789809
Ruby15bpm_120.mid	0,5730786317
Free as a bird K2bpm_120.mid	0,6040447046
Paperback writer K4bpm_120.mid	0,6061814556
Help K6bpm_120.mid	0,6128590971
Pompeij-Bastille1bpm_120.mid	0,6279761905
Beatles_Yellow_Submarine1bpm_120.mid	0,6639379578
The Beatles - All You Need Is Love3bpm_120.mid	0,740755627
Beatles_Lady_Madonna3bpm_120.mid	0,7545189504
Beatles_Something4bpm_120.mid	0,7943548387
HowFarWeveCome3bpm_120.mid	0,7974683544
Beatles_Something3bpm_120.mid	0,8379287155
TheBeatles_Strawberry Fields Forever12bpm_120.mid	0,9687108886

Fig 1: **Compression Ratios Numericaly**

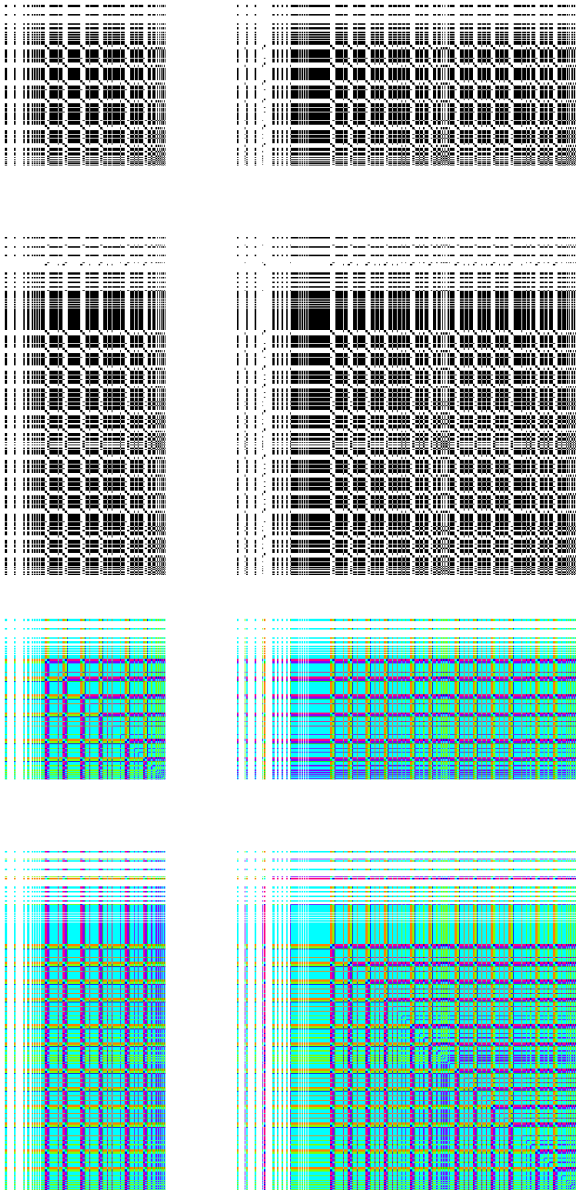
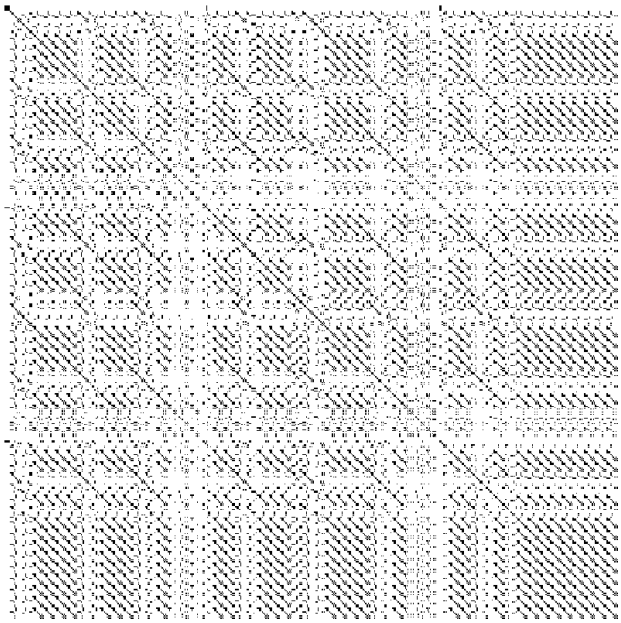


