Undergraduate Project Report

2012/13

How heavy is heavy?

Research in the influence of amplifier parameters on the perceptual "heaviness" in guitar timbre

Name:	ZHANG Mincong
Programme:	Telecommunications with Management
Class:	2009215107
QM Student No.	090465589
BUPT Student No.	09212915
Supervisor:	HARTE Christopher

Date 2013.5.15

Table of Contents

Abstract

This project aims to investigate the amplifier parameters and try to derive an objective measure for the perceptual heaviness in guitar timbre. The “heaviness” in a guitar tone depends on a number of features but mainly affected by the equalization curve and the level of harmonic distortion applied to the sound. Four parameters (drive, bass, mid and treble) that have influence on the perceptual heaviness are selected to be studied. Two well known, heavy guitar riffs are recorded to be the source audios for experiments.

A Just Noticeable Difference (JND) experiment was carried out. For each parameter, Listeners needed to adjust the value of one target parameter and match the adjustable sound to the target sound. Two sounds were the same guitar sample and had the same settings except the target parameter. 20 volunteers did this experiment and 400 data were collected. After analyzing the results, the step-size, representing the full range of each parameter, were determined.

With the scriptable software REAPER, the sound database with all the combinations of parameters was generated. Then a web-based listening test was conducted. Listeners were asked to compare two audios of the same riff with different parameters, and rate which they thought was heavier. Finally 893 listeners registered and did the test. According to the weight factor, audios were sorted from “heavy” to “clean” in the database. Results were analyzed, and the correlations between parameters and heaviness ratings were shown. The results can be used as as a feature in music information retrieval systems, as well as music and genre detection.

Project ID QCAH10 How heavy is heavy?

摘要

本项目旨在研究吉他音箱的参数，以及尝试客观测量出不同参数下吉他音色“重”的程度。吉他音色的“重”的程度取决于很多因素，但主要受到音箱中的均衡曲线以及谐波失真的影响。本项目选择了四个影响吉他“重”音色的音箱参数（drive, bass, mid 和 treble）进行研究，并且录制了两个有名的重金属乐段作为源音频进行实验。

首先进行一个恰可察觉差（JND）实验。对于每一个参数，被测试者可以调节目标参数的数值，并且将这个可调节的音频与目标音频进行对比。两个音频为同一个吉他乐段，除了目标参数不同之外，其他参数配置完全相同。二十个志愿者参与了这个实验，并且收集到了四百个数据。分析的结果确定了四个音箱参数的变化梯度，该梯度能代表参数的全范围变化。

通过可编写脚本的音乐软件 REAPER，生成了拥有所有参数组合的音频数据库。然后项目发布了一个听力测试网页。参与者需要对比不同参数下两个相同的乐段，然后选择出他们认为较“重”的乐段。在后台数据库中，根据权重所有音频得到了从“重”到“轻”的排序。分析结果显示出了参数组合和“重”音色的相关性。结果数据可作为一个特征用于音乐信息检索系统，也可用于音乐流派的检测。

Chapter 1: Introduction

This project is about the sense of "heaviness" in guitar timbre. Generally the "heaviness" of guitar sound depends on many features including the equalization curve and the level of harmonic distortion applied to the sound. Guitar amplifiers have the equalization (EQ) settings, which control the intensity of bass, mid and treble frequencies when the sound is played. Also in guitar amplifier, the drive parameter, controlling the intensity of harmonic distortion, influences the heaviness as well.

In this project the aim is to investigate these parameters (drive, bass, mid and treble) and try to derive an objective measure for heaviness that could be used as a feature in music information retrieval systems. Two well known, heavy guitar riffs are selected to be the source audios for generating the sound database. Before the main listening test, a pre-experiment is conducted to determine the step-size for each parameter so that to fully represent the whole range. Then a perceptual listening test is carried out to find the correlations between parameters and heaviness ratings. Finally the results are analyzed to show the relationship between parameters and perceived heaviness.

Chapter 3: Section 3.1 introduces the production of the sound database. Helped by my supervisor, I manage to get the recording of several typical guitar riffs played by a guitarist. These riffs were directly recorded from the guitar without applying any distortion. Then I import Waves GTR3 Virtual Studio Technology (VST) plugin [1] to a music software REAPER [2], which supports Python scripts, for generating a number of distorted guitar riffs. Also I choose the high gain "Monster" amplifier based on a Marshall 100W stack, which is able to simulate the typical heavy metal sound in 1960s and 1970s. Successfully scripting REAPER in Python, I am able to change different settings in the amplifier and create different distortions automatically.

Chapter 3: Section 3.2 shows how to determine the proper increments for each parameter according to the results of Just Noticeable Difference (JND) [3] experiment. In the experiment JND means the least change that people can tell the difference in terms of hearing. This experiment aims to find out the general least range for each parameter that people cannot tell the difference. On the contrary, at the edge of this range, the JND can be acquired to represent the whole range of each parameter.

With the scriptable REAPER this experiment can be conducted efficiently. In this experiment, 20 volunteers (10 males and 10 females) are invited to do the test. For every test, a target audio with certain combination of parameters is played to the listener. Then a sound with the same settings but adjustable target parameter is played. The listener could adjust an unlabelled slider to change the value of target parameter until the adjustable audio sounds the same as the target audio. Audios are shuffled so listeners will hear a random order of different values in one target parameter. All audios can be replayed.

There are four parameters (drive, bass, mid and treble) to test and for each parameter there are five target values. Eventually 20 data from each listener and 400 data in total are acquired. After certain analysis, I decide to set the increments for drive to 0.1, 0.3, 0.5 and 0.8, and set the increments for other three settings to 0.1, 0.3, 0.5, 0.7 and 0.9.

Chapter 3: Section 3.3 focuses on the construction of the listening test website. The purpose of the web-based listening test is to compare two audios of the same riff with different parameters, and select the heavier one. Audios will be sorted from “heavy” to “clean” in the database. And the results can show the correlation between parameters and heaviness.

With the help of JND results, I script REAPER to generate all the audios as database of the website. Then I setup my listening test website. Finishing the frontend of the website, as well as a register system in my website, two algorithms are applied to make it efficiently collect the data and sort all the audios according to web users’ rating.

The “weight” is applied to all the audios. At first all the audios’ weights are set to zero. For one round of testing, 20 questions are provided. Every question requires user to choose the perceived “heavier” audios in two provided audios. The first algorithm takes the responsibility to select the audios for each question. And the second algorithm does the weight calculation after the audios are chosen by the user. The details of these two algorithms are illustrated in Section 3.3.3 and Section 3.3.4.

Finally the results are analyzed in Chapter 4:. According to the statistical analysis, both drive and bass parameter had positive correlation between heaviness. However, for mid and treble parameter, high level of heaviness is located on two extreme points, but low level of heaviness is located in the middle.

Chapter 2: Background

The notion of "heaviness" is applied to several instrument timbres, including drum timbres, bass timbres and guitar timbres. But normally the "heaviness" is used to describe guitar timbres [4]. Timbre perception studies have attempted to connect verbal descriptions with some features of the acoustic signal. However, both perception and verbal description can be multidimensional and have many influence factors. Because of multidimensional perception of timbre, which is caused by the interaction of several acoustic factors, the correlations are difficult to measure. However, in a single genre of music, a listener can distinguish greater or lesser degrees of a given timbre [5].

So this project focuses on one dimension of perception - "heaviness". And only one timbre - guitar timbre is chosen to be the acoustic element to study. Guitar riffs are recorded without the mixture of any drum, bass and vocal elements.

A methodology has been used to study timbre. This study linked a verbal description of tone quality with acoustic features [5]. Besides, this study suggest to artificially reduce or increase an identified acoustic elements to find if it is more or less heavy than original sound. Therefore in my project I try to conduct JND experiment and find the proper step-size of parameter increments.

2.1 The development of guitar distortions

Early 1960, a band "The Kinks" produced a pioneering song "You really got me". In this song the members of this band found that their guitar sounds were limited and did some innovation — cutting the speaker - and kept playing power chords to create a new amazing guitar sound.

