DETECTING MUSIC IN A NOISY, EVERYDAY ENVIRONMENT

Bachelor's Thesis
Ivo Bril, s2001179, i.bril@student.rug.nl

Abstract: This paper is written to describe how music can be broken down to features understandable for a computer using Continuity Preserving Signal Processing. The music used in this project was mixed with ambient noise consisting of seventeen everyday scenarios. By analyzing the sound samples and extracting general features that may indicate music it has been made possible to design and implement a detector which can detect music in seventeen everyday scenarios using these features. This paper shows how the features were found, the design process of the detector, its early results and mentions possible improvements.

1. Introduction

The everyday feat of recognizing sounds, their sources and their meaning seems like a trivially simple task. Learning new songs or humming along with a song one has never even heard of before is entirely possible with human hearing. Only when one starts to work with sound recognition or analysis and encounters complex problems does one realize the strength and robustness of the auditory systems that living creatures possess in this world. A typical question scientists working in sound classification and evaluation could ask is 'what would it take to let a computer algorithm recognize music?'. Considerable efforts have been made in this field of research, all trying to explain music in ways a computer could understand it. Formulas have been designed for recognizing the onset of music (Zhou & Reiss, 2011) and there has been philosophized about interesting ways of solving problems the field still has (Bregman, 1990). Yet there has not been done a great deal of research in recognizing music in real-life situations. For this thesis, real-world situations refer to real-world scenarios, like shopping, being at home or some place else. Certainly, applets have been developed that try to determine what kind of song is playing. But these techniques approach a different problem. Instead of using a small sample of recorded sound and trying to find a match in a huge database of samples (Wang, 2006), the challenge tackled in this paper demands generalizable features that are indicative of music. The system described in this paper will report most music detected out of the signal it receives and says when the music was playing. One of the bigger problems with everyday scenarios, universal for most sound analysis, are the levels of ambient sounds. How to separate music from these ambient sounds will be dealt with in the next section.

1.1 Dealing with noise

One problem with real-world sound detection is being able to ignore all the interfering ambient sounds that will be present in the signal. Since humans can do this, why not try to imitate it? Andringa (2002) has developed a system which uses a representation that closely resembles the cochlea and its property for frequency segmentation. This system, named Continuity Preserving Signal Processing (henceforth named CPSP), mimics the way the basilar membrane of the human hearing systems relays information about incoming signals.

The rigid, curled-up membrane is sensitive to different frequencies at different positions (e.g. high frequencies at the base of the membrane because of its rigidity). But what makes it an interesting way of signal processing is the fact that it preserves continuity in time, place, and frequency. This allows one to follow the development of signal components through time and frequency. The latter property is what makes it stand out the most from more standard signal processing approaches, which often times discards knowledge about the actual structure of the music in favour of making the signal more analysable for the system.

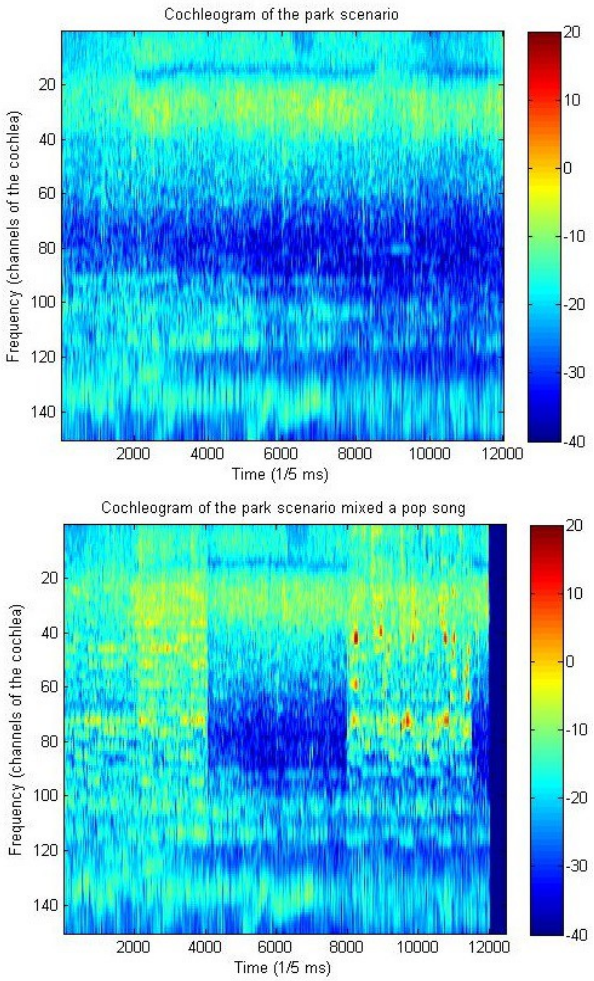

Figure 1.1: Two cochleograms, the upper being the park scenario and the bottom being the same scenario mixed with a pop song. The pop song is present from 0 till 4000 and 8000 till 12000.

1.2 Cochleograms

The data generated by CPSP comes in the form of a cochleogram. This is a spectrogram-like graph, containing frequency over time, with colour codes for the amount of decibels. The frequency is represented by channels because it mimics the cochlea. The greatest advantage of using this technique is the fact that it keeps the physical properties of the signal as intact as possible, which in result leads to great smoothness in the cochleogram. This smoothness keeps each individual signal component more physically coherent, making the data less susceptible for artifacts. An example of a cochleogram can be seen in Figure 1.1.

1.3 Classification

If cochleograms contain valuable information about the signals, what do they tell us about music? Do cochleograms show anything indicative of music when the signal contains music? This thesis has tried to find features in music that are present even in mixdowns of (non-studio-quality) music with an everyday scenario. How these features were found and how they've been used will be described in the method section. By finding these features a path is paved for a detector to use these features in detecting where music is present.

1.4 False positives

By definition, the detector searches for features that are highly indicative for music. Not only does this make it easier to make a robust system, but it also tries to pinpoint the essentials of music and what makes it special in comparison to other sounds. This does, however, make it of utmost importance that the features found are highly indicative. If these features cause the system to recognize anything from crying to sirens as music, then it is possible that the features that were chosen were not specific enough. It could be debated that these kinds of sounds, if they contain these kinds of features, are music and should be recognized as such. For the usability of this system however, I would argue otherwise. This system tries to recognize music, the kind of music humans have composed for the past millennia. Therefore, crying and siren-like sounds should not be classified as such. The questions that this paper tries to answer are as follows:

"Does a signal of music and ambient sounds, represented in a cochleogram, show any visual properties that are highly indicative for the presence and location of said music?"

