Final Report

Automated Analysis of Music Performance Style

Author: Natalie Machin
Student No: C1321728

Module: CM3203 40 Credits

Supervisor: Dr. Andrew C Jones

Moderator: Dr. Matthew Morgan

Abstract

There are a few major problems when it comes to the computational processing of music. One issue is the need to classify particular performance styles, another is looking at what quantitatively makes a piece of music 'good' or 'professional'.

Previous work on this matter has been done trying to align different versions of the same musical piece to account for differences in tempo or playing technique. This report focuses on the identification and extraction of particular musical features, such as dynamic range and pitch.

Algorithms were also created to manipulate those features that are extracted. These manipulations are used to build an experiment to identify which musical features are most important for an individual to perceive a piece of music as professional, beginning to tackle the issue of what makes a piece of music 'good'.

Acknowledgments

I've had many peoples support throughout this project, not just on an academic level. However my significant gratitude goes to my supervisor Dr. Andrew Jones, who without his guidance, teaching and advice my progress in this project would've been severely hindered.

Additionally, the code base of the work done previously on this project by various individuals provided a basis for my work, the progression of this project would not have been feasible without that, so my acknowledgments go out to those individuals.

Finally, the large amount of open and free scientific reports and data allowed for extended research into the topics of this project and created better understanding of techniques. So my thanks go to the writers and contributors to those.

Contents

1 Introduction

1.1 Motivation

A important problem in the computational processing of music is the need to identify and extract major features of a particular performance, and then analyse these features to classify a given piece of music.
These 'major features' mentioned can include things such as, tempo, pitch, variations in dynamics and timbre.

Can we then identify which of these features are relevant to identify a piece of music as 'good' or 'professional'. For example, is a consistency in tempo integral for a piece to be perceived as professional?

The analysis of these features could then potentially be applied to quantifiably identify a 'professional' piece of music and create an automatic classification system.

To begin to tackle these problems, we need to find reliable ways to extract these key features of musical pieces, such that a quantitative measure is created for objective comparison between different pieces.

Then research needs to be conducted into seeing what operations can be applied to the features, for example how they can be manipulated to decrease the quality of the piece. Or interpolating between two different versions of the same piece.

Futhermore, can we identify common feature traits in specific musicians to then be able to automatically identify a piece as that particular performer?

1.2 Original Aims

The original aims that were laid out at the beginning of the project are listed below.

- *Reliable extraction of performance features from a musical piece*

  In order to be able to successfully classify a piece of music, we need to know what characteristics the piece has. Being able to extract specific features (such as dynamic range, timbre etc) would be the first step in quantifiably identifying characteristics of a piece.

- *Investigation into which features are relevant to the perception of performance style*

This aim is looking to identify whether there are common features for a particular musician, i.e. features they regularly use that could be used to identify their particular performance style.

- *Investigate what operations can be applied to styles*

  Here we are looking to see in what ways certain styles can be manipulated, either to improve them or to cause a different classification of the performance style.

- *Research into quantifiably identifying a good musical performance*

  Using information gathered from the other aims, can we identify the most common features in professional musician's performances, and therefore use this information to build a professional vs amateur classifier?

1.3 Adaptation of Aims

As work progressed throughout the 12 weeks of the project, the aims were adapted slightly to fit the work that was eventually carried out.
This was partly due to the large scope of the original aims, and the way the work developed meant the aims could be directed to more specific ones.

- *Reliable extraction of performance features from a musical piece*

  This aim remains as it was in the original statement, looking at being able to extract the specific parts of the audio that make up the relevant feature (e.g. dynamic range, tempo etc).

- *Investigation and implementation of feature manipulation*

  Once specific features had been extracted, can these be manipulated within the audio, potentially to artificially generate a different style of performance or to increase/decrease the level of professionalism.

- *Investigation into which features are relevant to the perception of music professionalism*

  Can we distinguish which features, when manipulated, cause the biggest effect on the perception of professionalism?

2 Background

This project was built as a continuation from previous work undertaken by others, so in this chapter this work will be discussed along with the issues that were come across during the first few weeks.

2.1 Previous Works

The code base for the original work, which was implemented with MATLAB, was made freely available at the start of this project.

There were a few key areas that this work was focussing on:

Alignment of Musical Performances

When you have two different renditions of the same piece, the timing of specific notes may not end up being the same, due to variations in tempo from the performers adding more expression to the performance.

This causes a problem, because in order to be able to accurately compare two pieces, for example by their volume profile, the note onset times need to be aligned otherwise the comparison will not be valid.

Dynamic Time Warping

Dynamic Time Warping (DTW) was a technique developed originally for speech recognition, see Sakoe and Chiba (1978). However it can also be applied for many other uses, including to temporally align audio signals to reduce variances when comparison is undertaken.

DTW works by finding the lowest cost path between the points of two signals. Initially a cost map is generated for all the points between signal 1 and signal 2, and then the optimal path (in this case minimising cost) is computed to determine how the time axis of the comparison signal should be warped.

Figure 1 represents an example of a cost map, with the two signals on the left hand side and along the top, and each square represents the associated cost (distance) between those two elements of each signal.

Finding the alignment would then involve calculating the path with the smallest distance through the grid.

Then in order to find the warp path within a reasonable time complexity, a dynamic programming algorithm is used to find the minimum cost path.

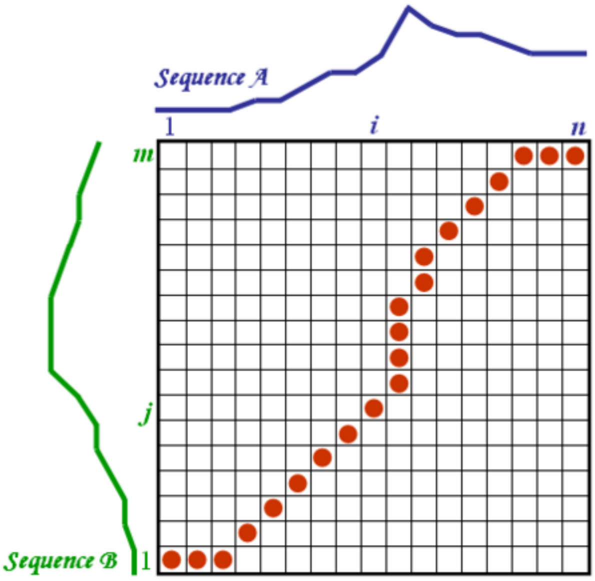


Figure 1: DTW cost map and minimum path example

Once the shortest path has been found in which to warp the signal to, a Phase Vocoder is used to manipulate the audio.

The concept of a Phase Vocoder (PV) was originally introduced by Flanagan and Golden (1966), which was then subjected to many adaptations to reach the level they exist in today.

PV's are used to modify either the frequency or time scale of a signal, without affecting the other. For example, usually if the time domain was compressed (audio speed increased) this would also increase the frequency (shift the pitch to a higher note). However with a PV you will not get the secondary effect.

In this particular usage of a PV, the warped audio is having its time domain stretched/compressed in order to match that of the second signal, and this is being done without affecting its frequency (pitch).

Feature Extraction

MIRToolbox

The MIRToolbox is a toolbox that can be added into MATLAB, with a variety of functions for extracting musical features from audio signals.

The toolbox was designed and created by Lartillot and Toiviainen (2007) and has an extensive list of functionalities for features extraction, as shown in Figure 2.

While the functionality available is long, the actual code behind the toolbox is fairly closed off, with it implementing their own unique data structures rather than utilising MATLAB's standard ones. This means easy adaptation of their feature extraction methods is not possible, and would require further research.

Usage of this toolbox should be investigated further for any future work, but the main function utilised for this project was the *mironsets* function. This produced an envelope of the audio signal, and identified the times and amplitudes of the notes within the piece.

The toolbox also contains a function called *mirgetdata* which outputs the generated data into standard MATLAB data structures for your own further use.