Fig 2,3:Darude



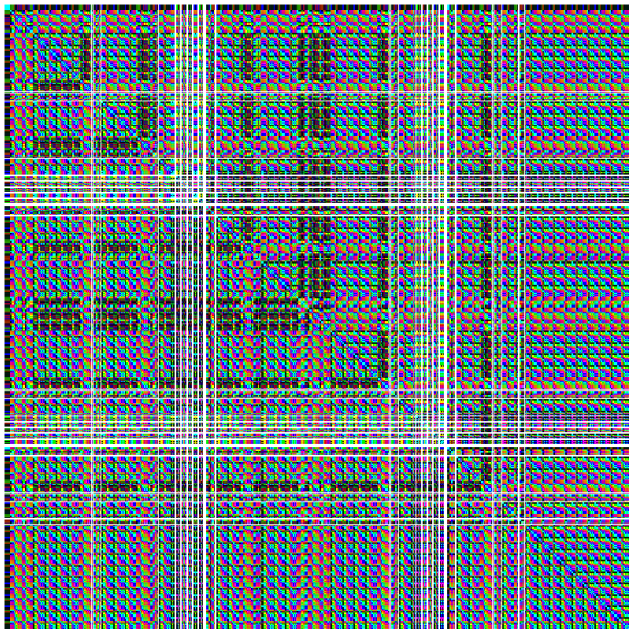


Fig 4,5:I Feel Fine (lead guitar)

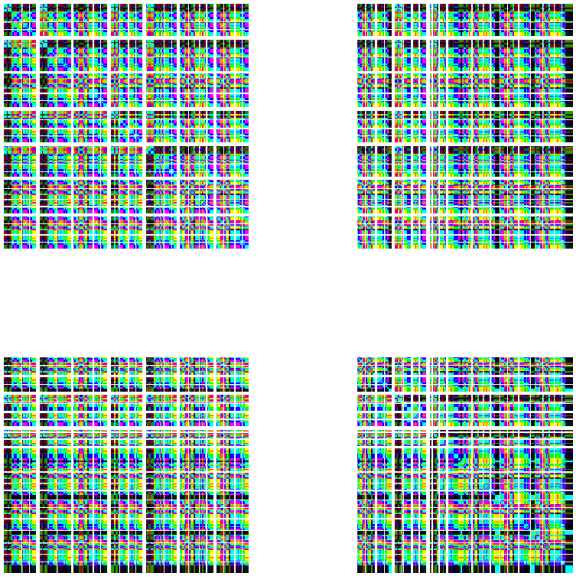
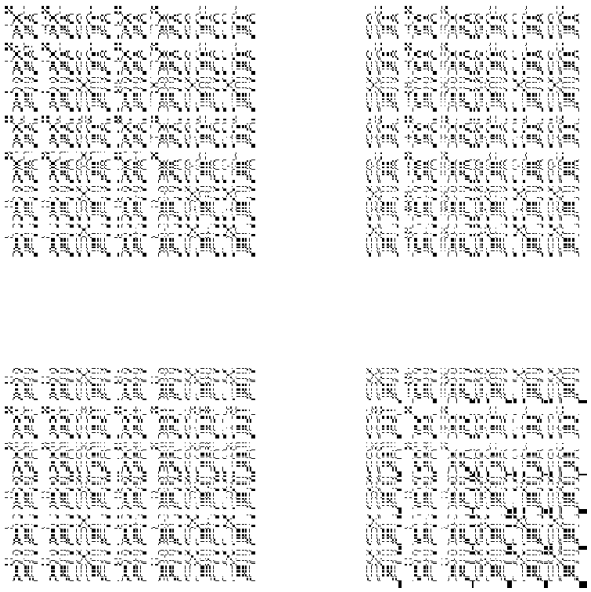