"Can a detector, using features that are highly indicative to music, accurately detect the presence of music in noisy, everyday environments?"

1.5 Pre-existing ideas

Music has existed for centuries, has had all kinds of influences affect it, and is one of our cultural ways of self expression. The music we listen to in

this era has basic elements anyone can describe. They boil down to a few things:

- Rhythm: the sequential and consistent occurrences of sounds that indicate the tempo of a song. This is one of the possible features that the detector could use, as it certainly is indicative of music.
- Tonality: a highly changing or prolonged presence of sound that is sinusoid. Most music is tonal, be it by singing or the use of instruments, so it is likely to be a distinct structure in the cochleograms.

The expectation is that these pre-existing ideas will show themselves in the cochleograms as well. Another expectation is that, given the right approach, these features (and other features if they are found) allow a detector to find music fairly well.

1.6 Thesis format

Due to the nature of this research, which is of an exploratory nature, the thesis will consist of two method and result sections. The first method and result sections will explain what features were found to be indicative of music and how they were found. The second will explain how these features were then used to create a detector for the presence of music.

2. Finding features

Method

In this section will be described what pre-existing ideas there are about possible features, what cochleograms tell about music, and what functions are needed to find and extract the features that are present in the signal.

2.1 An informal analysis of a pure signal

Before looking at the mixed signal, it is good to check whether there is anything interesting in the unmixed signals. As can be seen in Figures 1.1 and 2.1 there are differences between the structure of music and that of a signal existing of ambient sounds.
Whereas the music signal has high amounts of decibels and strong horizontal linear structures.

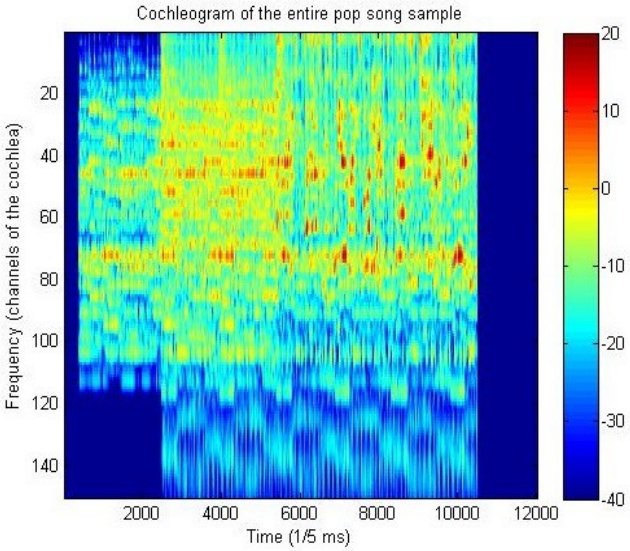

Figure 2.1: A Figure of the same pop song as seen in Figure 1.1, but without the pause in the middle.

The ambient sounds show hardly any of these features. This kind of analysis was done with all scenarios and music signals and the difference was present in every comparison. This shows that this music selection has some indicative properties. The next step is seeing whether these features are present when the signals are mixed. The next section will explain how the signals are mixed and why.

2.2 Mixed Signals (test data)

So music shows some interesting structures when visualized, but do they remain when they are mixed with an everyday scenario? The sound samples from everyday scenarios available at http://www.daresounds.org/ were used (Andringa, Van der Linde & Krijnders, 2009). They include a wide variety of real-world scenarios. From low level of ambient noise (e.g. bedroom), to high level (e.g. shopping district). These samples are each one minute long, with no specific beginning or end. The reason this set was used is because of the variety in scenarios and the good quality of the samples. The music-styles chosen to mix with these scenarios vary from rap to modern classical music. The test data has five styles in total (rap, hard rock, electronic, piano-pop and modern classical music). The five music-styles were chosen as such because they encompass enough variety of styles for this thesis to test its detector. It is in no way an all encompassing set of styles to say that it can

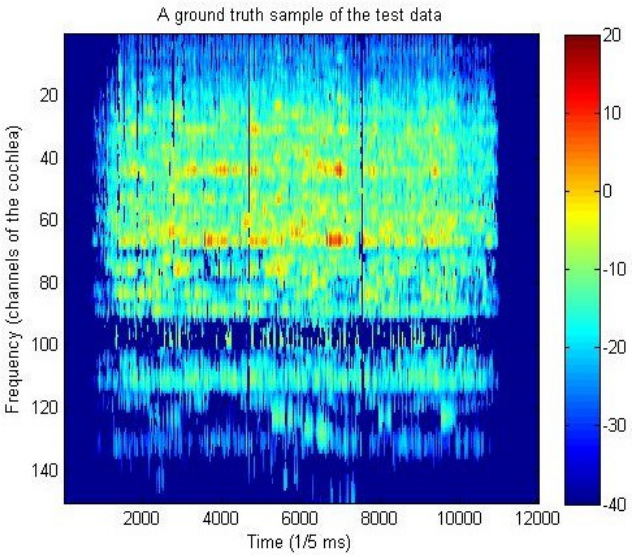

Figure 2.2: A cochleogram showing the ground truth of the study room scenario mixed with the classical piano music sample.

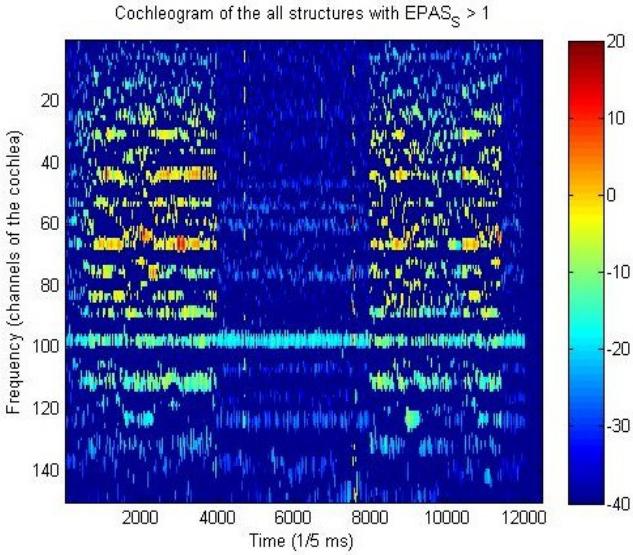

Figure 2.3: A cochleogram showing the structures which have *EPAS_S* values greater than one. The signal consists of a study room mixed with the classical piano music sample.

handle all music-styles. The music is mixed with the scenarios using Adobe Soundbooth, with each music sample being mixed in four ways with the scenarios:

- The music is only present during the first 20 seconds of the sample.
- The music is only present during the last 20 seconds of the sample.
- The music is present during the first and the last 20 seconds of the sample.
- The music is present during most of the sample (50 sec.), with five seconds of silence before and after the music sample.