In terms of heaviness and volume, Jim Marshall developed Marshall Amplifier to make full loudness. He created the first 100 watt Marshall Amplifiers. Then the band "The Who" used first Marshall Stack and provided powerful music show [6]. They started the music revolution, because the next band made their music loud and the next much louder.

In 1960s, Jimi Hendrix expanded the guitar sound beyond just volume and distortion, pushing the development of heaviness. At that time guitarists used Fuzz Tones to make music very electric and distorted, and also sounds can be developed in the studio. Turning up the volume and creating some distortions through the amplifiers and guitars made the beginning of heavy metal.

I choose these two riffs - “Enter Sandman” by Metallica and “Smoke on the Water” by Deep Purple as the experiment samples in this project. Deep Purple is cited as one of the pioneers of hard rock and heavy metal. “Smoke on the Water” was first released in1972 and was played through a distorted Marshall amplifier. And Metallica is one of the founding "big four" of thrash metal. “Enter Sandman” was released in 1991. These two well-known guitar samples can represent heavy metal music in different period.

2.2 Guitar amplifier and Virtual Studio Technology (VST)

A guitar amplifier is an electronic amplifier designed to amplify the electrical signal of an electric guitar[7]. The material of electric guitar strings is metal. With the planted pickups in the guitar's bridge, vibrations of the strings keep cutting the magnetic induction line, and then the sound signals are “picked up”. The signal is transmitted to the amplifier for further modulation. Guitar timbre can be modified by most guitar amplifiers. Mainly the amplifier can emphasize or de-emphasize certain frequencies and add electronic effects.

There are many controls in guitar amplifiers and each of them has unique effect to the guitar tone. But mainly the “heaviness” is affected by the equalization curve and the level of harmonic distortion applied to the sound. The equalization curve is controlled by equalization (EQ) settings, which normally have bass, mid and treble faders. These settings can be used to emphasize or de-emphasize relevant frequency level. The harmonic distortion is controlled by drive or gain fader. Different amplifiers have different control name towards the harmonic distortion, either drive or gain will be used. But amplifier will not use both of them. Besides, “Presence” is another parameter that has influence to the “heaviness”, but the mechanism of this parameter is complicated so it will not be discussed in this project.

Vacuum tubes (or called as valves in Britain) were mainly used in guitar amplifiers before 1970s. After that period transistors became more popular and they were widely used in guitar amplifiers. In my project I will simulate the guitar sound by Marshall 100-watt amplifier, which is a classic valve amplifier [8]. In this amplifier, the “drive” is used to control the harmonic distortion.

Virtual Studio Technology (VST) is a type of interface, with which digital sound can be modulated in certain effects and further music recording can be achieved. Digital signal processing technique is used in this software. So traditional recording can be simulated in this software. However a Digital Audio Workstation is required. VST plugins can be imported to one Digital Audio

Workstation to run. In this project REAPER [2] is used as the host Digital Audio Workstation. The Waves GTR3 VST plugin [1] is imported in REAPER, and the high gain "Monster", which simulates the well known Marshall amplifier, is chosen. Figure 1 on page 10 shows the interface of Waves GTR3 VST plugin.

Figure 1: Waves GTR3 VST plugin

However, since the VST plugin is used to simulate the amplifier, the nature of the equalization controls on the amplifier model can be analyzed. A 2-second white noise sound was played through the model with drive setting = 0 (so that the guitar would sound clean). And the outputs for different EQ settings were recorded. For one EQ control, the increment was set as 0.1 from 0 to 1, while the other two EQ controls set to 0.5. The REAPER was scripted to generate all the output audios. After generating all the files, thirty 2-second wav files were processed to show the frequency response of the amplifier. The response is plotted in Figure 2 on page 11.

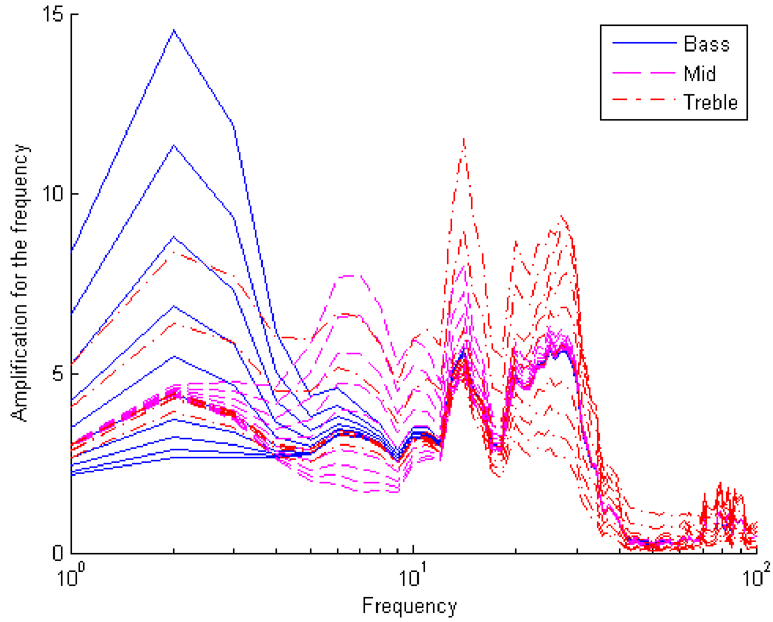

Figure 2: Frequency response of the amplifier

Applying different amplifier parameters, I tried to link a real amplifier to an oscilloscope to see the wave forms. Figure 3 on page 12 shows the pictures. The first original signal is a sine waveform. When it was applied certain level of bass parameter, the waveform was expanded. But when it was applied certain level of treble parameter, the waveform was narrowed. When the drive parameter increased, the top of both two signals were cut to some extent. This was the mechanism of harmonic distortion.

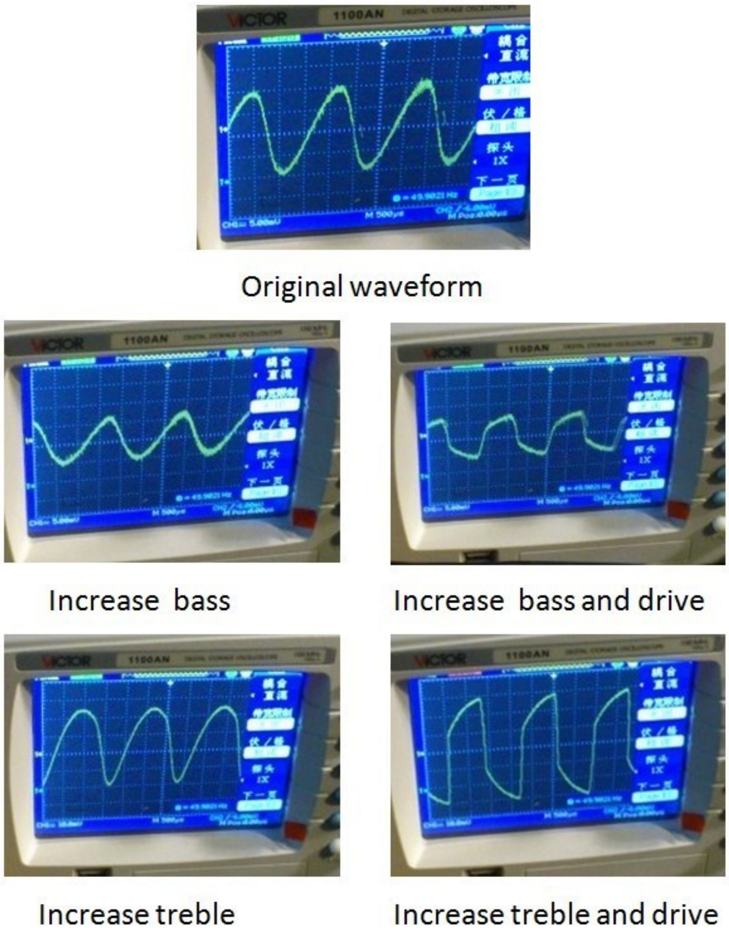

Figure 3: Waveforms

Chapter 3: Design and Implementation

3.1 Production of sound database

In order to conduct the experiments, not only a digital audio workstation was required to simulate different guitar amplifiers and generate the distorted guitar sounds, but also it should be scriptable. That's because a sound database with a large number of audios were required to generate, and the first experiment would need to change settings in real time.