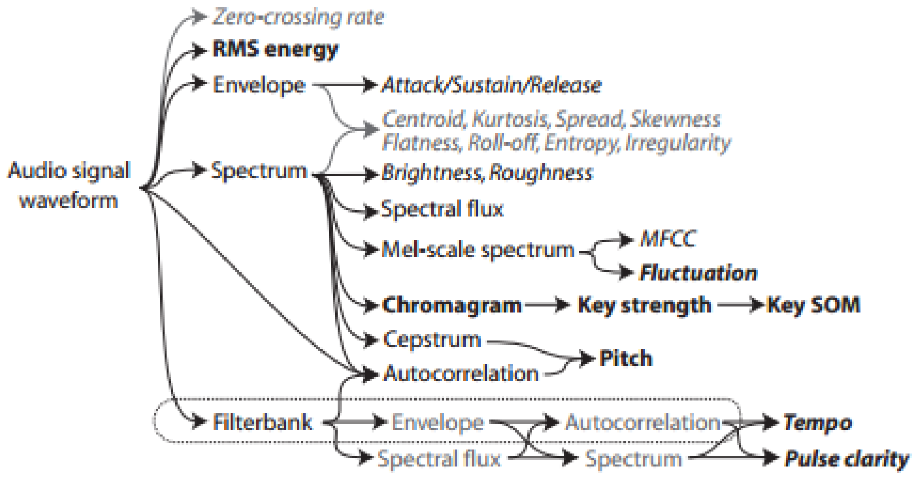


Figure 2: Summary of the available functionality within the MIRToolbox

Music Classification and Manipulation

While there is limited research into what quantitatively makes a piece of music good or professional, there is extensive work looking into the manipulation of music and the classification of genre.

Silla Jr, Koerich and Kaestner (2008) took a machine learning approach to try and automatically classify music genre. This used a pattern recognition approach whilst also splitting the signal into time segments, which produced better experimental results that usually produced from global segment classifiers.

Similarly and more complex, Tzanetakis (2002) wrote his thesis on the 'Manipulation, Analysis and Retrieval Systems for Audio Signals'. This focused on not only classifying music by genre (and also sub genre) but also providing ways of manipulating the music.

One example usage given within his thesis is the idea of being able to filter out specific melodies from a piece of audio. Explained by the scenario of being given a 24-hour recording of a site with birds, and applying a filter in order to identify the bird song (and therefore presence) of any endangered birds.

The ways in which music genre is classified with pattern recognition could potentially be adapted for the classification of professional music performance, although the performance style of different musicians would most likely vary as they add expression and improvisation to the piece, meaning there may not be a common pattern among professionals.

The manipulation of music however may prove useful in future work when looking at different techniques for manipulating specific musical features.

2.2 Issues

Alignment Function

Once access was given to the code base, the first task was to look through the existing functions to try and run them.
As the initial aim of the project required the alignment of two different pieces, it made sense to try and run the existing alignment function to see what starting point the project had.

However, to begin with the function had a few bugs within (potentially due to different execution environments) so these needed to be addressed and fixed before the program could run without throwing an errors.

After getting the code to successfully run, there were still issues though. Although the 2 pieces had been put through the DTW function, a large percentage of the tests did not end up resembling each other, meaning the time scale had not been correctly warped to match one another.

This suggested an issue with the actual DTW and PV code, therefore more research into its execution would need to be done during the project, if the original aims were to be tackled.

Processing Power

Running a full piece of music (roughly 3.5 minutes) through the DTW function would require a large amount of processing power, especially as MATLAB can be a bit slow in execution times.

Attempts to execute the code with the full piece caused the program to freeze and eventually the computer to crash.

This issue was tackled by just using 20 - 30 second extracts of each piece which was able to execute without crashing.
However this method led to another issue that needed to be taken into consideration.

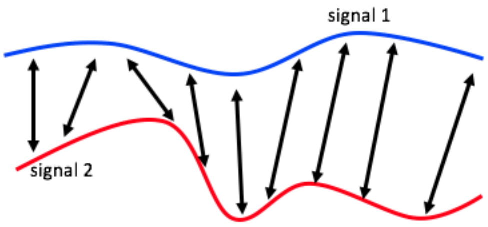


Figure 3: Note/beat aligning during DTW

Number of notes in audio segment

When using the method of extracting a small segment from each of the pieces to align to one another, these segments need to be the exact same section. Meaning they both need to have the exact same number of notes and for these notes to be the same.

The reason for this is that, dependent on how the DTW function is working (beat detection or note detection etc), the piece that is being warped needs a corresponding note/beat in the second piece for its time domain to be warped to match.

If, for example, the piece being warped had one extra note, there would be nothing in the second piece for that note to be matched to, and would therefore create an irregularity within the audio.

The way in which notes/beats are matched to one another during warping is demonstrated simplistically in Figure 3. The lines represents the two audio signals and the arrows are pointing to the individual beats/notes that are matched to one another.

2.3 Audio Data Sets

Within the code base that was handed over, was a large collection of audio data that could be used for this project.

As one of the motivating aims of this project was to be able to automatically classify a piece of music as professional or amateur, the audio data sets contained a large number of different renditions of the same piece, with some of these being by professionals and some by amateurs.

Two primary pieces were used during this projects research and implementation:

- Fur Elise - Ludwig van Beethoven
- Cello Suite No.1 Prelude - Johann Sebastian Bach

The initial research for the DTW alignment function was conducted using Fur Elise, as the beat detection to find alignment points within the piece seemed to work more accurately on the percussive style of the piano rather than for the stringed instrument.

However, the Cello Suite piece was then used for the eventual feature manipulation within the experiment, as timbre is more relevant in a stringed instrument, and the variations in dynamics would not need to be perfectly matched to note onset times.

3 Experimental Design

3.1 Aims Progression

At the beginning of the project, how the aims were going to be tackled needed to be considered before starting any work.

The first aim, that involved extraction of performance features, was fairly self explanatory. It would involve identifying the parts of the audio that made up that particular feature, and then researching techniques that would target those parts.

Moving further into the aims was when things started to be adapted slightly. Creating a work plan for the proceeding aims required the audio alignment via DTW for comparing and classifying of musical pieces. However as the attempts to get this to run reliably and consistently were not successful, this meant that the project would have to start to take a new direction.

Whilst trying to find a new focus for the project, a few questions were asked, such as what exactly makes a piece of music 'good', is it the technical and statistical aspects, or is it how it is perceived by humans? If it is how it is perceived by humans, are there specific aspects of the music that are most important to that perception?

These questions lead to the idea of creating an experiment to see which features were most relevant to the perception of music professionalism.

3.2 Code Base Development

To identify the most important features with regard to perception of music quality, the idea was to implement ways to manipulate certain features (such as tempo, timbre etc) with the aim to reduce their quality.

Participants could then judge the audio with these manipulated features and give their opinion on how professional the piece sounded. Therefore identifying the participants perception of the piece.

We would then be able to see which feature, when its quality is reduced, causes the biggest drop in professionalism rating. And then from this we can infer that that feature has more significance to the perception of professional quality of music compared to the others.

In order to create these modified recordings for the experiment, a toolkit of operations that would manipulate certain features of the audio needed to be created. This created a starting point for the software development process, as the eventual aim was to have a working, reliable manipulation function for each feature.

This implementation can be seen within the code base, as each individual manipulation function was developed and then eventual full algorithm that applied the manipulation periodically to the audio.

4 System Overview

The goals of this project focused primarily on the individual tasks that needed to be undertaken to prepare for the experiments and then execute them. Because of this, development of an integrated system was not a major priority.

The functionality created within the code base can be split into 4 sections. These sections all worked individually but simultaneously contributed to working to fulfil the project aims.

4.1 Audio Alignment

Although the audio alignment was not part of the eventual adapted aims, a large portion of the time spent on the project was on this and editing the existing DTW function to try to get it to work accurately.

The general structure of the existing DTW code is shown in Figure 4.