The four different placements were chosen to make sure that there is an adequate number of differing mixdowns. Ideally all mixdowns would have more randomly placed music samples instead of the fixed positions they have now, but this was not possible to do with Adobe Soundbooth and would increase the workload tremendously as it had to be done by hand. Still, Soundbooth was chosen due to its intuitive design and its usability. Another possibility was Audacity, but its functionality is less and it is not as intuitive as Soundbooth. The music was made more realistic (noised up) using sound-altering effects from Adobe Soundbooth. These effects made the music sound as if it was played in a distant room, mimicking the way music would sound in real life (see the appendix for specifics). This makes the music more fit for the scenarios as opposed to the clean sound of studio-quality music. With each scenario the volume was equalized to ensure that a more realistic mixdown was created. The appropriate effects and volume equalization was determined by listening to it. The actual test data was then created by subtracting a matrix of the dB values of the scenario from the matrix with dB values of the mixdown. This results in a matrix with only the music present in the signal. For every scenario there were five music styles, per music style there were four placements. As each test file is one minute long, this leads to a total of 340 minutes of test data. An example of a test file can be seen in Figure 2.2.

2.3 CPSP

CPSP has many possibilities for examining the data. Some of which are very useful for finding information about rhythm, structures inside the signal and being able to distinguish between sound and noise. When CPSP processes a signal, it creates a struct '*D*' which contains several matrices with information about the signal. In the following subsections will be explained what functions or matrices from *D* were used for the analysis and why.

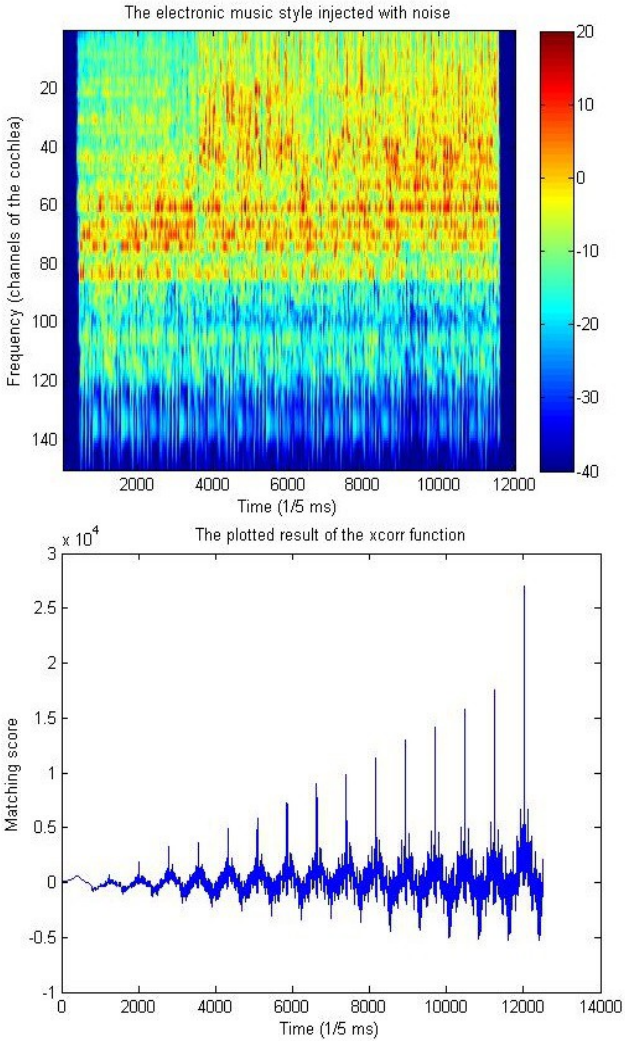

Figure 2.4: The results of the *xcorr* function shows that there is a clear rhythm. Peaks indicate a reoccurence of a structure in the given channel. The distance between the peaks tells how much time has passed between the reoccurences.

2.3.1 Volume and tonality

D.EdB is the cochleogram represented in matrix form. This matrix shows the signal in the time/frequency domain, with the dB levels (volume) made visible through a colour range (as one could see in Figure 1.1). This matrix was used to visualize the results of the other functions used and to make the Figures used in this thesis.

D.EPAS_S is a matrix that allows the user to identify tonal components in the signal. CPSP takes the entire signal and evaluates all structures from the signal in their adherence to the expectation for noise. If a structure in the signal diverges statistically from noise, the data points in that structure get a high absolute value.

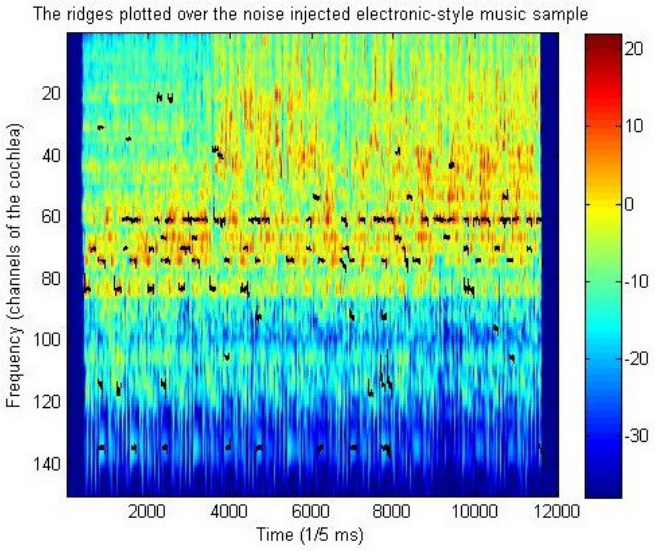

Figure 2.5: A cochleogram showing that a music sample lends itself well to the *findRidgesInPeakMask* function. The black lines represent the ridges that were found.