I used the REAPER [2], which is a platform to implement Virtual Studio Technology (VST) plugins and generate samples. After configuring the environment for both Python3.1 and REAPER, I managed to script this software so its internal functions and parts of the APIs could be accessed. In addition, Waves GTR3 VST plugin [1] was used. In the plugin the high gain 'Monster' amplifier was able to simulate classical heavy metal distortion based on a Marshall 100W stack [8]. Figure 4 on page 14 shows the interface of this software.

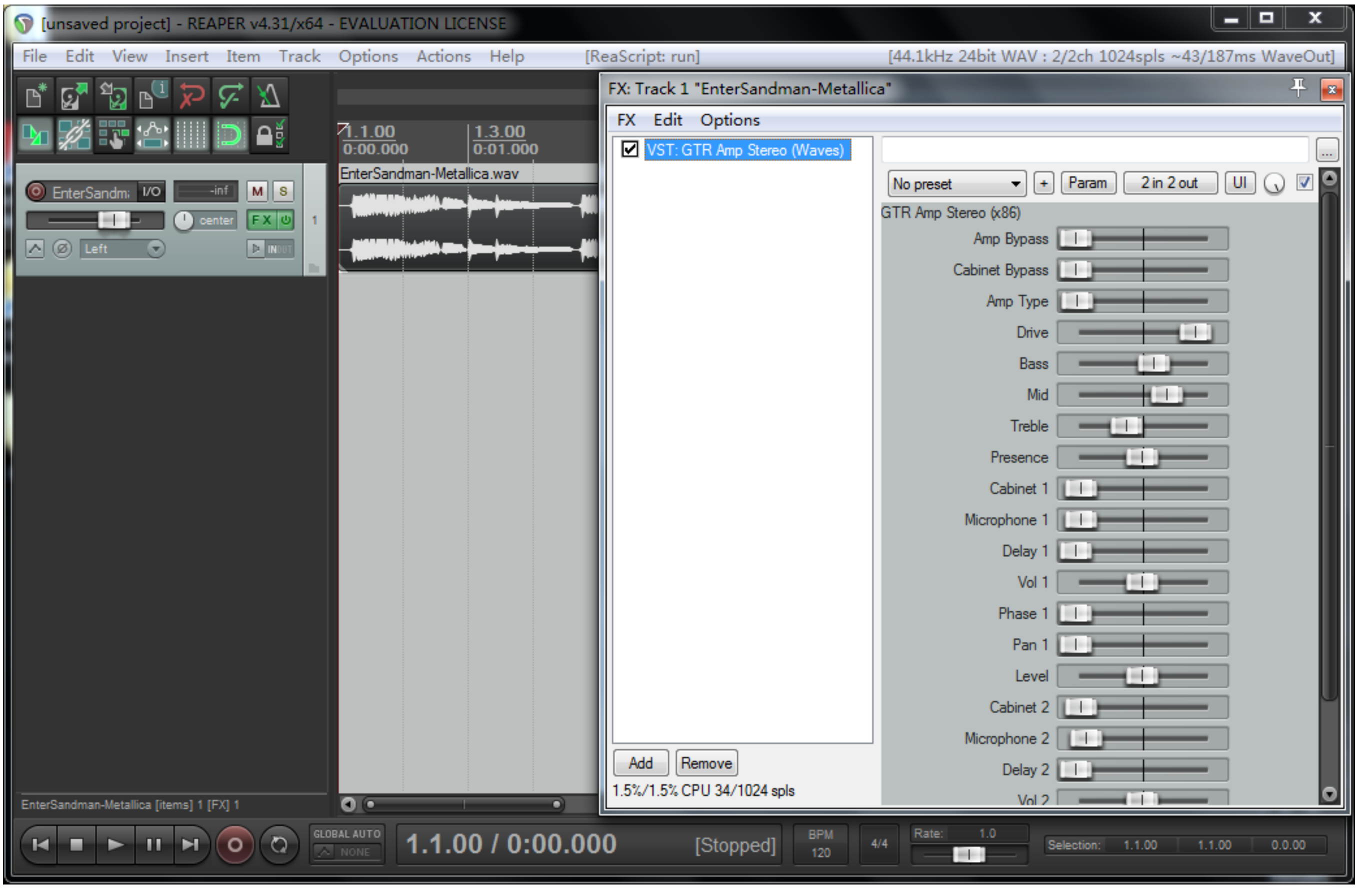

Figure 4: REAPER Interface

Generally in amplifier settings, "heaviness" was affected by these four amplifier parameters: drive, bass, mid and treble. Two well known, heavy guitar riffs ("Enter Sandman" by Metallica and "Smoke on the Water" by Deep Purple) were chosen to be the experimental samples. After certain programming, the sound database was able to generate. But the increments for each parameter needed to be determined. Thus it was necessary to carry out the next experiment.

3.2 Just Noticeable Difference experiment

Just Noticeable Difference (JND) is the smallest detectable difference between a starting and secondary level of a particular sensory stimulus [3]. In this research JND means the least change in certain amplifier settings that people can tell the difference in terms of hearing.

This project focused on researching people's JND in the riffs that applied different amplifier settings (drive, bass, mid and treble). And this experiment aimed to find out the general least range for each parameter that people cannot tell the difference. On the contrary, at the edge of these ranges, the JND can be acquired to represent the whole range of each parameter.

With the help of the JND results, the proper increments for drive, bass, mid and treble parameters could be determined separately, so that to limit the number of samples in database to a reasonable level but also fully represented all the settings. This experiment would ensure the accuracy of sample data in the web-based listening test.

3.2.1 Conduct the experiment

This JND experiment was carried out using REAPER. All these following allocations in the experiment intended to eliminate the potential environmental factors that might influence the accuracy of JND experiment. 20 volunteers (10 males and 10 females) were invited to do this experiment. The guitar samples and settings were provided in the following order:

5 males (‘smoke on the water’) drive, bass; (‘enter sandman’) mid, treble;

5 males (‘smoke on the water’) treble, mid; (‘enter sandman’) bass, drive;

5 females (‘smoke on the water’) drive, bass; (‘enter sandman’) mid, treble;

5 females (‘smoke on the water’) treble, mid; (‘enter sandman’) bass, drive.

Then the REAPER was programmed and the experiment was conducted in this way:

1) For each target parameter and each listener, set the other three parameters in Gaussian random values (between 0 and 1) instantly.

   -Using Gaussian random function (random. gauss (mu, sigma)) was to make it less likely to generate some odd settings in pure random function, like bass=0, mid=0, treble=0.9. Obviously these settings would not be used by guitarists in our real life. Instead, using Gaussian random settings could acquire more random values in the centre so that to get more realistic settings.

2) Generate audios with five reference increments (0.1, 0.3, 0.5, 0.7 0.9) but with shuffled names in a list like A3.wav, A0.wav, A2.wav, A1.wav, A4.wav. And in the folder the audios were put in order.

   -This was to provide reference values evenly yet avoid the opportunity for listeners to form a pattern of increments in their mind - everything should rely on their sense of hearing.

-The details of each audio were print in a data.txt file, including the name, value of each parameter and the initial value of other three settings.

-Different target audios were generated in separate folder in case of playing other audios when testing one value of one parameter, so that to reduce the influence between similar audios.

3) Let listener play the target audios and find out the most similar value by adjusting the target parameter in a specific user interface with unlabelled fader and code name of the target parameter.

- All the audios could be replayed several times.