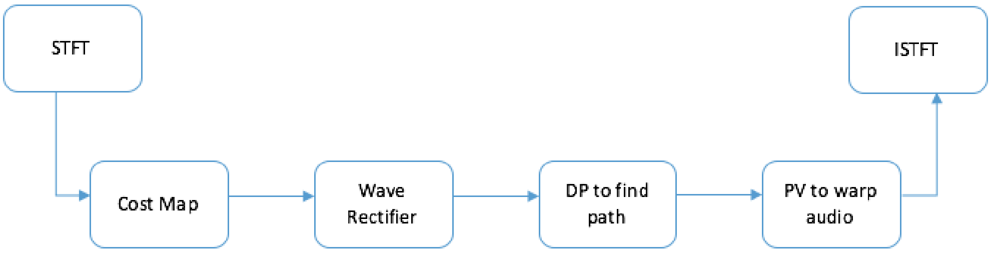


Figure 4: Diagram to show to major components of the DTW function

STFT: Short Time Fourier Transform. As the signal is originally saved as its amplitude values, the STFT decomposes the signal into the frequencies that make it up. This is calculated for sections of the signal as it changes over time.
Cost Map: This generates the distance between all points of a signal to all points of the second signal.
Wave Rectifier: An audio signal contains positive and negative energy values, so a half-wave rectifier is applied so that only positive energy values are left for use in the dynamic programming.
Dynamic Programming: Using the euclidean distances, the dynamic programming (DP) function is used to find the minimum cost path through the cost map matrix.
Phase vocoder: This is then used to warp the audio signal to match to the second signal, according to the minimum cost path that was found

using DP.
ISTFT: Inverse Short Time Fourier Transform. This converts the signal back into its amplitude values to give the original signal.

However, despite various editing and adaptation of the existing code, the alignment function was never in a full reliable working state.

Time Stretching

Initial issues revolved around the use of two different renditions of the same piece, and the fact that there may be more notes in one of the pieces (an issue explained earlier in this report).

To try and tackle this issue, the same audio segment would be used as the 2 signals in the DTW function, however one version of the audio had been artificially stretched (decreasing the tempo). This ensured that the two signals had the exact same number of notes/beats.

This was implemented by using code for a phase vocoder found online, and making slight adjustments to fit the projects needs and to account for a few errors.

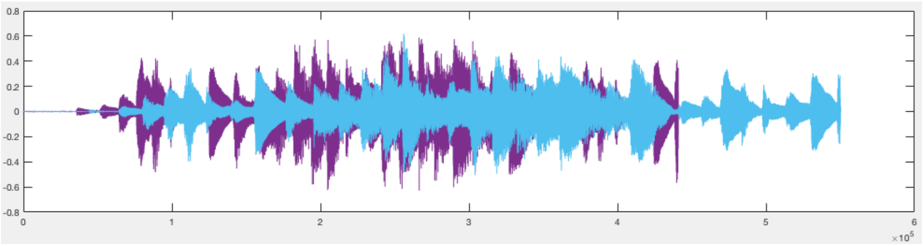


Figure 5: Plot of two of the same audio signals but with one being stretched by a factor of 0.8

The result of running the time stretching function on a piece of audio is shown in Figure 5, with the blue signal's tempo being slowed down by 0.8 of the original.

Unfortunately, after running DTW with these stretched signals, the function was still not successful at warping the stretched audio back into the original.

This suggested that the way in which the DTW function was identifying different beats or notes was not working accurately, as they were the same piece so the same number of notes should have been detected.

Beat Detection

After considerable testing, it was observed that the function was much more effective on pieces with a clear beat or clear note onset times. For example, it was much more effective on percussive instruments which have an obvious start to a note as opposed to a stringed instrument who's note onset time is not so exact.

4.2 Feature Extraction

After discovering the MIRToolbox library during the initial project research, a couple of feature extraction functionalities were found within that may prove useful in future work.

Note Onset Times

The function *mironsets* would create an envelope of the signal, and identify the time and amplitude of each note within the piece.

However the accuracy of this function purely depended on how clear the notes within the music were, so for example, it would be much more effective on a percussive instrument rather than a stringed instrument.

The code to generate the MIR note onset times is shown in Algorithm 1 and the output plot is shown in Figure 6.

Algorithm 1 Generating MIR note onsets envelope

```
[audio, sr] = audioread('audio.wav');
audioMIR = miraudio(audio);
audioOnsets = mironsets(audioMIR);
```

Dynamic Variation

Although the MIRToolbox was a closed off piece of software, creating its own data structures, it also offered a function to output the data it generated into a standard MATLAB data structure.

```
[noteTimes, noteAmps] = mirgetdata(audioOnsets);
```

Here the note times and amplitudes are saved into the MATLAB matrix. As the amplitudes represent the loudness of each note, this information could then be used to identify variations in the dynamics.

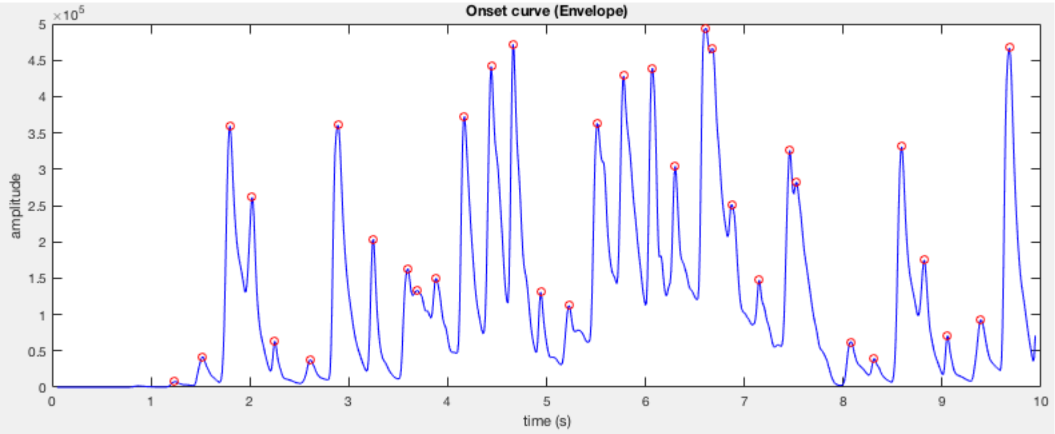


Figure 6: MIR Note Onset Envelope for a 10 second clip of Fur Elise

4.3 Audio Editing

During the course of this project there were various simple edits that needed to be made to the audio files in order to facilitate functions or to prepare the audio segments for the eventual experiment.

Converting from Stereo to Mono Audio

Some of the audio files within the code base were in stereo format (played out of 2 channels), however some of the functions (including the MIRToolbox) required the audio to be in mono format (only 1 channel). So a function needed to be written to do this.

Algorithm 2 Converting from stereo to mono

```
function STEREO2MONO(audio)
    audioChannels = audio(column1) + audio(column2);
    mono = audioChannels / 2;
    return mono;
end function
```

Algorithm 2 shows the procedure for converting stereo to mono, the columns within the audio matrix represent the 2 channels. Each corresponding value is added together and the average is taken to output a matrix with only one column.

Extracting a segment of the audio

As mentioned before, the amount of processing power proved to be an issue, so the functions needed to be ran on shorter clips of the audio rather than the whole piece.

Algorithm 3 shows the pseudocode to extract a specific section of the audio signal, with startTime and endTime being the positions in the original audio that the clip is created from.

Algorithm 3 Pseudocode for extracting a segment from a piece of audio

```
function PIECESEGMENT(audio, sr, startTime, endTime)
    totalSamples = length(audio);
    totalDuration = totalSamples/sr;

    t = time vector of totalSamples evenly spaced points
    between 0 and totalDuration;

    segment = get data between startTime and endTime using t;

    return segment
end function
```

Audio Fade In/Fade Out

For the segments of audio files used within the experiment, a fade in and out was applied to the clip as the audio would start in the middle of a particular note/bar in the music.