The values in the *D.EPAS_S* matrix consist of standard deviations of the mean of noise (for a more elaborate explanation, see the appendix). One of the pre-existing ideas about music mentioned at the beginning of this section was tonality, so naturally *D.EPAS_S* is an extremely useful representation. As can be seen in Figure 2.3, putting a threshold on all *EPAS_S* values gives a great insight in structures that are indicative of tonal sounds.

2.3.2 xcorr & findRidgesInPeakMask

xcorr is a function that looks at a given channel of the signal and determines whether there are structures in the signal that repeat themselves. If these repetitions are present in a consequent matter, one can say that there is a rhythmic character to it. Using this function makes it possible to find rhythm in a signal, though the correct channel has to be chosen. Finding the correct channel is a problem which will be addressed in the method section of the detector. As rhythm is also one of the pre-existing ideas about music, this function is used to try and see if it really is a suitable feature. As can be seen in Figure 2.4, the noise-injected music signal itself shows a strong rhythm. *findRidgesInPeakMask* finds sequences of peaks in energy. It connects peaks that it finds within a given range and makes plot-able lines which can

be plotted over the *D.EdB* cochleogram, thus showing coherent ridges found in that signal. This function has some usability for making the computer find the longer lasting tones itself. The ridges can switch from channel 40 to 45 in a few time frames due to the function allowing a jump between channels (an example can be seen in Figure 2.5). This makes it harder to find prolonged horizontal structures of sound. One of the things that describes music is a rhythmically present sound which can change in frequency over time. The changes between frequencies is something that this function can detect, while still being able to keep track of the ridge that it has found.

Results

All of the above mentioned functions have been used in the pursuit of finding features for music. Some worked, some were quickly abandoned and some have not been used to their full extent yet. In this section will be explained what features were found.

3.1 dB differences

One of the more obvious features, as one can see in Figure 1.1, is the amount of energy in dB present in the more prominent notes of the music (found in the *D.EdB* matrix). A high dB value could therefore indicate the presence of music. If it had been so easy, this thesis would not have existed. Take the following cochleogram (Figure 3.1) for example, it shows huge amounts of energy, but it is not all from the music. This shows that using dB in finding music is treacherous, because a high amount of dB does not necessarily mean that the signal contains music.

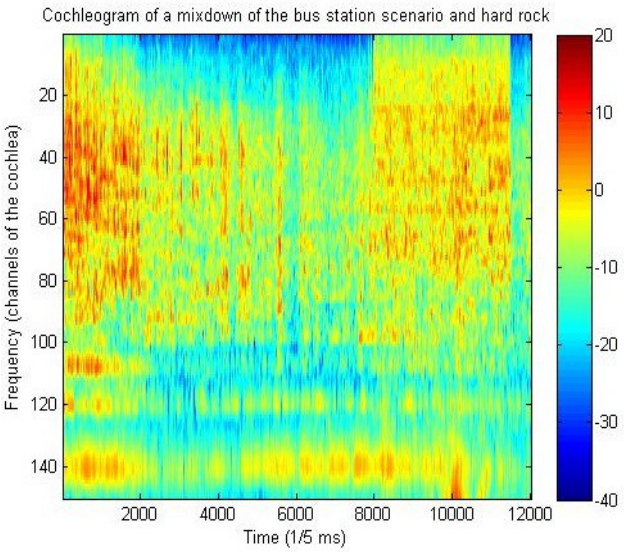

Figure 3.1: A cochleogram consisting of one of the everyday scenarios, combined with hard rock music in the last 20 seconds of the signal. The great amounts of dB in the beginning are helicopter noise.

3.2 Tonality and frequencies

The *EPAS_S* tell a lot about the tonality of the signal. As can be seen in Figure 3.3, there is a certain range of frequencies where a lot of the values are likely to be tonal sounds. This characteristic is present in all samples, which means it is not necessarily true for all music. This range is interesting as it tells something about the sounds found outside this range. They could certainly still be music, but the possibility of it being noise becomes much more likely. Especially the lower frequencies seem to be more

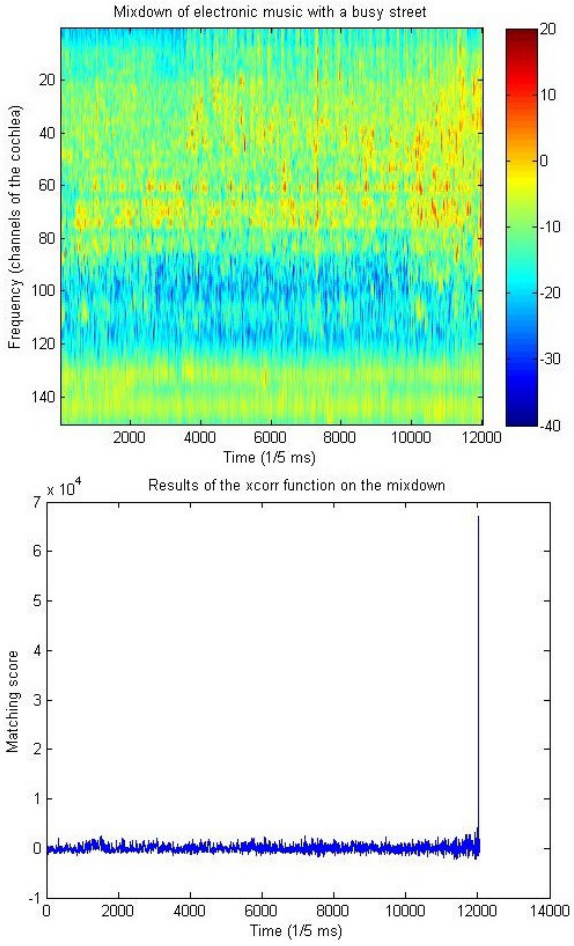

Figure 3.2: The results of the *xcorr* shows that there are no clear matches, so no rhythm can be found. Channel 150 was examined here.