- Listeners could not try to guess the target values due to the unlabelled fader. Also they didn't know the actual parameter that they were tested.

4) Once the listener finished adjusting the unlabelled fader and pointed out what they thought the same sounds, he/she would press a certain button on the keyboard to record the current value in the data.txt file. But the window would only show 'Current value is recorded; please play the next audio for testing.'

-Figure 5 on page 16 shows the example windows. And Figure 6 on page 17 shows the scene that listeners are doing the experiment.

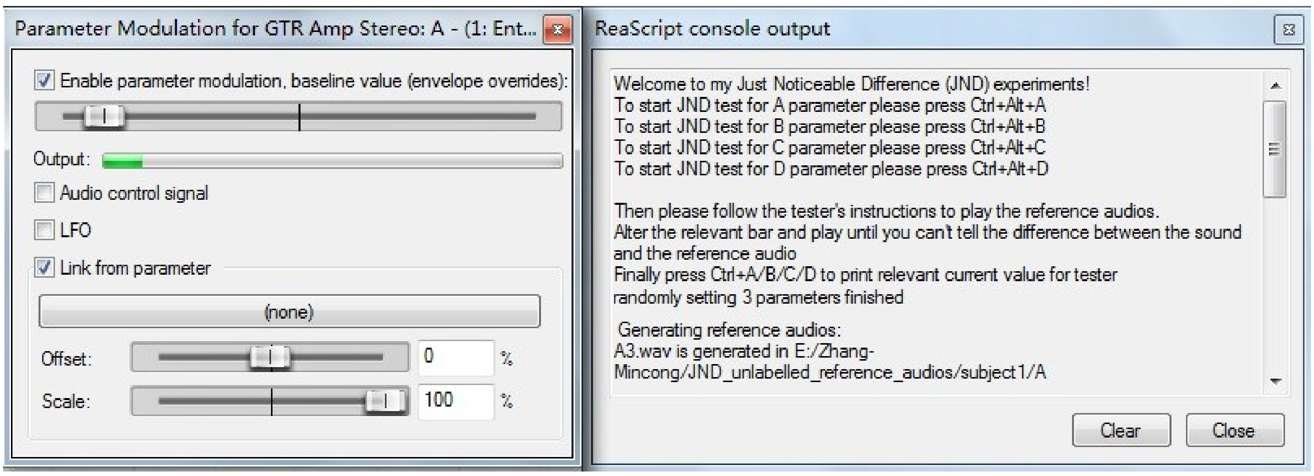

Figure 5: Example Experiment Windows

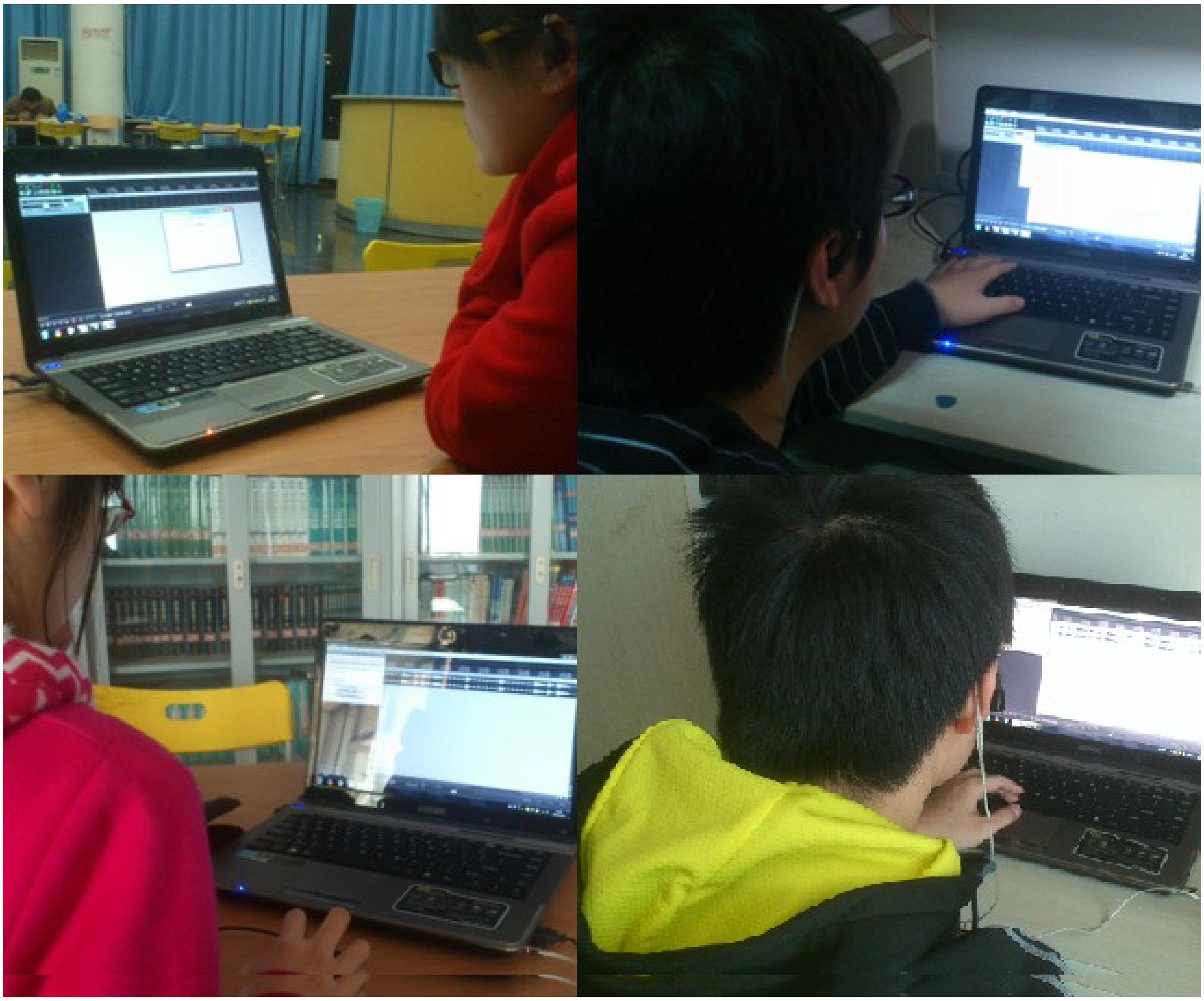

Figure 6: JND Experiment

3.2.2 JND Results

For each listener there were 4 target parameters' experiments: drive, bass, mid, treble. For each target parameter, there were 5 target values. So finally I had 400 data illustrating the JND after finishing this experiment among 20 volunteers.

These data were drawn in Box-whisker Plots in Figure 7 on page 18. The median value was drawn as the central mark on each box. Two box edges show the first and third quartiles, respectively (the 25th and 75th percentiles of approach values listed form the lowest to highest datum). The ends of the whiskers are the adjacent values, showing the extreme data values that were not considered outliers .The mark '+' shows the outliers.

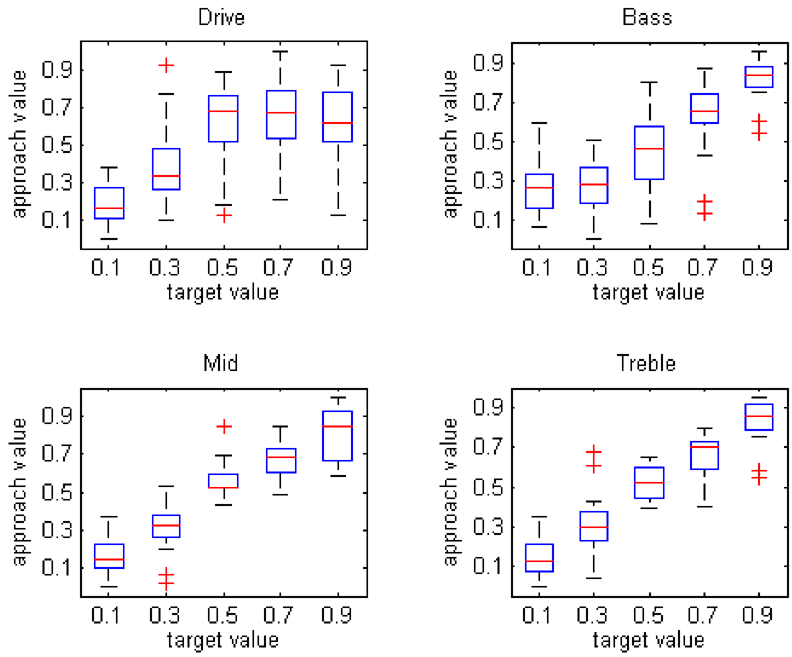

Figure 7: JND Results

I also calculated the root mean square (RMS) and standard deviation (STD) in Table 1 for each target value in each parameter.