Having a fade in/out meant the start and end of the clip was not as harsh sounding and so would not affect the listeners perspective of how professional it sounded.

The general process for applying a fade in/out can be found in Algorithm 4, where *sr* is the sample rate of the audio, and *dur* is how many seconds the fade lasts for.

Algorithm 4 Pseudocode for applying a fade in/out to a piece of audio

```
function fadeInOut(audio, sr, dur)
    fadeSamples = dur * sr;
    fader = time vector of fadeSamples evenly spaced points
    between 0 and 1;

    fadeAudio = audio;
    fadeAudio(1:fadeSamples) = apply fade in to original audio;

    fadeAudio(end-fadeSamples) = apply fade out to original audio;

    return fadeAudio
end function
```

4.4 Feature Manipulation

Throughout the project, various techniques were investigated and implemented in order to manipulate the particular features.

Some features had multiple techniques identified for manipulation, however some of these were more effective than others. However only one technique per feature was utilised for the eventual audio used within the experiment.

4.4.1 Dynamics

Amplification

One way of increasing the dynamic range is by raising the amplitude values, as the loudness of the audio is represented by the amplitude.

However, the risk of distortion needs to be taken into account before increasing the values too much. The highest value the amplitude can have before distortion occurs is 1, so all the values need to be scaled up by a specific factor, with the factor causing the original highest value to become a 1.

The idea behind using the amplification, is to apply it periodically along the audio signal. This artificially creates a much large range in the dynamics, as the volume will suddenly increase and then decrease again.

Dynamic Range Compression

Alternatively, this technique would be used to decrease the dynamic range.

A Dynamic Range Compressor (DRC), is a process that works to attenuate the volume of loud signals and either boost or leave quieter signals unchanged. With the purpose to reduce the dynamic range of the signal.

This has various applications, such as to make the music louder without exceeding an amplitude limit. By compressing the highest peaks of the signal, the overall volume can then be increased without causing distortion.

Figure 7 shows both the original audio (yellow) and compressed audio (blue) after applying a DRC to it. You can see the higher amplitude peaks of the yellow signal have now been reduced to a lower value. The blue signal now has a much lower dynamic range as it mostly is all staying at the same amplitude.

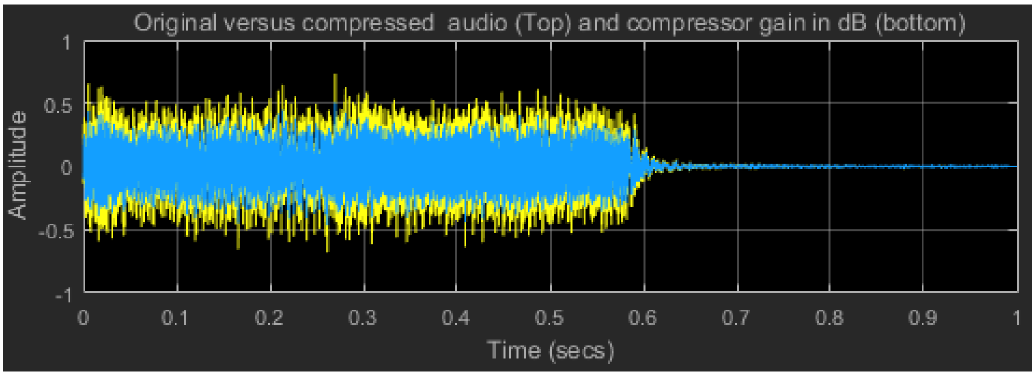


Figure 7: Original and compressed audio after using a DRC

This particular manipulation of the dynamics is relevant to this project, as a key feature of a professional music performance would be the variations in dynamics. Crescendo's and diminuendo's are included in a piece to add expression and emotion to the music, whereas a piece played all at the same volume would be somewhat blander.

By applying a DRC to the piece of music, we are removing the dynamic variation and potentially reducing the quality of the performance, therefore making it seem less professional.

4.4.2 Tempo

Phase Vocoder

Previously in this report the functionality of a Phase Vocoder (PV) was discussed, where it can be used to modify the time scale of a signal without affecting the pitch.

This was the singular way of manipulating the tempo, either to increase or decrease it. However it seemed more accurate to have a decreased tempo to reduce the level of professionalism, as an amateur performer may play the piece slightly slowly in areas they are not confident in.

4.4.3 Timbre

Bandstop Filter

One idea to manipulate the timbre of the audio was to boost or attenuate certain frequencies in the signal. If, for example, the bass frequencies were reduced, this might create a lesser quality tone.

A bandstop filter is used to attenuate frequencies within a specified range and leave all others untouched. This can be used to filter out a specific frequency that is thought to have a contribution to the overall tone quality.

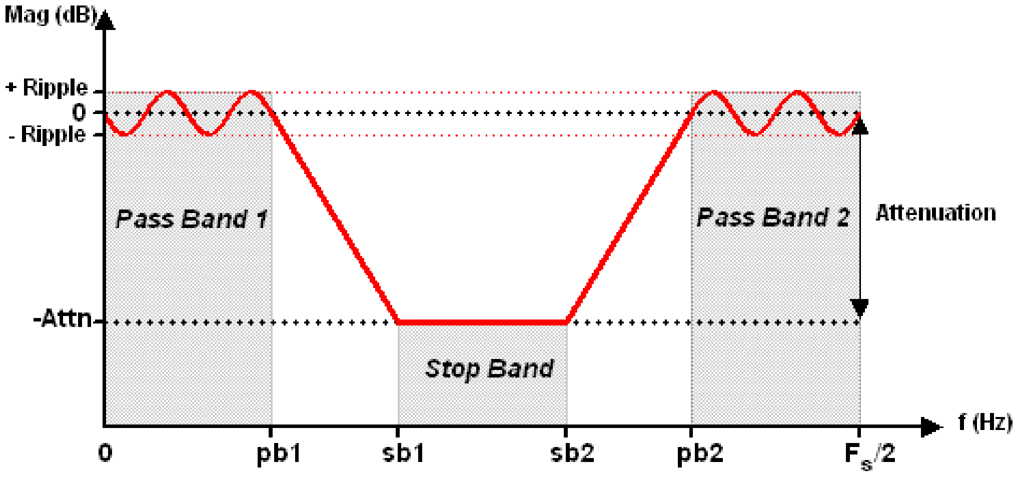


Figure 8: Depicting the design of a bandstop filter

Shown in Figure 8 is the general workings of the filter, with any frequencies within the Stop Band having the magnitude (in dB) reduced. So the attenuation is not abrupt, it is done gradually starting from the ending/starting frequency of each Pass Band.

Vibrato

Vibrato is defined as a pulsating effect produced in music by tiny, fast variations in pitch. This effect can be used moderation by performers to add expression and warmth to a piece.

Although vibrato is perceived and implemented as a pitch variation, it is recognised as a characteristic of timbre and as an integral part of tone.

As this effect is supposed to be used in moderation to add more expression, one idea for manipulation of the audio is the create an excessive vibrato effect. This would disrupt the tone's quality at the same time as causing some discrepancies in pitch.

4.4.4 Pitch

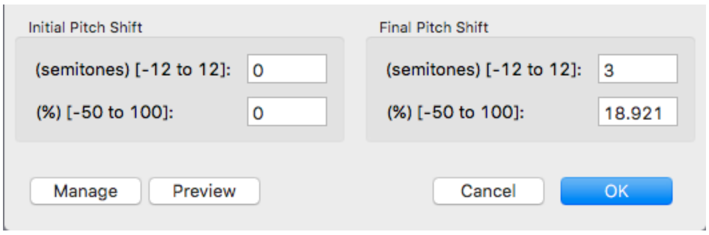
Phase Vocover

Initially, the idea for altering the pitch was to utilise the Phase Vocoder (PV) already mentioned earlier in this report. With this, the tempo could be increased/decreased without affecting the pitch, then by resampling the audio to the correct length this would then alter the pitch as required.