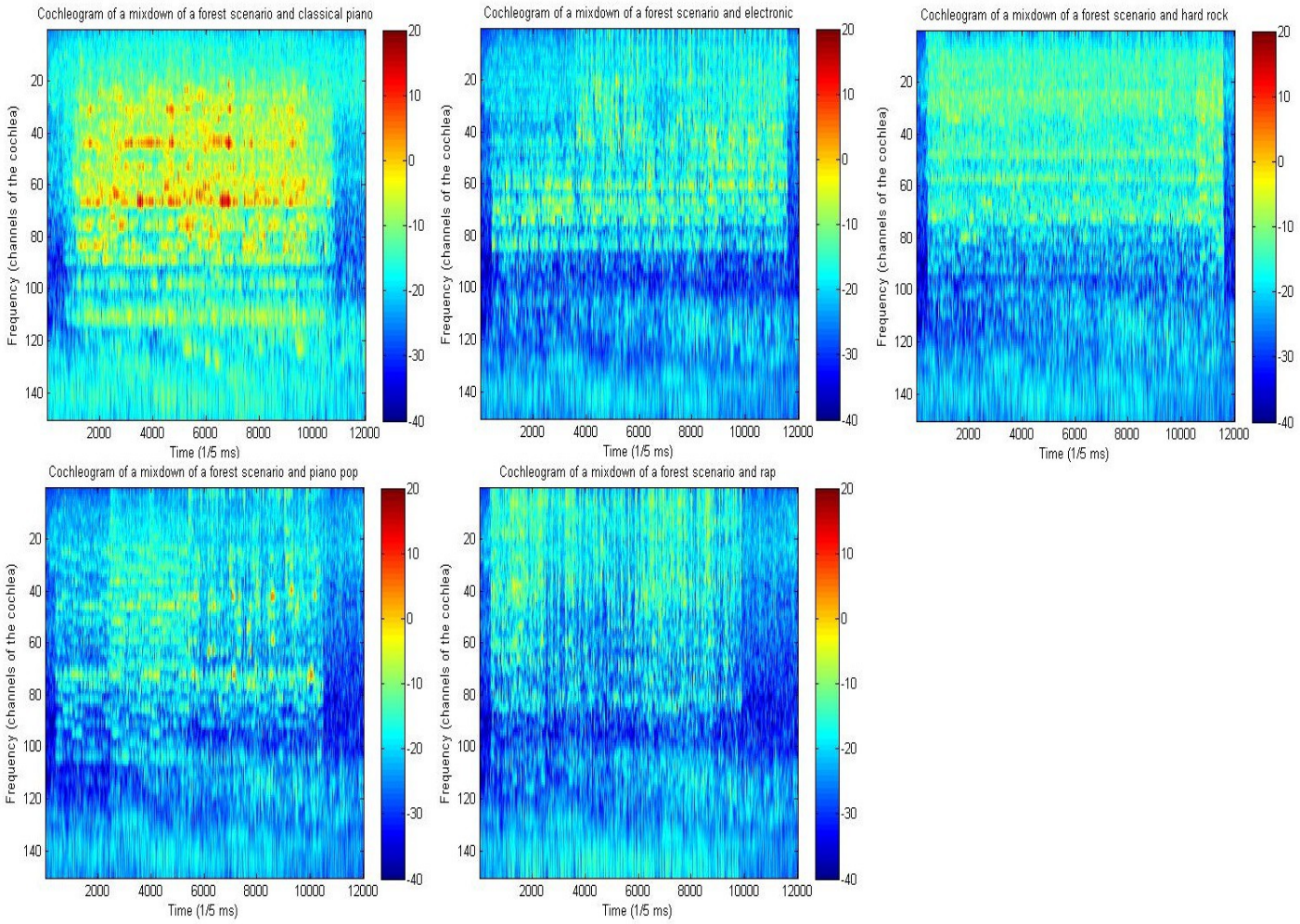

Figure 3.3: Five cochleograms, each of the forest-scenario mixed with a different music sample. The music is present during the entire signal, except for the first and last five seconds (the last two stop around 50 seconds in the signal).

predominantly occupied by noise than sounds. The strongest tonal sounds are usually found in the range of 260 to 4200 Hz. This range is one of the features that shows promise if more knowledge of frequencies had been used. Tone ladders are a frequent sight in music and the fact that certain notes have fairly specific frequencies makes them strong possibilities for music detection and analysis. This thesis has not used this knowledge for the detector due to a lack of time and in-depth knowledge about the subject to implement it in the detector.

3.3 Rhythm

One of the more prominent features in a signal of pure music is the fact that it contains a certain rhythm. Most western music can be described in terms of beats-per-minute (bpm). Using the function *xcorr*, it is possible to find correlations between the presence of pulse-like sounds and the frequency with which they are present in the signal. If there is a correlation between time and the presence of pulses, it has a high probability of telling the rhythm of the music that is present in the signal. This feature was not as useful as it seems. Most pulses from rhythm are present at the lower frequencies (e.g. lower than 260 Hz), which is also the range where ambient sounds reside. This leads to a conflict between the signals and finally in the loss of consistency in the rhythmic pulses, which makes it difficult to find the rhythm using the *xcorr* function. An example of this can be seen in Figure 3.2.

3.4 Ridges

One of the more useful structures found in the signals were ridges. Ridges are horizontal lines of consistently high dB or a high amount of

standard-deviations (when using *EPAS_S*). They indicate a prolonged presence of sound, which indicates a high chance of it being music or something else non-noise related. This feature was easily found by looking at the structures of the sound in the cochleograms (Figure 2.3 shows these ridges to great extent). It has proven its self as a valid feature for detection and it was central in the design of the detector

These were the features found to be most prominent in the music-styles examined with the functions mentioned earlier. They are not a complete set of all possible features found to be highly indicative of music. They are merely the results from analysing a few music-styles combined with seventeen different scenarios.

4 . Detector

Method

For this section, the way the detector is built and tested will be explained. This includes explanations for choices made in the design of the detector. Figure 4.1 shows a diagram of the steps taken by by the detector and what data it needs.

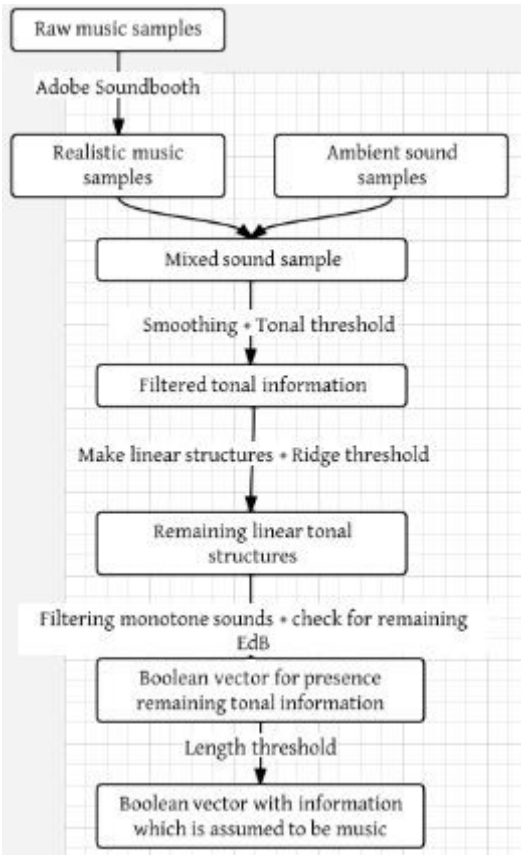

Figure 4.1: A diagram showing the process with which the detector determines whether music is present in the signal.