Table 1: JND error data

Settings	Error	Target Value				
		0.1	0.3	0.5	0.7	0.9
Drive	RMS	0.1228	0.2245	0.2372	0.1963	0.3539
	STD	0.1002	0.2124	0.2153	0.1965	0.2166
Bass	RMS	0.2063	0.1389	0.1967	0.2041	0.1283
	STD	0.1304	0.1408	0.1876	0.1884	0.1041
Mid	RMS	0.1177	0.1227	0.1078	0.0940	0.1624
	STD	0.1036	0.1251	0.0905	0.0910	0.1423
Treble	RMS	0.1144	0.1525	0.0842	0.1102	0.1260
	STD	0.1056	0.1556	0.0829	0.1066	0.1098

Firstly, the drive boxplot in Figure 7 on page 18 shows that the listeners could estimate the values when target drive values are 0.1 and 0.3 to some extent. However, when drive target value increased to a higher level (0.5, 0.7 and 0.9), people were not sensitive to the target values and the approach values could not maintain around the target values. Also the RMSs (increasing to above 0.35) and

RMSs (all around 2.0 when target value increase from 0.3 to 0.9) show the less accuracy of high target values. Therefore it was safe to set JND for drive as 0.2 when target value $\leq$ 0.5, but for constraining the sample space I decided to set JND as 0.3 when target value > 0.5.

Secondly, the bass boxplot in Figure 7 on page 18 illustrates that the listeners were not sensitive when bass target values were in a low level. People mistakenly approached a higher value (median value reached about 0.3) when target value was 0.1. Additionally, the figure indicate that listeners were not very sure when target value is 0.5. The relevant higher RMS and STD at 0.5 proved that as well. But with the increasing of bass value people had a better estimation. Therefore I decided to set 0.2 as the JND of bass.

It is noticeable that the approach values of mid and treble parameter had a linear pattern. And the differences between the target values and median values of these two parameters were no greater than 0.05. Table 1 also showed that the RMSs of mid and treble were approximately 0.1, and the STDs were around 0.1 as well. Despite of some acceptable fluctuation in the results, I could figure out the appropriate JND of mid and treble parameter was about 0.1.

3.2.3 Determine step-sizes and generate sound database

Finishing the JND analysis, the increments for each parameter could be determined. However, the settings' step-sizes were set greater than their relevant JND level but similar scale. That's because set increments equal to JND would provide lots of samples that merely had slight difference but poor user experience in terms of hearing. Thus the step-sizes for the four settings were settled as:

1) Distortion (drive): 0.1, 0.3, 0.5 and 0.8;

2) EQ settings (bass, mid and treble): 0.1, 0.3, 0.5, 0.7 and 0.9.

Fully using the information from JND results, the increments were able to represent a full range of these settings. Again, the REAPER was scripted to generate the sound database. The determined increments led to $5\times5\times5\times4=500$ different combination of parameters for one guitar sample. Thus for two guitar samples ("Smoke on the Water" and "Enter Sandman"), 1000 audios were generated and the sound database was produced.

3.3 Web-based listening test

After JND experiment the sound database was acquired and could be used in the web-based listening test. I setup my listening test website by using Django, which is a Python Web framework

[9]. The purpose of this web-based listening test is to find out the correlation between parameters and the heaviness rating. Two versions of the same guitar sample would be provided to listeners. And they would listen and rate which one they thought was heavier. Audios with different combination of parameters were sorted from "heavy" to "clean" by applying weight factor in the database. The collected results were used to determine the relationship between amplifier settings and heaviness. Furthermore, the settings that affect the perceived 'heaviness' could be well-studied.

3.3.1 The frontend of the website

As for the website pages, a main html file was written as a base. Every other html file inherited the base html file but with unique functions. After setting the URL for each page, the website can be visited through browsers. Also JavaScript was used to handle the basic events of the website.

A statement of the purpose of the listening test was posted on the home page. Apart from that, a register system was set up. This system required only users' email address as their username but no password. Instead, the system generated the password (which was consisting of one famous band name and two random numbers) for each user when they first registered in my website. Then the system will send username and password to listeners' email automatically by using SMTP function in Python. This register system provided not only reliable test environment (including saving listener's work if they unfinished and want to continue later), but also had no access to users' regular password, which might be leaked and caused problems. Figure 8 on page 21 and Figure 9 on page 21 shows the screenshots of the homepage and the register page.

