ECS7006P – Music Informatics – 2021/22

Coursework 2 Report – Tolly Collins – 200553283

1 Introduction

In this assignment we investigate an approach to audio identification. This is a process where a system is presented with a query audio fragment and its task is to identify which musical recording the fragment belongs to. Audio identification is a process which requires a high degree of robustness [1], as query fragments can have significant levels of background noise and distortion. This is also a system with high-specificity, in that an exact copy of the original query audio must be retrieved [2]. High specificity requires a fine level of detail in the content analysis, but this requirement on detail must be balanced with both the robustness requirement, but also a compactness of information due to a potentially vast comparison database leading to long search times. Here, we follow the audio fingerprinting approach based on spectral peaks [3], applying the approach of inverted lists to allow for quicker database search times compared with the approach of time-window matching.

2 Method

The first part of the audio matching process is to create a database of fingerprints, one for each audio file. To generate a fingerprint, first a magnitude spectrogram is calculated. A max filter is used to find local amplitude peaks, and a threshold is applied. This threshold acts as a universal level across all of the database items, and it also serves to reduce the quantity of information within a fingerprint. The tuning of parameters such as the threshold level is discussed below. The result of this peak-picking step is called a constellation map.

The next step is to convert this constellation map into a hash map, which is done for several reasons. Firstly, there is a level of downsampling where the frequency information is grouped into a smaller number of bins. The information in the time domain can also be downsampled into a smaller number of bins. Furthermore, we convert the information within each bin into a binary value of ‘1’ representing at least one peak within that bin, and ‘0’ representing no peaks. We can see that this process serves to significantly reduce the amount of information for comparison, and the downsampling and binary representation can give a degree if robustness against noise in the query recording. On the other hand, the information cannot be downsampled too far, otherwise the system will lose the ability to select the correct track for a query and the accuracy will decrease.

From the hash map, an inverted list is created, which is the audio fingerprint which will be stored in the database. This consists of key-value pairs of frequency bin position and time bin position wherever the hash map has the value ‘1’.

When a query is given to the system, its audio fingerprint is calculated in the same way. Crucially, the parameters governing the size of the time and frequency bins must be the same for both the database and query fingerprints for comparison to be possible. However, we did investigate the effect of choosing different peak threshold levels between database and query items.

For each database item, a matching score is calculated. This score gives an integer value corresponding to the maximal number of hash map positions which have the value of ‘1’ in both query and lookup fingerprint for any given time-shift of the query fragment. The database items with the largest scores can then be returned as matches for the query.

3 Parameter tuning

3.1 Visual Tuning

In order to gain a broad understanding of the sensible range for the adjustable parameters, we represented each stage visually. First, the constellation maps were compared to the spectrogram of several audio examples to gauge what sort of threshold levels and bin sizes would be sensible.

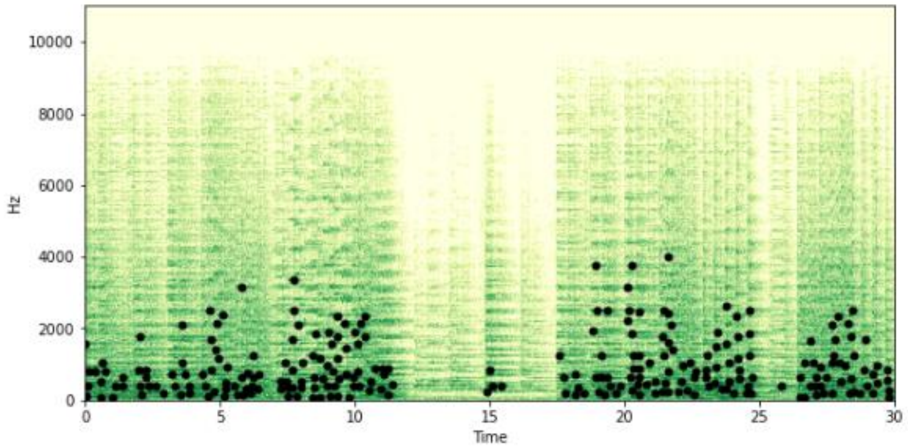
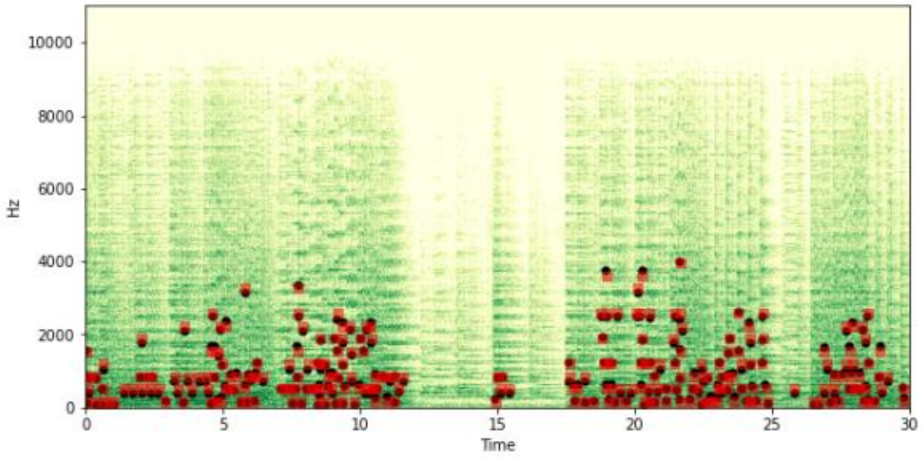