4.1 Detector

The detector is built with a simple idea in mind: take the information from a cochleogram (like the ones in Figure 3.2) and use the features found to create criteria. All the data that meets the criteria sufficiently is considered music. Each of the criteria used for this detector will be discussed shortly in the order in which it is applied, what it does and why it was used.

4.2.1 Data has to be tonal

The very first criterion is an obvious but important one: the data has to be (remotely) tonal. With *EPAS_S* it is known what structures could be considered to differ from noise, structures which could possibly be music. However, a lot of the matrix values only have a small standard-deviation from noise, so there has to be a threshold. The key here is to keep enough information about the signal in the matrix so there is enough data to work with, but getting rid of the unnecessary information. A standard-deviation of one to one-and-a-half is effective most of the time. To make this threshold a bit flexible the detector takes the average of all cells of the *EPAS_S* matrix and adds this to the 1.5 threshold. This average holds information about the overall noisiness of the signal, lowering the threshold if there is a negative average (e.g. a low overall deviation from noise). And it increases the threshold if the signal differs greatly from noise. This threshold is administered after a *movAv* has been applied to the *EPAS_S* values. *MovAv* makes each matrix cell the average of it's surrounding cells, where the reach of its surrounding points is determined by the user (the range and used values can be found in the appendix). This makes the stronger differing values more prominent in the signal.

4.2.2 Filter for Ridges

Now that the ridges are more prominently present in the signal, it is time to find them in the signal and discard any other excess data. For this

the function *lengthAreas* is very useful. Explained briefly, this function takes ridges and makes all values in that ridge equal to the total length of that ridge. So *lengthAreas* will make the longest ridges have the highest value. Some of the interesting ridges aren't fully connected to each other. With that in mind there has to be a small gap allowed between the ridges to keep the longer ridges connected. This way, the longer ridges, which are more probable of being music, remain present in the data.
Now there are many ridges, each with their own values. The data still contains many small snippets of sound that passed the first criterion. These snippets are often very short, so a new criterion is necessary. This case is the same as the first criterion as there is need for another threshold that keeps the valuable data, but kicks out the remnants of other ambient tonal sounds. The threshold that was settled for in this thesis is a minimal length of 150, which is equivalent to three quarters of a second. By trying out several values it turned out that ¾ of a second strikes a balance between keeping enough interesting data and removing enough ambient tonal sounds. This latter threshold is purely practical in the sense that it relieves the matrix from possible ambient sounds while maintaining enough information about the signal to work with.

4.2.3 Keeping out monotone sounds

Up until now everything is going fine, the criteria do a great job of filtering out everything that isn't a ridge. Yet one of the stronger ambient sounds is still in the data, sounds that differ from noise in the same way that the music ridges do. These sounds consist of long tonal components produced by, for example, electronic devices (e.g. refrigerator, air conditioning, etc.). Figure 2.3 shows such a source, the light-blue structure around channel 100. They are present during the entire signal and have become one long ridge due to the computations that have been done so far. The latter part is what makes this problem easier to solve, because finding an extremely long ridge is fairly easy. The way the detector tries to deal with these ridges is by looking at each row and counting the fraction of cells which has an EdB level higher than the lowest EdB possible in the signal. The detector tries to find channels which have sound for at least 80 percent of the entire signal. Channels that meet this criterion have either a lot of small ridges or one big ridge. Due to the fact that the remaining data has been severely filtered, the first option is rare. The latter, on the other hand, is found in most scenarios with a clear monotonous ambient sound. The channels for which the 80 percent mark stands are thrown away, along with three channels on each side to make sure that the ridge is removed completely. The 80 percent mark was chosen due to the fact that the filters up until now could have diminished the monotone sounds slightly. Making the percentage higher would therefore create a risk for leaving too many lengthy ridges in the data. Lowering the threshold would also bring about problems as the chance of losing the precious ridges that have been found with the filters becomes higher. 80 percent has been found to strike the right balance between these to problems. As this thesis is trying to make a detector and show what is necessary to create such a system, there has not been done any further research in finding the optimal percentage for this filter.

4.2.4 The length of the ridges

The ridges that remain are very close to being classified as music. The data consists of tonal ridges that have a length of at least ¾ of a second and do not span over more than 80 percent of the signal. The final step now is to see where the ridges are and take note of this in a vector consisting of the time. For every column where some energy can be found, the detector keeps track of where it starts and ends. All columns from the start until the end are registered in the array as ones (The array starts out with only zeros). This new array has now been filled with the knowledge of where one or more ridges were present. This array will be tested by comparing it to a similar array made with the test data. Before it compares them there is one final criterion that has to be met. This criterion demands that the segments of ones in the array has to be longer than five seconds. Music is something that is listened to in longer segments, so this should be reflected upon in the criteria. Because the ridges vary in their channel it is not possible to do this when it is still two dimensional data, as can be

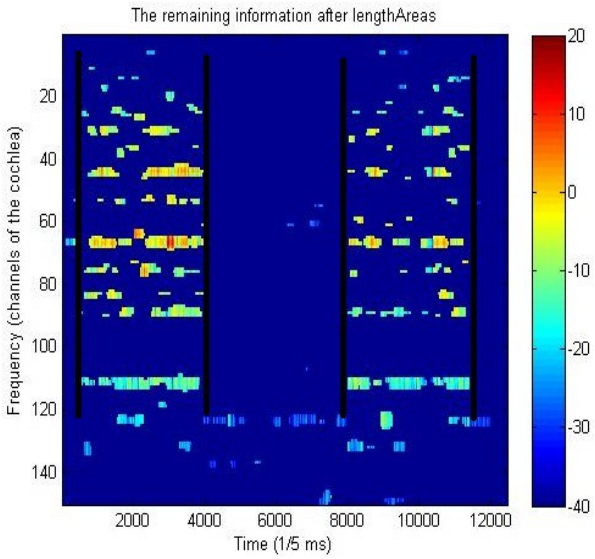

Figure 4.2: A Cochleogram where the final step would be to convert it to a 1 by 12000 matrix, only keeping the information where there was at least some energy. The music is present between the black lines (which are purely an indication).

seen in Figure 4.2. The lines give an indication of where the music is. Especially the right segment shows that demanding the ridges to be longer than five seconds would result in throwing away too much the data. The data you see outside of the segments are all relatively short and don't seem to be from music. By demanding music to be longer than five seconds, there is a fail-safe for the small snippets of tonal ambient sounds that made it all the way through all criteria. Again, due to the aim of this thesis, no research has been done in trying to find the optimal length. With all these criteria, the detector ran through all the data, the results of this run will be revealed in the next section.