However, applying a pitch shift to the entire piece was not a feasible way of manipulating the level of professionalism. As a piece can be played in a different key for preference sake, this does not show a reflection on the quality of the performance.

Another idea was to apply the pitch shift to random notes through the piece, utilising the note onset times gathered from the *MIRToolbox* function *mironsets*.

Unfortunately, this just resulted in the piece sounding like there were errors in the recording of the piece, rather than errors made by the player, so this method of manipulation was not continued.



Initial Pitch Shift
(semitones) [-12 to 12]: 0
(%) [-50 to 100]: 0

Final Pitch Shift
(semitones) [-12 to 12]: 3
(%) [-50 to 100]: 18.921

Figure 9: Audacity's Sliding Pitch Shift process

Sliding Pitch Shift

Throughout this project, other ways of manipulating the features, other than creating functions in MATLAB, were continually investigated. This seemed to make more sense than recreating functionality that already exists.

Audacity is a popular platform for editing and recording audio. With various features for modifying sound within it, one of these features including a Sliding Pitch Shift.

The pitch shift functionality can be used to slowly change the pitch of the audio over the course of the sound file.

Shown in Figure 9, you can select how many semitones you want the pitch to be eventually shifted by the time it reaches the end of the piece.

5 Implementation

5.1 Overview

The programming language used for this project was MATLAB for a number of reasons. Firstly, the existing code base from previous work was all written in MATLAB, so remaining in this environment made development a lot easier.

Also, one of the key elements of MATLAB is that it is used for signal processing, making it the ideal environment for manipulating and displaying audio signals.

In addition to the MATLAB code, in the existing Dynamic Time Warp (DTW) function, a dynamic programming C routine was used. In order for this to execute within the MATLAB environment it needed to be configured for *mex* support.

Mex files allows for C or C++ subroutines to be called from the MATLAB command line the same was that a built-in function would be.

The programming style used was not one to aim for a cohesive system. Each function was mostly designed to worked independently for its own specific purpose. Although slightly more complicated functions would call subroutines for the more intricate parts of the process.

This mostly meant that the system was loosely coupled, as a change in one function would not cause undesired changes in another function.

5.2 Audio Editing

As described in the System Overview section, various simple functions were created to edit the audio files slightly to adapt them for their intended use.

The conversion from stereo to mono audio, and the extraction of a small clips from the whole audio were fairly simple to implement and can be found in the complete source code.

Fade In/Out

The fade was made by creating a matrix with the amount of fade samples points, with values from 0 to 1 (called a time vector). The time vector was then multiplied against the amplitude values of the original audio samples to create a fade in effect.

For example, the first value of the time vector was 0, therefore the first sample of the original audio was multiplied by 0, causing the amplitude to then be zero, therefore reducing the volume to nothing. As these vector values increased to 1, the amplitude values increased, giving the fade effect.

The same principle for the fade out was used, however the time vector was just reversed.

Listing 1: Code for the fade in and out function

```
function [fadeAudio] = fadeInOut(audio, Fs, dur)
% Fade in and out - create an in/out fade for audio
% The argument 'dur' is how many seconds the fade lasts

fadeSamples = round(dur.*Fs);
fader = linspace(0,1,fadeSamples)'; % Creater fader

fadeAudio = audio;

% Apply fader to start
fadeAudio(1:fadeSamples) = audio(1:fadeSamples).*fader;

% Apply fader to end
fadeAudio(end-fadeSamples+1:end)
        = audio(end-fadeSamples+1:end).*fader(end:-1:1);

end
```

5.3 Feature Manipulation

5.3.1 Manipulation for the Experiment

A general algorithm for preparing the audio for the experiment needed to be created. The idea was to apply manipulation in 3.5 second chunks, periodically across the audio.

The reasoning behind this was because if the manipulation was applied to the whole piece, the discrepancies may not be noticed due to the consistency in the audio, therefore invalidating the experiment. Having inconsistencies in the audio makes the decrease in professionalism more evident meaning we are more likely to get reliable results.

Listing 2: General code for audio manipulation

```
function [audio] = manipulationName(audio)
% Function periodically manipulates clips of the audio

N = length(audio);

x = 1;
startTime = 5;

while x < 3 % Loop 2 times

  endTime = startTime + 3.5; % 3.5 second clip
  seg = pieceSegment(audio,sr,startTime,endTime);
  seg = manipulation(seg);

  sample = startTime * sr; % Get sample value of start
  audio(sample:sample + length(seg)-1) = seg;
  % Replace original section of piece with new version

  x = x + 1;
  startTime = startTime + 6;
  % Manipulate in 6 second intervals

end
```

Listing 2 outlines the general code for preparing the experiment audio. As this project was working with 20 second audio clips, looping 2 times seemed appropriate, but this would be changed for longer pieces of audio.

The algorithm works by extracting the 3.5 second clip, manipulating it, and then placing it back in the audio. And then repeating this 6 seconds later in the piece.

5.3.2 Dynamics

There were two investigated techniques for manipulating the dynamics; the Dynamic Range Compressor (DRC) which decreased the amount of dynamic range, and amplification which would increase the dynamics range.

The eventual technique used was the amplification, as although the DRC was quantitatively effective at reducing the dynamic range, there was not much audible difference, which was a large requirement for the design of the experiment.

As outlined in the System Overview section, the amplification worked by scaling all amplitude values up by a certain value, such that the original highest value then equalled 1. This ensured that the amplification was applied but no distortion occurred.

Figure 10 shows the result of the amplification being periodically applied to create the experiment audio. The blue signal is the original audio, with the purple being the manipulated version. We can clearly see where the amplitude values have been increased as a result of amplification.

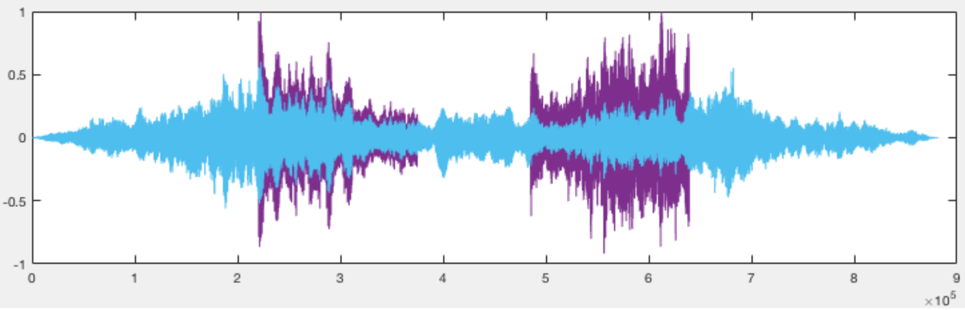


Figure 10: Difference shown after applying dynamics manipulation

5.3.3 Timbre

While the BandStop filter was effective at filtering out specific frequencies, applying this created an effect of a reduced recording quality, rather than a reduced performance quality.

Because of this, it was decided to use the vibrato function for Timbre manipulation. The code for this was reused from an external source so can be found within the *Libraries* section of the source code.

The two main parameters for the vibrato function are the *modulation frequency (MF)* and the *modulation width (MW)*.

The MF is in KHz, and apples to the rate at which the signal is varied. So the amount of KHz the frequency is modulated by during the vibrato. The MW is how long each modulation should last in seconds.

In the eventual Timbre manipulation the values MF = 3 and MW = 0.0008 were used, to signify a frequency of 3KHz and 0.0008 seconds.

5.3.4 Tempo

The singular technique investigated for manipulating the Tempo was the Phase Vocoder (VC), which can be used to increase or decrease the tempo of a piece of audio without affecting the pitch.

The code for this can be found within the *Libraries* as it was reused from an external source.

However, the general algorithm for preparing the audio for the experiment had to be adapted for the tempo function. This was because, as a clip was extracted from the audio, and then changed to a slower tempo, this also stretched the clip meaning it was a little longer than it originally was.