Fig 6,7:Let it be



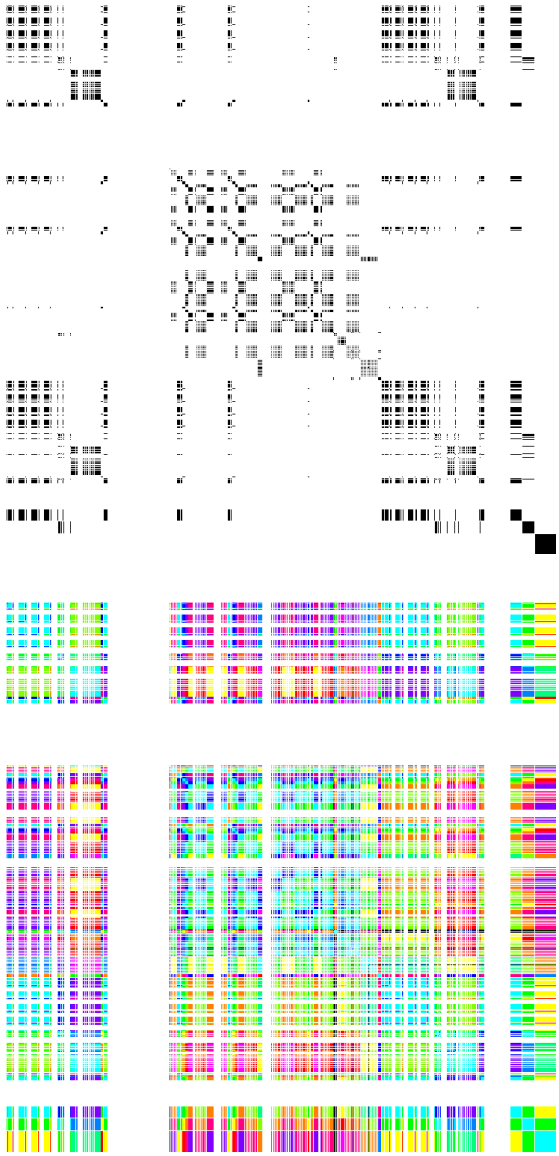
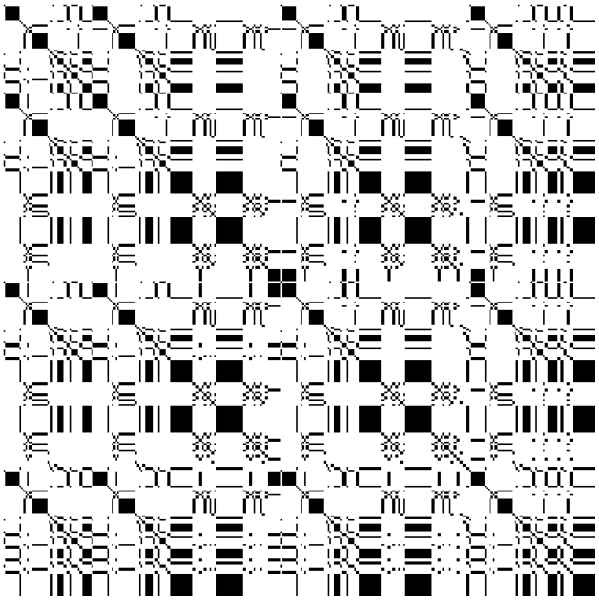


Fig 8,9: Sgt Peppers



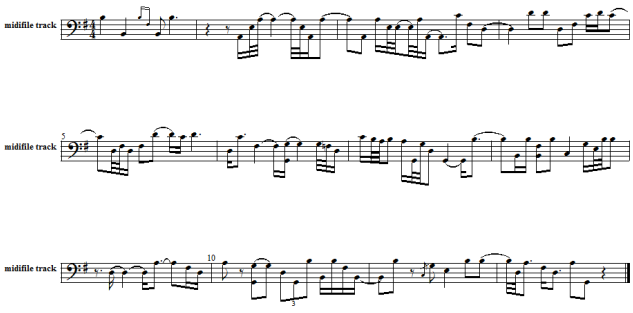
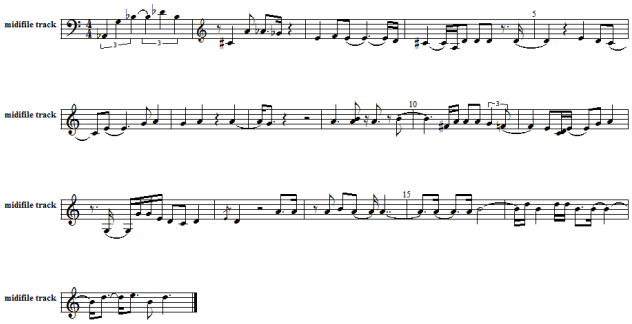