Results

The performance was measured in two ways: how much music (i.e. columns) the detector had found that was actually present and how much music it had found too much. The total score over all seventeen scenarios, each with the five music styles and four placements was as follows:

- Percentage of supposed music found too little (false negative): 12,49%
- Percentage of supposed music found too much (false positive): 24,11%

So the criteria of the detector seem to be too weak, some tonal, non-music structures get through. The detector, thankfully, does a steady job of finding the actual music. The overall performance, however, doesn't provide all the peculiar differences between each scenario or music style. Table 4.1 shows the results divided over each scenario. It shows that scenarios with stronger and/or more ambient sounds, such as a busy street or a bus station, are harder to analyse. When ambient sounds are more prominently present in the signal, they differ more from the structures that noise would have. This, in return, leads to more non-music tonal structures present in the *EPAS_S* values. It seems that the criteria are not flexible enough to handle a more cluttered signal. In some of the more quiet scenarios the detector finds a lot more music than there is actually present. However, not all quiet scenarios suffer from this, so why do some do? These scenarios do because there is a monotonous sound source present, a strong ambient noise that is present throughout almost the entire signal. It seems that these kind of signals are not as easily dealt with, because the detector has a criterion to deal with this kind of structure. The difference per music style seen over each scenario is shown in table 4.2 including the averages for each music style. This table shows the performance of the detector for each music style used in the test set. The results reveal that most scenarios lead to the same

Table 4.1: A table with the results divided per scenario. The values consist of the averages of each music style per position.

Scenario	Percentage Missed	Percentage Too Much
Beach	6,52	0,97
Bedroom	0,09	10,45
Bicycle	18,79	24,42
Bus	9,4	34,87
Bus Station	22,66	51,32
Busy Street	35,49	46,75
Flat	3,61	7,21
Forest	10,31	0,5
Hallway	0,03	44,1
Kitchen	4,09	35,1
Livingroom	3,35	9,02
Park	10,27	43,87
Pedestrian Area	24,53	22,11
Quiet Street	31,53	1,21
Residential Area	8,74	28,51
Study	0,05	36,87
Supermarket	22,82	13,12

results, so the music style does not matter much. However, there is one music style which is hard to detect using the criteria from this thesis, namely rap music. This music style only has similar results to the other styles when it is a quiet scenario, and even then it differs occasionally. One reason for this difference in performance is due to the structure of rap music as a whole. Rap music consists mostly out of a beat and rhythmic rhymes. So instead of singing and prolonging notes, rap is more staccato-like in its structure. So rap only has ridges if the beat and tune of the song have prolonged notes, something which is not all that frequent in the genre. This lack of ridges makes it hard for the detector to find the music in signals where other, more prominent ridges are present. Although this detector can not detect rap well, it could be detected better using a detector more focussed on rhythm.

5. Conclusion

Hopefully, this thesis gives an interesting insight in the world of music and signal analysis. As it turns out, music has some properties that make it unique in a sense, but the ones on which this thesis capitalized differ between music genres. However, it can be said that prolonged tones do encompass one of the more usable features found to be indicative of music. Rhythm seems to be a feature that is not strong enough for signal analysis. The fact that it is one of the more prominent properties of music makes it hard to believe that rhythm can be ignored as a possible detectable feature in mixed signals completely. This thesis failed to use the function *xcorr* to its fullest, as it works best if the right channel can be found to find the pulses in. As this can be a different channel for each new signal that the detector gets to analyse, a dynamic way of determining the right channel would be a solution to the problem encountered in this thesis.

Another possible gem for this kind of signal analysis is the information residing in the frequencies of music. As this thesis has shown, there are certain areas within the frequency channels of CPSP that include most of the tonal energy and its structures. Capitalizing more on this using tone ladders and other knowledge

Table 4.2: The results of each music style per scenario separated into precentage missed and percentage found too much.

Scenario		Beach	Bedroom	Bicycle	Bus	Bus Station	Busy Street	Flat	Forest	Hallway	Kitchen	Living Room	Park	Pedestrian Area	Quiet Street	Residential Area	Study	Supermarket	Average
Music-Style	Type of result																		
Classic	Percentage Missed	3.09	0.14	9.22	4.51	4.36	4.98	2.38	3.57	0.00	0.00	0.75	1.43	19.21	7.45	2.08	0.10	9.60	4.29
	Percentage too much	2.47	9.98	21.64	35.72	50.83	34.18	9.75	0.09	46.11	36.49	9.02	41.23	21.80	0.37	29.30	32.85	1.69	22.56
Electronic	Percentage Missed	6.65	0.15	23.38	0.00	17.56	28.33	2.14	0.29	0.06	0.25	0.15	0.38	51.24	29.29	0.63	0.00	23.68	10.83
	Percentage too much	0.94	10.04	25.82	32.96	48.97	37.20	6.12	0.74	44.28	34.99	11.89	41.11	33.72	0.60	25.86	37.80	0.96	23.18
Hard Rock	Percentage Missed	6.85	0.07	2.36	0.00	32.31	22.30	4.89	4.63	0.00	6.20	3.69	1.43	18.35	32.56	7.72	0.00	17.30	9.45
	Percentage too much	0.20	12.74	18.14	30.77	50.63	38.37	6.46	0.60	43.41	34.62	7.41	40.02	18.01	0.44	26.53	44.92	24.06	23.37
Pop	Percentage Missed	0.29	0.04	8.08	0.00	27.75	49.62	0.15	0.75	0.09	0.00	0.14	0.00	11.00	22.54	0.09	0.17	8.80	7.62
	Percentage too much	0.38	11.19	19.53	30.76	53.02	52.97	6.65	0.58	43.12	32.55	9.37	39.68	17.82	0.40	26.45	35.39	9.81	22.92
Rap	Percentage Missed	15.71	0.05	50.89	42.51	31.31	72.21	8.46	42.29	0.00	13.98	12.01	48.12	22.85	65.79	33.16	0.00	54.72	30.24
	Percentage too much	0.87	8.28	36.98	44.13	53.13	71.03	7.03	0.49	43.56	36.85	7.43	57.29	19.20	4.26	34.43	33.42	29.10	28.68

from music theory about frequencies (e.g. instruments produce multiple harmonics), it is certainly possible to make a more robust and flexible detector. The problem of filtering monotone sounds is an essential one for this kind of detector to work. The solution used in this thesis has proven itself as being too lenient. As for the performance of the detector presented in this thesis, it does a decent job. Although it is too rigid in the way it applies its criteria, it gives a good impression of what can be done with CPSP in this field of research.