This then meant that when the slightly longer clip was to be placed back in the original audio, the space it was inserted into was not quite big enough, causing a jump in the eventual audio where it skipped over several thousand samples.

To tackle this issue, before placing the stretched audio back in, all proceeding samples were moved along in the matrix by the new added length of the stretched clip. The edited code for this can be found in Listing 3.

Listing 3: Extract from Tempo manipulation function

```
endTime = startTime + 3.5;
seg = pieceSegment(audio, sr, startTime, endTime);
stretch = pvoc(seg,0.8);
% Use phase vocoder to stretch audio by 0.8

% Move all proceeding vectors along by the added length
move = length(stretch) - length(seg);
% Loop from end of audio
for j=N:-1:endTime*sr
    audio(j+move) = audio(j);
    % Making samples in front equal to earlier samples
end

sample = startTime * sr; % Get sample value of start
audio(sample:sample + length(stretch)-1) = stretch;
% replace original section of audio with new version
```

5.3.5 Pitch

As described in the System Overview, the attempts at utilising the Phase Vocoder to manipulate individual notes did not produce effective results. More research and adaptation needed to be invested into this to get more promising results, however due to the limited time span of the project, other means of manipulation were used.

This led to using Audacity's *Sliding Pitch Shift* functionality, where you can choose how many semitones the pitch should be shifted by over the course of the audio.

For the creation of audio clips for usage in the experiment, a pitch shift of 3 semitones was implemented.

5.4 External Code and Libraries

Table 1 shows a list of Libraries and how they were used throughout the project, included in this is all the code that was reused from external sources.

Library	Source	Licence	Usage
MIRToolbox	University of Jyväskylä	GNU General Public License	Mironsets used to collect the note onset times and their individual amplitudes
Phase Vocoder - pvoc	D. P. W. Ellis - Columbia	n/a	Used for manipulating the tempo, and also utilised in the Dynamics Time Warping function
Vibrato	Prof. Dave Marshall	n/a	Used for timbre manipulation
Phase Vocoder - audioStretch	5/2005 Bill Sethares	n/a	Original phase vocoder used before more efficient one above was found
DTW - dtwWarp	D. Ellis (2003) Columbia	n/a	Another implementation of DTW found when looking at ways to improve the DTW version in the existing code base

6 Experiment

6.1 Introduction

If we are to be able to successfully classify a professional music piece over an amateur one, we need to know what particular features of music contribute the most to the perception of a professional performance.

The purpose of this experiment is to try and identify what features are most important for humans to perceive a piece of music as professional. In other words, is there a specific feature (or features) that, when edited, causes the piece to seem less professional.

This experiment will look into 4 different features, dynamics, tempo, timbre and pitch, and whether the manipulation of each of these decreases the rating of professionalism.

Additionally, the musical experience of the participant will be recorded (i.e. beginner vs experienced musician), along with their reasons for rating each piece as less than maximum value.

6.2 Problem definition

What musical features are most significant with regard to the professionalism of a piece of music's performance?

While this is the official problem definition, a different title was used for the distributed questionnaire. This was so the purpose of the experiment was not given away to the participants, as this may have influenced their responses.

Experiment title: *Research into human perception of professional music performances*

6.3 Objectives

- To see if the individual manipulation of dynamics, tempo, timbre and pitch affects the perception of professionalism
- To see what the differences in perception are between people of different musical experience.
- To find out how accurately people identify what they think is wrong with the piece

6.4 Variables

Independent Variable: The feature manipulated within each piece of music, i.e. the dynamics, pitch, tempo and timbre.
Dependent Variable: The participant's perception of the professionalism of each piece. Given as a rating out of 10 (with 10 being the highest level of professionalism).

Additionally, a control measure of an audio piece with nothing altered will be included within the questionnaire to compare the participants responses against.

6.5 Hypotheses

Research Hypotheses:

- The individual manipulation of a musical feature will cause a decrease in the professionalism rating
- People with higher musical experience will be able to identify more accurately what is wrong with the piece.

Null Hypotheses:

- The manipulation of tempo will not cause a decrease in the rating of professionalism.
- The manipulation of dynamics will not cause a decrease in the rating of professionalism.
- The manipulation of timbre will not cause a decrease in the rating of professionalism.
- The manipulation of pitch will not cause a decrease in the rating of professionalism.
- People with more music experience will not be more accurate at predicting what's wrong edited in the peice.

6.6 Sample

The sample used for this experiment was a combination of a self-selecting sample and an opportunistic sample.

To obtain the self-selecting portion of the sample, the questionnaire was advertised to the available public and then filled out by any willing volunteers who came across it.

As the experiment requires a number of participants with high musical experience, the opportunistic method was used to obtain these. Using a contact in the Music department of the University, the questionnaire was distributed amongst these music students to be filled out.

6.7 Ethics

When conducting research using music, the licensing and copyright requirements must be taken into account to ensure no infringement has occurred.

Fortunately, the piece used for this experiment (*Cello Suite No.1 Prelude by Johann Sebastian Bach*) is within the Public Domain, meaning all rights for the music have expired and therefore no license is required.

6.8 Procedure

6.8.1 Questionnaire Preparation

Before data could be collected, the questionnaire needed to be prepared for distribution.

The audio files used in the questionnaire were 20 second clips from two different versions of Cello Suite No.1 Prelude. A control version for each was included, with no manipulation, and then a individual version of each with the specific feature manipulations (dynamics, tempo, timbre and pitch). This produced a total of 10 audio clips, 5 for each rendition.

For the feature manipulation, as outlined in the Implementation section of this report, this was done periodically on small sections of the audio clip (rather than the whole piece).

Additionally, the audio files needed to be randomly ordered so the same manipulation was not played twice in a row, and also the different renditions were alternated so the same piece wasn't playing twice in a row.

6.8.2 Questionnaire Construction

The questionnaire was constructed using a free online survey building tool, *TypeForm*. The main feature needed when selecting a online survey tool was the ability to embed audio clips/videos into the questionnaire.

The majority of online tools required you to pay for this service, which was why *TypeForm* was selected. It allowed the embedding of YouTube videos, so the audio clips were uploaded to YouTube as a video to then be placed within the survey.

The initial question of the survey was to determine the musical experience of the participant. This would be interesting in later analysis to see if a particular feature was more important to beginners compared to professionals.

The rest of the questionnaire followed the same structure, the audio clip was presented to be listened to, the participant then had to rate how professional they thought the piece was out of 10 (with 10 being the most professional). The follow up question was then for the participant to comment on what they thought was wrong with the piece. This allowed for some qualitative as well as quantitative feedback.

6.8.3 Questionnaire Distribution

The main form of distribution of the experiment was via social media. This allowed for the self-selecting portion of the sample as anyone who came across the survey would then need to decide themselves to complete it.

However the survey was also targeted at Music students, via the Cardiff University Music Facebook page, with the aim of obtaining an array of experienced vs non-experienced musicians.

Additionally, the questionnaire was shared with staff within the School of Computer Science to be passed on to others.

6.9 Results

From the participant responses, the average rating of professionalism was calculated for each audio clip.

First the average rating was split into the people that labelled themselves beginner, moderate and experienced with regard to musical experience (the spread of each category in the responding sample can be seen in Table 1). Then the average rating was also calculated for all participants.

Table 2 shows the average ratings for each piece from each group of people and also from all the participants. The audio clips are described by which rendition it was (1 or 2) and what its manipulation is, with the version

Category	**Percentage of sample**
Beginner	17%
Moderate	50%
Experienced	33%

Table 1: Showing the spread of participants in the sample

with no manipulation being called 'Control'.