Fig 1: Constellation map with a peak magnitude threshold of 5. The spectrogram has a window length of 1024 samples and a hop size of 512. A Hann window is used.

Hash maps were then graphed to gain an understanding of the relationship between the hop sizes, threshold levels and how many constellation map peaks were going into each hash map bin.



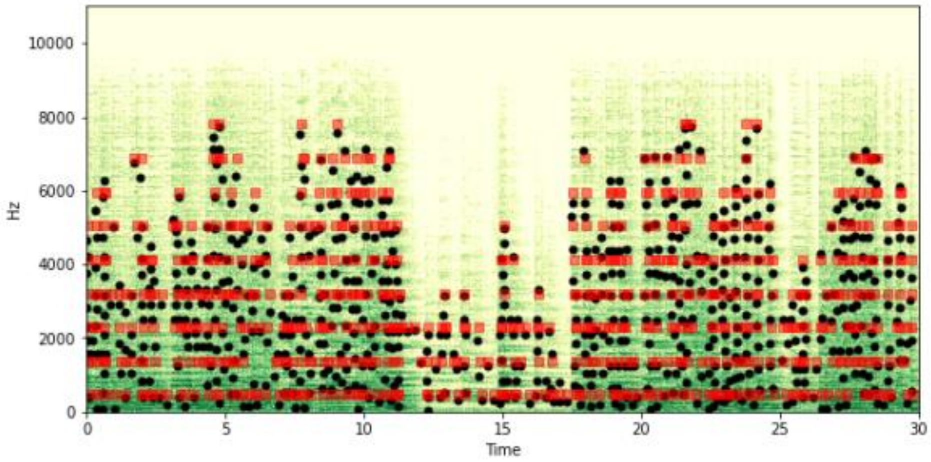


Fig 2: a) A hash map (red squares correspond with hashes with the value '1') corresponding to the constellation map from fig. 2. b) A hash map for the same audio sample with fewer frequency bins and a lower peak-picking threshold.

We can see in the figure above that the choice of parameters makes a substantial difference to the fingerprint obtained. The above example has 32 frequency bins and a peak magnitude threshold of 5, and we see that most of the information is retained in the hashing process. In the lower graph we see the effect of lowering the peak threshold to 0.3. This means that the constellation map has many more peaks, particularly at the higher frequencies. This could be important for the accuracy of the identification process. However, we also see that with only 12 frequency bins, a lot of the information is lost and it seems that the resolution here is too low.

2.3 Parameter searching and evaluation methodology

It is common to use Mean Average Precision scores as an evaluation metric for audio identification systems. However, in this case there was only one relevant database item per query, so we focused on 'mean accuracy'. In this case, a score of 1 was given if the correct item was identified with any one of the top three matching scores, and 0 otherwise. The average across all queries was then taken. The average area under the ROC curve (AUC_ROC) score was also calculated to give an idea of how high up the confidence list the correct item was on average. An item being identified correctly with highest confidence would be given a score of '1.0', as the recall would be 1 at all confidence levels.