Fig 12: 9 Pattern.mid



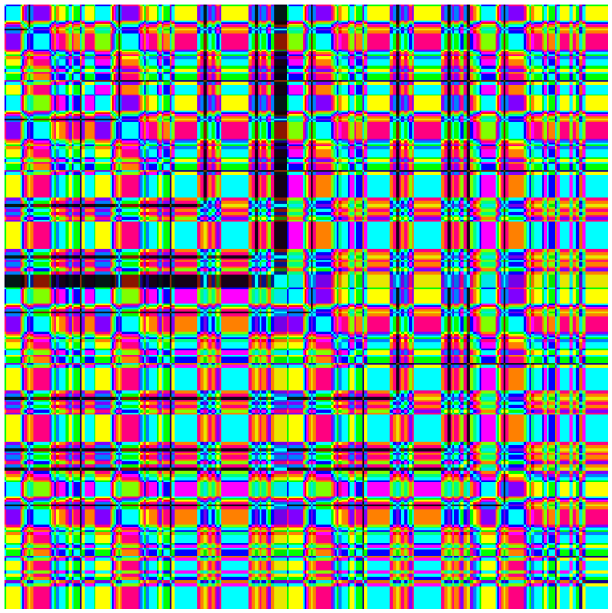


Fig 13: 10 Pattern.mid

Fig 10,11: Yesterday

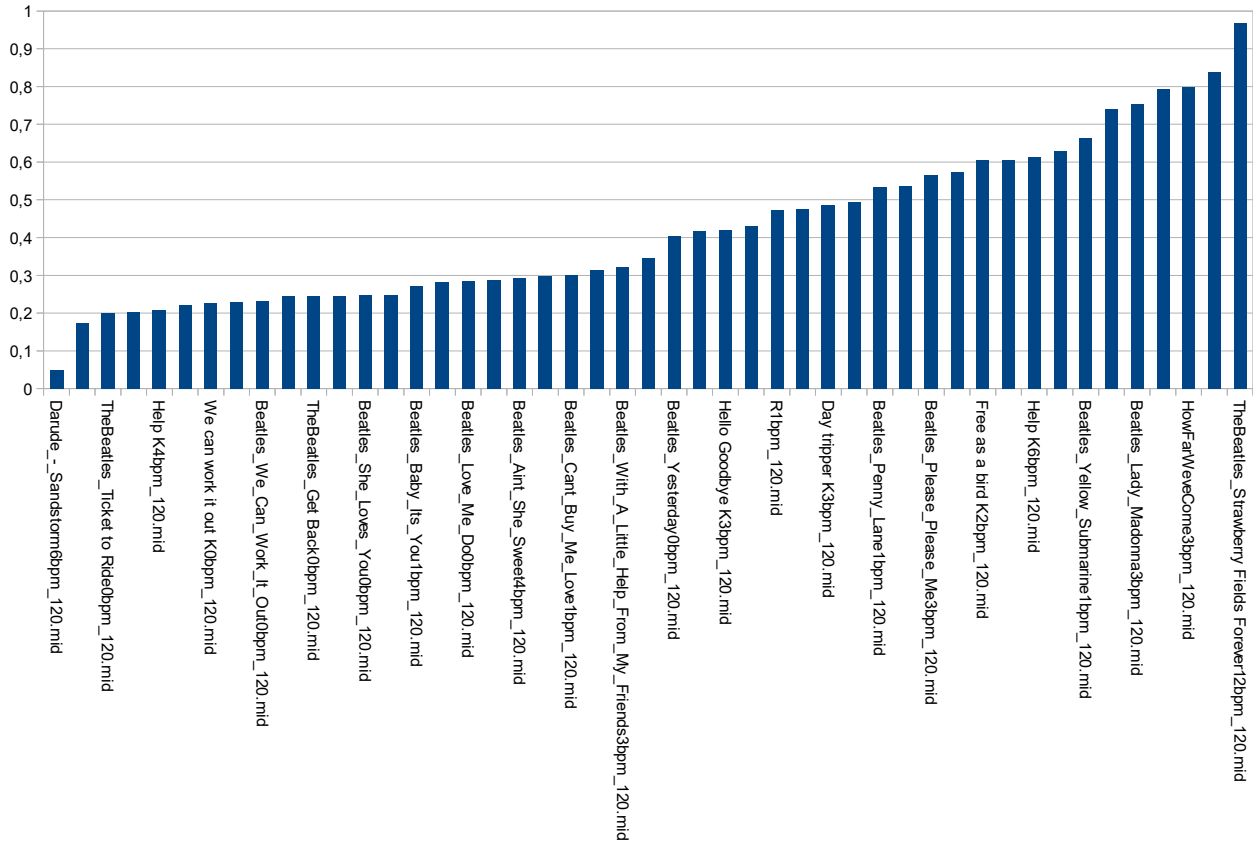


Fig 14: Compression Ratios Chart