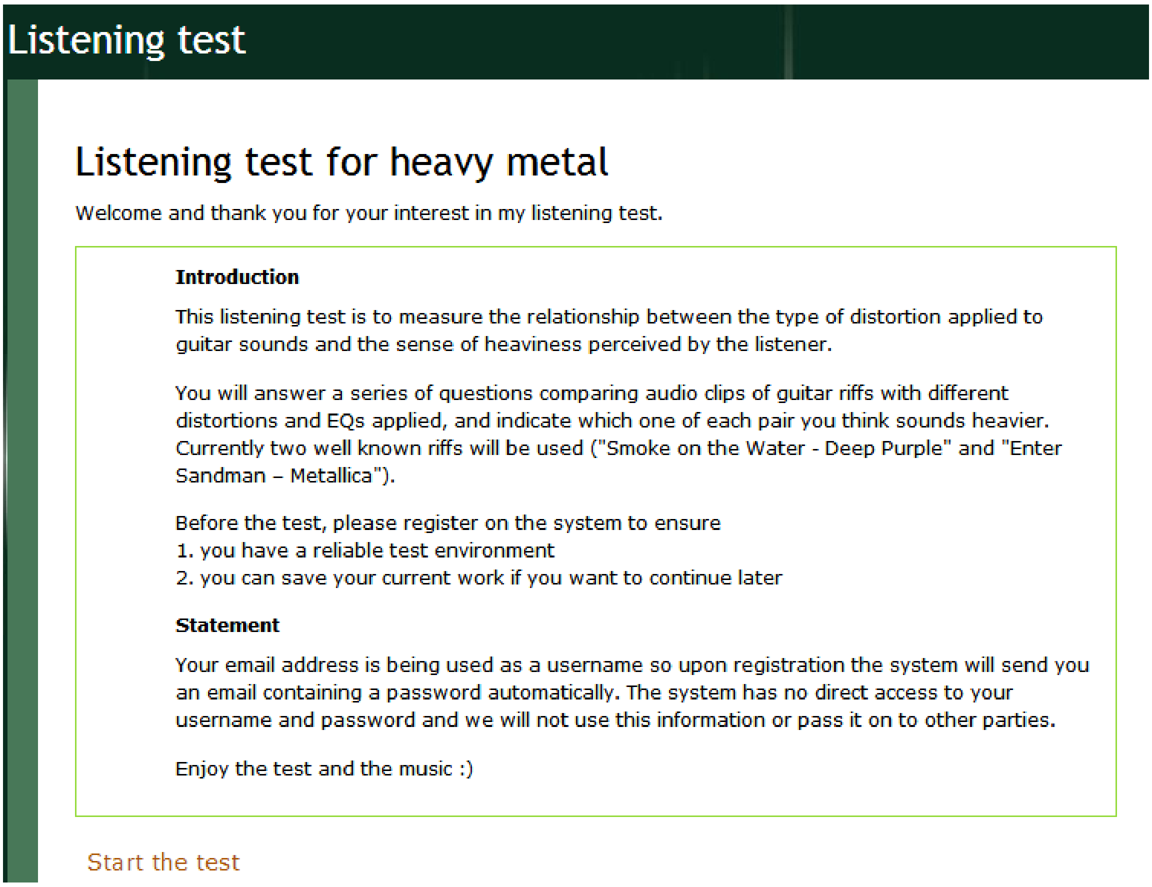

Figure 8: Homepage

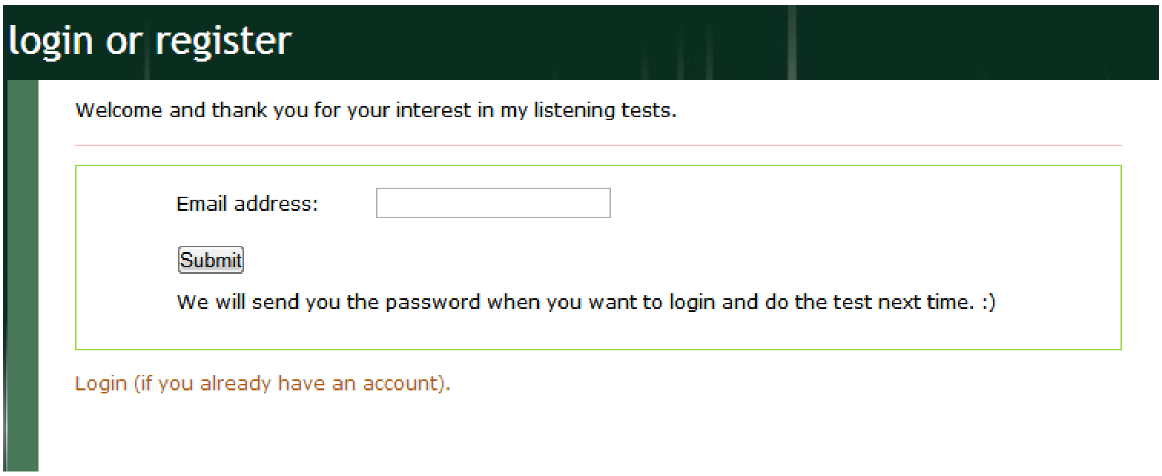

Figure 9: Register Page

Additionally, user experience was good enough because the first time to do the test users had no need to think of password. If they wanted to do more test later they would check their password via email and login again. When a new user enter the test, an example would appear at the beginning of the test, showing two audios with clean guitar sound and distorted (heavy) guitar sound. An illustration of guitar riffs' heaviness was posted as well. After collecting listener's information about heavy metal this page wouldn't appear any more. Figure 10 on page 22 and Figure 11 on page 22 shows the screenshots of the information collection page and the question page.

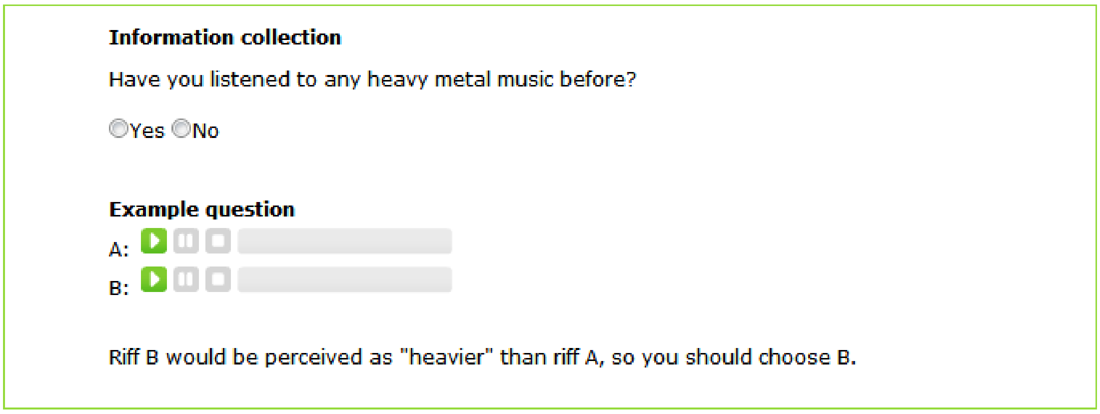

Figure 10: Information Collection Page

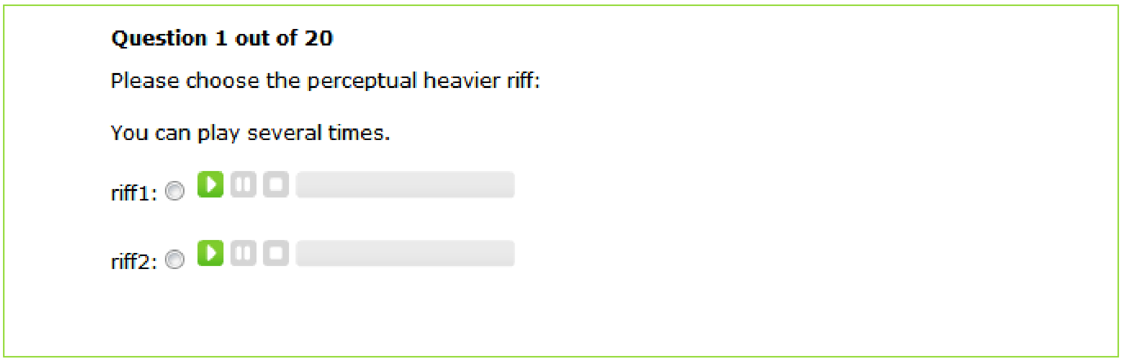

Figure 11: Question Page

3.3.2 The backend of the website

SQLite was chosen as the database and "Mp3s" class was created to store audios information. "Web users" class was to store users' information. Moreover, "Pairs" class was created to store the pairs of audios that had been selected for testing. Figure 12 on page 23 shows the backend interface (admin page). These classes were defined in the models.py file.

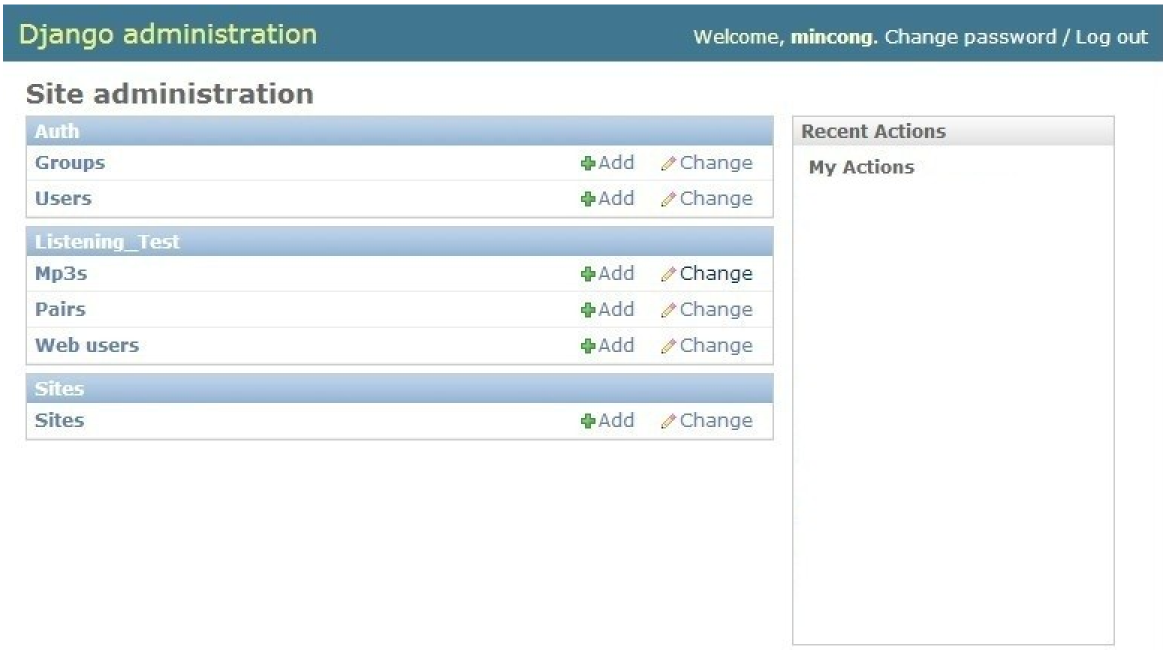

Figure 12: Backend Interface

There are two lists of guitar riffs with different distortions in the database, and two riffs are different. The two riffs are "Smoke on the Water - Deep Purple" and "Enter Sandman - Metallica" respectively. Question 1-10 is about the riff "Smoke on the Water - Deep Purple" while question 11-20 is about the riff "Enter Sandman – Metallica". In the database source1 and source2 were used to distinguish them.