	Beginner	**Moderate**	**Experienced**	**All Avg**
Control 1	8.67	8.67	8.83	8.72
Tempo 1	8.33	6.67	7.50	7.22
Timbre 1	6.33	8.00	6.17	7.11
Dynamics 1	6.67	7.56	7.50	7.39
Pitch 1	6.33	5.33	4.00	5.06
Control 2	9.00	8.67	7.33	8.28
Tempo 2	6.00	6.67	6.00	6.33
Timbre 2	7.00	6.67	6.17	6.65
Dynamics 2	5.33	5.56	6.33	5.78
Pitch 2	5.67	5.67	3.50	4.94

Table 2: Showing the average rating of each piece from the participants

The other data collected from the experiment was from asking the participants to comment on what they thought was wrong with each piece. This was interesting to note whether the listeners were marking down the professionalism of the piece due to the correct reasons or not.

Table 3 depicts the percentage of each group (and all participants) who correctly identified what was wrong with the piece. In the case of the controls, this refers to not incorrectly identifying a problem.

As the question where listeners gave their comments was an open question, this meant that there was some comments which were not relevant to what we were investigating.

Because of this, any comments that were not primarily to do with the quality of the audio alone have been ignored. One example of these ignored comments is, 'I do not know how the performer is presenting themselves on stage'. And also comments criticising the complexity of the piece were also ignored.

	Beginner	**Moderate**	**Experienced**	**All**
Control 1	66%	100%	83%	89%
Tempo 1	33%	66%	50%	55%
Timbre 1	0%	33%	50%	33%
Dynamics 1	33%	44%	33%	39%
Pitch 1	33%	89%	100%	83%
Control 2	66%	78%	83%	78%
Tempo 2	33%	78%	83%	72%
Timbre 2	33%	66%	66%	55%
Dynamics 2	66%	55%	50%	55%
Pitch 2	33%	69%	68%	63%
AVERAGE	40%	69%	68%	63%

Table 3: Percentage of participants who correctly identified the errors within the piece

6.10 Quantitative Analysis

6.10.1 Beginners

The beginners did not seem to have a specific feature that their rating of professionalism was affected by. They correctly gave the highest rating to the Control pieces, and then all of the feature manipulations (bar Tempo 1) produced a similar number, with Dynamics 2 being the lowest.

This mirrors their results for estimating what was wrong with the pieces. At an average of 40% the beginners weren't hugely accurate, which is understandable due to their lack of musical knowledge. However their standout piece, where 66% successfully identified the correct problem was the Dynamics 2 piece, the same that they rated as the lowest.

This could suggest that listeners are more likely to give a piece a lower rating if they are more confident with knowing why it isn't as professional.

6.10.2 Moderate and Experienced

The people with moderate musical knowledge and the experienced musicians had very similar outcomes, with their rating of professionalism being most reduced by the manipulation of pitch whilst also having a high percentage of them correctly identifying that it was in fact the pitch that was off.

Additionally, there was a large percentage of the moderate and experienced participants who identified the 2nd tempo manipulation.

Although the moderate and experienced participants had similar outcomes, the experienced musicians were much more critical of the change in pitch, giving very low scores of 4.00 and 3.50 for both Pitch manipulations (by far the lowest) whereas the moderate listeners gave scores of 5.33 and 5.67 (still one of their lowest results).

However, the Dynamics 2 piece was actually rated the lowest out of all pieces by the moderate group, which mirrors that of the beginners. We question the reliability of this results however, due to the lack of significant decrease in the first dynamics piece.

6.10.3 Overall

The results overall show that pitch is one of the biggest contributors to the perception of professionalism, along with being one of the easier feature manipulations to identify.

However all of the feature manipulations caused some decrease in the average rating, as shown by the original rating for the control pieces. More research and experiments would need to be conducted to determine the statistical significance of this.

The dynamics of the second piece caused a large decrease in the rating, mainly for both the beginners and moderate listeners. But this was not consistent with the first dynamics manipulation score, the second piece may have been overly manipulated or the first piece not enough. More research with a much large array of audio data sets would need to be conducted to determine which of these was an anomalous result.

6.11 Qualitative Analysis

6.11.1 Piece Comparison

Although the two different pieces included in this experiment were performed by professionals, the initial rating (control) where nothing had been manipulated showed a slightly lower rating for the second piece overall. Additionally, the participants produced a lower accuracy rate for identifying what was wrong with the piece, (in the case of the control it was wrongly identifying something that wasn't there).

Subjectively analysing the two pieces, the second piece seemed to have a rather large decrescendo meaning some parts of the piece were very quiet.

While variations in dynamics are integral for adding emotions to the piece, a too dramatic drop in volume could affect the quality of the piece.

Additionally, the performer added a fair bit of rubato to the piece that could potentially come across as unsureness in their playing, and therefore could've lead to the participants rating it down.

6.11.2 Open Question Analysis

We have looked at the accurateness of the answers participants gave for what they thought was wrong with the piece, however it is interesting to look at the common themes identified and whether participants gave any comments that would suggest the validity of the experiment was compromised.

Common Themes

Rubato within a musical piece is the increasing or decreasing of tempo within a phrase, done without altering the overall pace or count of the music, to create expressiveness. This was a feature identified frequently across all the audio samples by the participants.

This may have been due to it already being present in the second version of piece, and then adding the tempo manipulation made it even more prominent. While creating expressiveness with rubato is a fairly professional technique, excessive use of it can decrease the quality of the piece and this was the general opinion amongst listeners.

However, a few people did not identify the tempo changes as rubato, but simply as an effect of the inexperienced performer. Comments to this effect were made, *'sounds like someone who's still learning who wants to slow down to concentrate on one part of the song they are unsure of'*.

Another major theme amongst the comments was to do with the pitch shifting within the audio. The poor intonation (the accuracy of pitch during playing) was mentioned by many participants for the pieces with manipulated pitch.

One listener commented on how the performance sounded professional but the change of key into the minor made the piece less enjoyable to listen to and therefore they marked it down.

Also relating to pitch, it was commented on that the instrument sounded like it was out of tune rather than the performer playing the wrong note.

Experiment Comments

As this was a blind experiment, participants were not supposed to know that parts of the audio had been manipulated. Although, whilst creating some of the actual manipulations it was difficult to make it sound completely natural.

This was identified by a fair few listeners, although only when a dynamics manipulation had been applied. Comments were recorded about the volume control being played with, rather than the performer executing the dynamic changes. Additionally, one participant repeatedly referred to it as 'artificial dynamics', showing that they knew it was not a natural change.

Participants recognising that some artificial manipulation had been done brings into question the validity of the experiment, or the dynamics part of it at least, as they know the decrease in quality is not due to the performer.

If a humans perception of professionalism factors in both the performer and the recording quality then this knowledge may not drastically affect the results. More research would need to be conducted to confirm the validity. Another frequent comment made was about the complexity of the

chosen piece of music. Listeners felt that the piece was very simple and part of the standard repertoire for a cellist, so it was hard to create a proper opinion about their skill as a musician.

For this reason the participants may have given slightly lower ratings for the piece, where they would've given higher values if the piece had been more complex. This could have affected the overall consistency of the results due to the comparison of ratings between people who were bothered by the choice of piece and people who weren't.

Recording vs Performance Quality

A few answers were given questioning whether the decrease in quality was due to the performer or the recording quality. This may have affected the results, depending on which quality the participant was basing their rating off.

One perfect example of this was a participant who left the comment, *'bad recording but good playing'* and gave a score 9, meaning that they ignored the fact that it was a bad recording. However the reliability of the results would be affected unless all participants chose to ignore the bad recording when giving their rating.

Another recurrent comments were about disturbances in the background of the recording, being unsure if it was the performer or an audience member.

One way to tackle this effect on reliability and consistency would be to specify in the question that any decrease in recording quality should be ignored, and the professionalism rating should be purely done based on the performers execution.

6.12 Conclusion

This experiment was a positive initial step into researching what musical features are most important for humans to perceive a piece of music as professional.

The experiment also successfully covered its objectives and produced some initial answers to the questions they were asking.

- Manipulation of particular features did cause a decrease in the perception of professionalism. With pitch being the most significant affecting factor.
- The experienced musicians appeared to be the most critical of faults in the piece, recording the lowest ratings. While the overall opinions were fairly similar across the different levels of experience.
    - Beginners seemed to be more likely to rate down pieces when they knew what was actually wrong with it
- The moderate and experienced participants were much more accurate with predicting what was wrong with the pieces compared to the beginners.