6. References

[1] Zhou, R., & Reiss, J. D. (2011). Music Onset Detection. In W. Wang (Ed.), *Machine Audition: Principles, Algorithms and Systems* (pp. 297-316). Hershey, PA: Information Science Reference. doi:10.4018/978-1-61520-919-4.ch012

[2] Bregman, A. S. (1990). *Auditory scene analysis: The perceptual organization of sound*. The Journal of the Acoustical Society of America, 95(2)

[3] Wang, A. (2006), *Communications of the ACM - music information retrieval*. Volume 49 issue 8, 44-48. doi:10.1145/1145287.1145312

[4] Andringa, T. C. (2002). *Continuity preserving signal processing*. PhD thesis, University of Groningen, Groningen.

[5] Andringa, T., van der Linden, R., Krijnders, D. (2009). Dares_g1 database. *In DARES: Database of Annotated Real Environmental Sounds.* Retrieved September 10, 2013, from http://www.daresounds.org/database.php.

7. Appendix

The details of what the used CPSP functions do and how the database was created will be explained here.

7.1 CPSP Functions

Startup & functionsTest

To explain all the matrices and commands available properly, these two commands must be explained first. The first one is used to initialize all variables and directory paths needed to start reading a sound file. The second command opens up a UI which asks us to choose certain requirements we want for our analysis. These requirements serve to create a struct, filled with variables that depend on the requirements chosen in the UI. This paper used *calcRS*, *calcEdB*, *initPAS*, *calcPAS_S*, *calcPAS_T*, *basicTexture* and *calcBG*. After selecting requirements and a sound file, the struct called '*D*' is created, which contains the information used by the commands below. What the '*D*' struct contains depends on what requirements are selected Before any other commands are explained, let's see what information from the struct was used by this thesis.

The Cochleogram

D.EdB is a matrix of the entire sound file, where the rows represent the channels of the cochlea (instead of frequencies) and the columns represent the length of the signal, 1 cell being 1/5 of a millisecond. Each cell has an energy level which stands for the strength of the signal at that particular channel at that specific time. Plotting this matrix creates the cochleograms like the ones in this thesis.

The tonal matrix

D.EPAS_S is a matrix of standard-deviations. This matrix says something about how the local shapes deviate from the expectations of noise. Each cell is described in terms of standard deviations, where the value given to the cell means how strongly it deviates from possibly being noise (e.g. a high value means a strong possibility of it *not* being noise). So for example, a value of two indicates that the associated property of that part of the cochleogram

corresponds to two standard deviations from the mean of noise. By using this matrix it is possible to weed most of the obvious noise out. This makes it a strong tool for finding the outlines of the information that is interesting, the outlines of possible music.

Xcorr
xcorr is a function that, given a matrix as input, returns a matrix with an estimation whether sequences of energy found on the x-axis correlate. It keeps track of structures found in the given channel and notes a high value if the structure at a certain time in the signal resembles an earlier noted structure. For detecting music this can be a valid tool for finding rhythmic structures in the signal, as the time between two peaks can be used to determine the amount of beats-per-minute. Finding the right channel to search for the rhythmic structures is crucial. The channels mainly chosen in this thesis ranged from 140 to 150. As mentioned in the conclusion, a dynamic way of selecting the right channel is a direction that is likely to be more fruitful.

MovAv
movAv is a helpful function. It requires a matrix with values and two uneven numbers to know how far it needs to look horizontally and vertically (e.g. making both variables five would make it look two cells in the direction of each adjacent cell). For each cell in a matrix it sums all the values found in the range of the variables and takes the mean of the sum. This leads to the strengthening of cells surrounded by high values and the weakening of cells surrounded by low values. This in return leads to stronger ridges and the weakening of small spots of noise in the signal, leading to an overall boost in smoothness. Doing this too often would almost always lead to ridges, so to prevent artefacts from occurring in the data this function should be used in moderation. The range used for this detector was 5.

LengthAreas
lengthAreas has an interesting functionality. It receives a matrix of values (possibly bound to a criterion), a dimension in which it must operate and a gapsize. It looks through the matrix using the direction given by the dimension (e.g. a two would make it evaluate each column per row) Each nonzero value found is replaced by 'a', the number of consecutive nonzero values. The gapsize tells *lengthAreas* how many zero values it may ignore per sequence before it finalizes the sequences and starts to search for another one. *LengthAreas* makes it possible to filter the signal for information relevant for music. For example, long strokes of energy in one frequency in the signal could point to prolonged musical tones. If prolonged sequences of tonal sounds are indicative of music, then this function will help in finding them. The gapsize used in this thesis was 50.

7.2 Database of mixed samples

Adobe Soundbooth
The exact effect used to create more realistic music samples is a reverb effect called 'small room'. The music samples were created by selecting a minute in each song. For all songs except the classical piece this was the first minute.
The exact songs used per music style are:

Modern Classical Piano:
Lucovico Einaudi – Nightbook

Electronic:
Daft Punk – Digital Love

Hard Rock:
Queens of the Ages – Go with the Flow

Piano Pop:
Coldplay – Clocks

Rap:
Kanye West (Feat. Syleena Johnson) – All Falls Down

The signal-to-noise ratio was not determined by a parameter sweep. In hindsight, that was a better option than used in this thesis. For this database it was done by listening to each sample and equalizing the volume of both the scenario and the music sample using Adobe Soundbooth. To make it sound realistic, the volume of the

music sample had to be decreased most of the times, depending on the volume of the scenario.