For each logged-in user, the system would assign a unique session, and once the user logged out or closed the browser, his/her session would expire. Also when a user was doing the test, the cookie would be used to store the selected audios' information and update the database. Therefore this strategy supported multiple people doing the test at the same time. After many times of test the guitar riffs would be sorted from "clean" to the most "heavy" in the database. Figure 13 on page 24 shows the request was being processed in the command line.

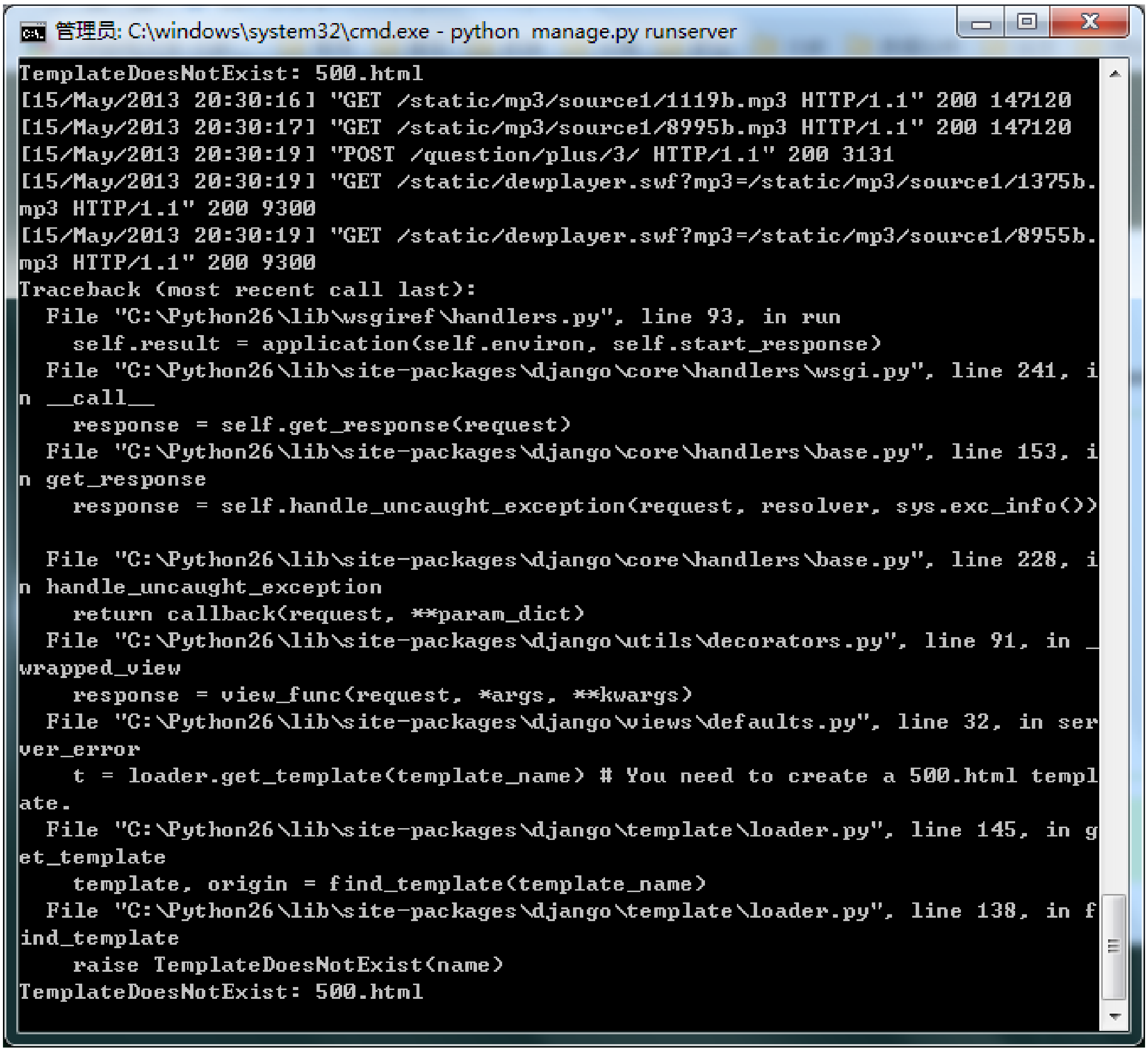

Figure 13: Processing Request

Mainly two algorithms were designed to optimize the sorting; they were written in the views.py file. Based on the JND results, there were a large number of audios to be sorted. In totally 500 combinations of four parameters, even comparing any two of these required C_{500}^{2} =124750 comparisons. Therefore an optimized strategy was considered to reduce the total number of comparisons. This strategy contained two algorithms. The basic idea is explained: Suppose sort three audios from "heavy" to "light". Let A, B and C denotes these three audios. Suppose after two comparisons, A is "heavier" than B and B is "heavier" than C, there is no need to compare A and C.

3.3.3 Algorithms for selecting two guitar samples

The flowchart in Figure 14 on page 26 illustrates the details of how the algorithms work. The "weight" was introduced to all the audios. Initially all the audios' weights are set to zero. Following the flow chart in Figure 14 on page 26, for each question in the listening test the system would scan all the weight level (at first there is only one weight level: 0), and selected one weight level. Otherwise the system would randomly select two samples from all the weight levels.

Then the "distance" calculation was introduced. Let $audio_1$ denote the first chosen audio; $[d_1 b_1 m_1 t_1]$ denote the value of four settings for $audio_1$. Let $audio_i$ denote the next chosen audio; $[d_i b_i m_i t_i]$ denote the value of four settings for $audio_i$. Let D denote the distance between $audio_1$ and $audio_i$. And the equation for the "distance" is below:

$$D = \sqrt{(d_1 - d_i)^2 + (b_1 - b_i)^2 + (m_1 - m_i)^2 + (t_1 - t_i)^2} \tag{1}$$

After calculating the distances between the first chosen audio (define as audioA in the next step) and all the other audios, a temporary buffer was used to store the results. If there were more than one element in the buffer, the system would sort the results and choose the audio with the maximum distance. Or if there were only one element in the buffer, the one audio would be chosen as the second audio (define as audioB). Again follow the illustration of the flowchart in Figure 14 on page 26, the "Pairs" is the class to store the pairs of audios that had been selected for testing (showed in Figure 12 on page 23). If the second audio was recorded in "Pairs", it indicated that this audio pair had been compared before, so the system would delete the information of the second chosen audio and provide opportunity for another pair of audios. And it would loop back to choose another audio in the buffer. The judgement of whether the chosen audio was recorded in "Pairs" was to accelerate the speed of finishing the total comparisons, and try to go through all the audios. Finally, the two audios would be selected and present to listener for comparison.