From the data gathered we can reject the null hypotheses and accept the research hypotheses.

As, the manipulation of features did cause a decrease in professionalism rating, however more testing would need to be done to obtain a statistically significant result.

Additionally, it was clear that the people with higher musical experience (moderate and experienced) were much more accurate at identifying what was wrong with each piece.

6.13 Future Work

This experiment gives relatively basic insight into the human perception of professionalism, and further research could be conducted to gain even more knowledge that can then be applied to future applications.

One way that this could be progressed is through running a similar experiment but with far more data sets and participants. So many more different renditions of the piece, all being manipulated in the same way, then to be tested on a much larger sample.

This way the results gained will be far more statistically significant and reliable, and a lot more knowledge of music perception will be able to be inferred.

Another development of this work could be to quantitatively identify when a piece stops being perceived as professional when manipulating a feature.

This would require specific quantitative values for the feature manipulation, and then maybe to focus on one particular feature and slowly increase the level of manipulation to see at what value it is no longer viewed as 'good'.

7 Conclusion

7.1 Summary of Progress

While this project started with a fairly different aim compared to the eventual one, the adapted aims have been successfully tackled and progress has been made overall into identifying what makes a piece of professional.

There were initial troubles with getting the audio alignment to work using DTW, which is what hindered the original aims of the project, however once a new path had been laid out there were minimal issues following.

The goal of manipulating specific features of music as been achieved, with reliable functions for this purpose being created. These functions provide a strong starting point for any future work with this aim.

The secondary goal of investigating which features are more relevant to the perception of music professionalism was also realised through the successful execution of an experiment. In this particular experiment, pitch was identified as the most significant feature, as the manipulation of this caused the biggest decrease in professionalism rating.

While the data cannot be considered statistically significant, as not enough data sets and participants were used, it gives an initial insight into humans perception of music and could be used for further research to produce scientifically sound results.

7.2 Future Work

As work progressed over the course of the project, many new potential areas of research identified that unfortunately the limited time span did not allow for.

Audio Alignment

While the initial work from previous versions of this project existed, the reliability of the function working was something that was working on at the beginning of this project and initially abandoned.

Future work could be applied to try and improve the DTW function even more so that it worked effectively every time. A starting point for this may be the quality of the beat detection within the algorithm, as during this project research it was suspected that the beats were not correctly being identified, which was what was contributing to the less effective alignment.

Dynamics Manipulation

The amplification function for the manipulation of dynamics was an effective way of altering the volume, however it was noted within the experiment feedback that the jump in volume seemed very sudden and artificial.

It may be worthwhile applying a fade in/out to the rise in dynamics to give the more natural feeling of a crescendo and decrescendo. This may reduce the feeling of artificial manipulation and give more reliable experiment results.

Experiment Question

The question used within the experiment was 'how professional do you think this piece of music is?', which was an effective way to judge the listeners perception of the professionalism of the piece.

However, it could be useful to include a secondary question of 'How much did you enjoy this piece?', also giving a rating out of 10.

This is because you could say that a piece of music could be deemed professional if the majority of listeners find it enjoyable. You could have a piece of music that is technically and performed professionally, but it may not be enjoyable to listen to.

it would be interesting to see if there is a difference in the average rating of a piece if the listener is specifically asked if they enjoyed the piece rather than if they think it is professional.

8 Reflection

This project has been an experience that has allowed a large range of learning opportunities. In hindsight choosing a project where I had little of the knowledge needed to tackle it was not the wisest move, with having never used MATLAB before and no experience in Digital Signal Processing.

However, this allowed for a very large learning curve from the onset of the project, becoming proficient in MATLAB and learning how digital signals are manipulated within a MATLAB environment.

Although the project started off with very ambitious aims and there were difficulties progressing at the beginning, I managed to adapt the aims and direct the work into a more productive path, and eventually still producing interesting and worthwhile results.

The vast amount of practical knowledge I have acquired is something that can only be gained from this kind of opportunity of spending many hours on a project. This experience produces the kind of invaluable skills that will be necessary when going into similar work places.

List of Abbreviations

STFT	Short Time Fourier Transform
ISTFT	Inverse Short Time Fourier Transform
DTW	Dynamic Time Warping
PV	Phase Vocoder
DP	Dynamic Programming
DRC	Dynamic Range Compression
MF	Modulation Frequency
MW	Modulation Width

References

Lartillot, O. and Toiviainen, P. (2007) *’A MATLAB TOOLBOX FOR MUSICAL FEATURE EXTRACTION FROM AUDIO.’*
URL: http://dafx.labri.fr/main/papers/p237.pdf

Flanagan, J. L. and Golden, R. M. (1966) *’Phase Vocoder’*
URL: http://www.ee.columbia.edu/˜dpwe/e6820/papers/FlanG66.pdf

Sakoe, H. and Chiba, S. (1978), *Dynamic programming algorithm optimisation for spoken word recognition* IEEE Transactions on Acoustics, Speech and Signal Processing, 26(1) pp. 43- 49

Silla Jr, C.N., Koerich, A.L. and Kaestner, C.A.A. (2008) *A Machine Learning Approach to Automatic Music Genre Classification.*
URL: https://kar.kent.ac.uk/24022/1/MachineSillaJnr.pdf

Tzanetakis, G. (2002) *MANIPULATION, ANALYSIS AND RETRIEVAL SYSTEMS FOR AUDIO SIGNALS.*
URL: http://www.cs.cmu.edu/˜gtzan/work/pubs/thesis02gtzan.pdf

Bibliography

Bochner, Salomon, and Komaravolu Chandrasekharan. *”Fourier Transforms. (AM-19).”* Google Books. Princeton University Press, 2 Mar. 2016.

Criel, Jo, and Elena Tsiporkova. *”GenTXWarper - Mining Gene Expression Time Series.”* GenTXWarper - Mining Gene Expression Time Series.
URL: http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWalgorithm.htm

Ellis, D. P. W. *”A Phase Vocoder in Matlab.”* A Phase Vocoder in Matlab., 2002. Web.
URL: http://www.ee.columbia.edu/˜dpwe/resources/matlab/pvoc/

J. Allen *”Short term spectral analysis, synthesis, and modification by discrete Fourier transform,”* in IEEE Transactions on Acoustics, Speech, and

Signal Processing, vol. 25, no. 3, pp. 235-238, Jun 1977.

Marshall, A. D. *"DAFX: Digital Audio Effects."* (2011) Web.
URL: http://www.cs.cf.ac.uk/Dave/CM0268/PDF/10_CM0268_Audio_FX.pdf

O'Connor, Karyn. *"VIBRATO: WHAT IT IS AND HOW TO DEVELOP IT."* Singwise. Web.
URL: http://www.singwise.com/cgi-bin/main.pl?section=articles&doc=Vibrato

Sethares, Bill. *"A Matlab Implementation of the Phase Vocoder."* Matlab Implementation of a Phase Vocoder. May 2005.
URL: http://sethares.engr.wisc.edu/vocoders/matlabphasevocoder.html

Wang, Quan. *"Dynamic Time Warping." Dynamic Time Warping (DTW).* Aug. 2013.
URL: http://www.mathworks.com/matlabcentral/fileexchange/43156-dynamic-time-warping–dtw-

"Amplifying Audio Signals." MathWorks Newsgroup. Sept. 2006. Web.
URL: https://uk.mathworks.com/matlabcentral/newsreader/view_thread/132962

"BandStop Filter." DADiSP Online Help.
URL: http://www.dadisp.com/webhelp/dsphelp.htm#mergedprojects/refman2/FncrefAE/BANDSTOP.htm

MathWorks. *"Dynamic Range Compressor."* Multiband Dynamic Range Compression.
URL: http://uk.mathworks.com/help/audio/examples/multiband-dynamic-range-compression.html