In total, there were 9 parameters tuned (see below). This process was conducted in several stages. The first was to create a Python class to perform random searches. The accuracy and AUC_ROC scores could then be saved for each set of parameters tested. For each instance of the class, the metrics from the previous instance were analysed and then a new range of values were provided for each parameter and a random selection was taken for each iteration. This process allowed a much quicker approach towards the optimal parameter values than would have been possible with a grid search [3]. It is also a more efficient approach than tuning each parameter in turn, as the performance relative to different parameters is not independent, so changing one parameter could give a different optimal range for another.

The query data were split into an evaluation set and a test set, so the best performing parameter combinations could be verified with a separate set of data.

3 Results

In general, the STFT spectrogram tended to produce better results than the Mel spectrogram. This suggests that a higher resolution at higher frequencies could be beneficial for the audio identification process, and that lower-frequency information can be grouped together without much loss in performance. It should be noted that much of the music content used for this study was from the Classical genre, which may contain less important lower-frequency information than some other genres such as dance. Shorter hop lengths generally produced better results, but the gain in performance was minimal and should be weighed against the increased calculation time. An FFT length of 1024 samples was generally better than 512.

Perhaps the most important parameter was the peak threshold level. For levels above 5, top-3 accuracy scores were in the region of 5-20%. When the threshold was lowered to around 0.3, the accuracy was generally above 50%. It was found that accuracy started to decrease again for threshold levels below 0.3, as this would be allowing too much noise information in from the query samples. Furthermore, there was no gain to the accuracy score made by assigning a different threshold level to the queries compared with the database fingerprints. However, the ROC_AUC score was marginally improved, by about 1-2% by having a higher query peak threshold compared with the database threshold. This may suggest that raising the threshold for query fragments could give a marginal robustness increase with fewer noise peaks affecting the hash map, moving the correct match up the confidence list on average.

The time and frequency lengths of the peak picking window had little effect on the accuracy as long as they were within a reasonable range.

For the hash maps, the time resolution had much more of an impact on the accuracy scores than the frequency resolution. We found that a time hop of 1 sample was optimal, but a much larger frequency hop of 32 STFT bins had no negative impact on performance. The optimal parameters are given below:

Parameter	**Optimal Value**
Spectrogram type	STFT
Spectrogram Window Length	1024
Spectrogram Hop Length	256
Maximum Filter Frequency Window Length	32
Maximum Filter Time Window Length	8
Peak Picking Magnitude Threshold	0.4
Multiplier for Peak Threshold for Queries	1.0
Hash Map Time Hop Length	1
Hash Map Frequency Hop Length	32

Best mean top-3 accuracy score	**0.64**
Best ROC_AUC score	**0.78**

4 Evaluation and Conclusion

A reasonable level of accuracy was achieved with this fingerprint matching approach. However, the search time was slow, rendering this system inappropriate for industrial use. Other methods have been proposed, such as using hash triples in target zones, which can reduce the search time [1, 3].

We find that the peak magnitude threshold level and the time resolution are the most important parameters. In general, there is a trade-off to be achieved between the accuracy level and the search time, with lower resolution representations giving quicker searches at the cost of accuracy.

For further investigations, it would be interesting to compare the accuracy of this method against the target zone method [3]. It would also be interesting to investigate further processing methods such as mean filtering of the audio signal for the query samples. It is possible that the impact of noise and distortion could be reduced with the right choice of pre-processing. There are also further parameters that could be tuned such as the choice of window function for the spectrogram calculation, or use of techniques such as the constant-Q spectrogram.

5 References:

[1] Muller (2021) – Fundamentals of Music Processing Using Python and Jupyter Notebooks, 2nd Ed. *Springer*

[2] Grosche et al (2012) – Audio Content-Based Music Retrieval. *Schloss Dagstuhl—Leibniz-Zentrum für Informatik, 3: 157–174.*

[3] Wang (2003) – An Industrial Strength Audio Search Algorithm. *ISMIR*

[4] Bergstra & Bengio (2012) – Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research.*