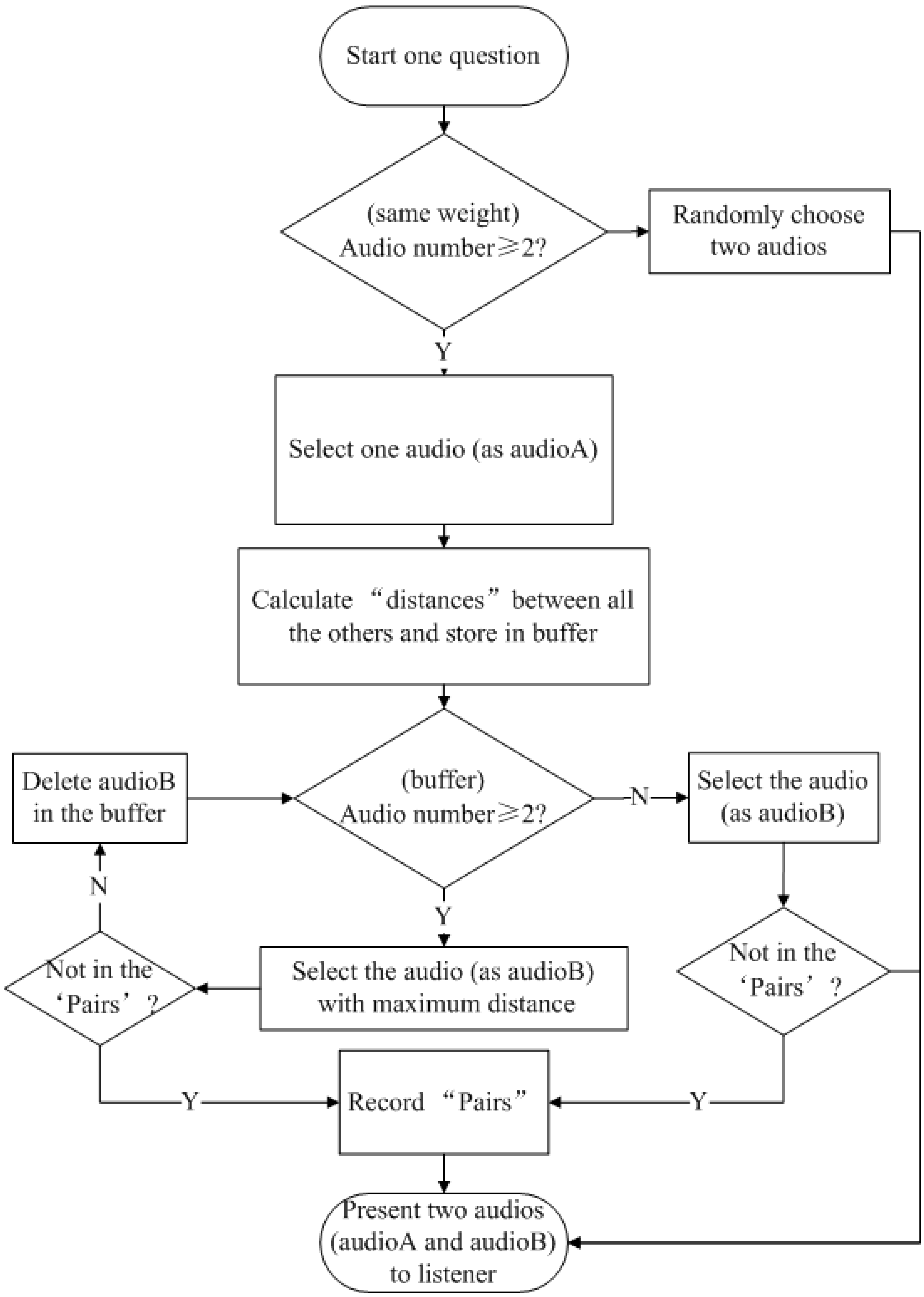

Figure 14: Flowchart for selecting two audios

3.3.4 Algorithms for calculation the weight

After all these calculations two audios were presented to the listener. After a perceived heavier one was chosen, the choice was sent to the system and my sorting algorithm began to work.

Let A denote the first audio and B denote the second audio, and their weights were weight A and weight B, respectively. Suppose the listener chose A as the perceived heavier one, the previous weights stored in the database of these two audios would be taken into consideration. And there were three situations for the next calculation, demonstrating in Figure 15 on page 28. If weight A > weight B, their weights would have not change, because in the database audio A had already "heavier" than audio B. If weight A was equivalent to weight B (weight A == weight B in programming language), in the database these two audios were in the same weight level. The system would assign weight A +1 to weight A, but for weight B there would not be any change. After this assignment, audio A was "heavier" than audio B in the database. But if weight A<weight B, it means in the previous calculations, audio B was "heavier" than audio A. According to this listener's choice, the system would adjust the weight factor for these two audios. It would assign weight B +1 to weight A, showing that audio A was "heavier" than audio B. For audio B there would not be any change. As soon as the listener finished one question and click "Next question", the weights of two audios would be updated in the database.

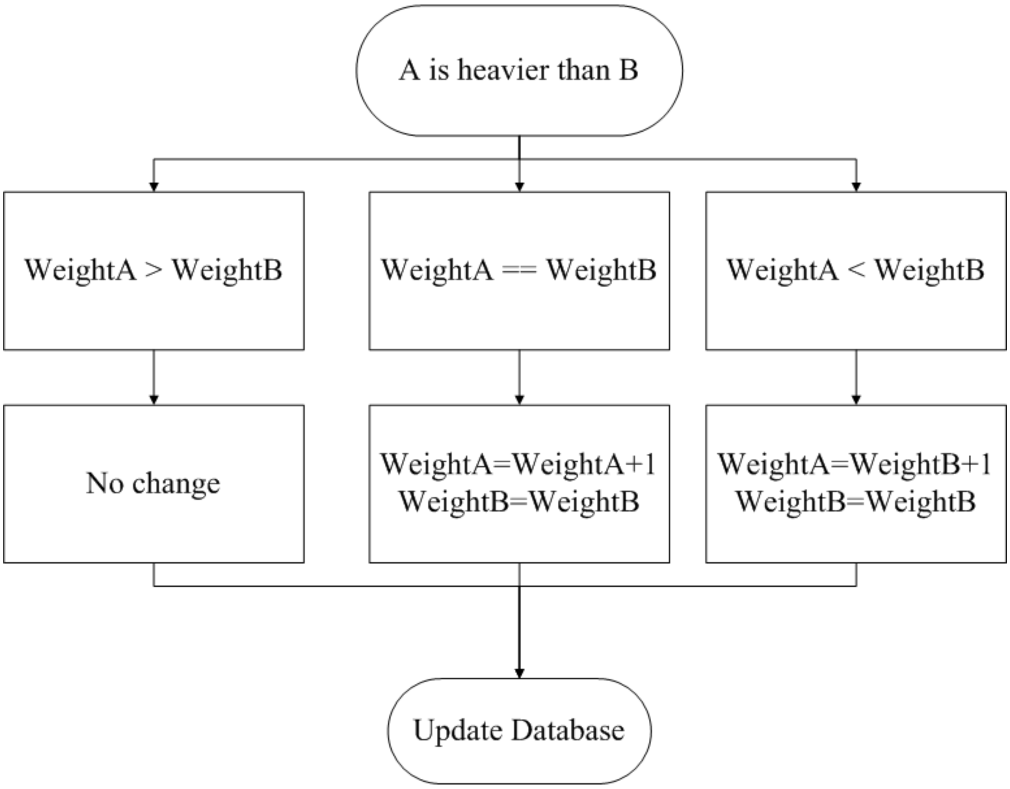

Figure 15: Flowchart for updating weight

3.3.5 Publish website

A Linode Virtual Private Server (VPS) [10] was rented to be the web server. After configuring the environment and uploading all files and audios to the server, the listening test website was published. Figure 16 on page 29 is the listening test website located at http://178.79.149.165/. The address of the listening test was posted on my SNS websites such as Weibo and Renren, as well as relative BBSs. Finally 893 web users registered in the website and the audios were sorted in the database. Figure 17 on page 29 shows a part of the sorted audios' information.

Project ID QCAH10 How heavy is heavy?

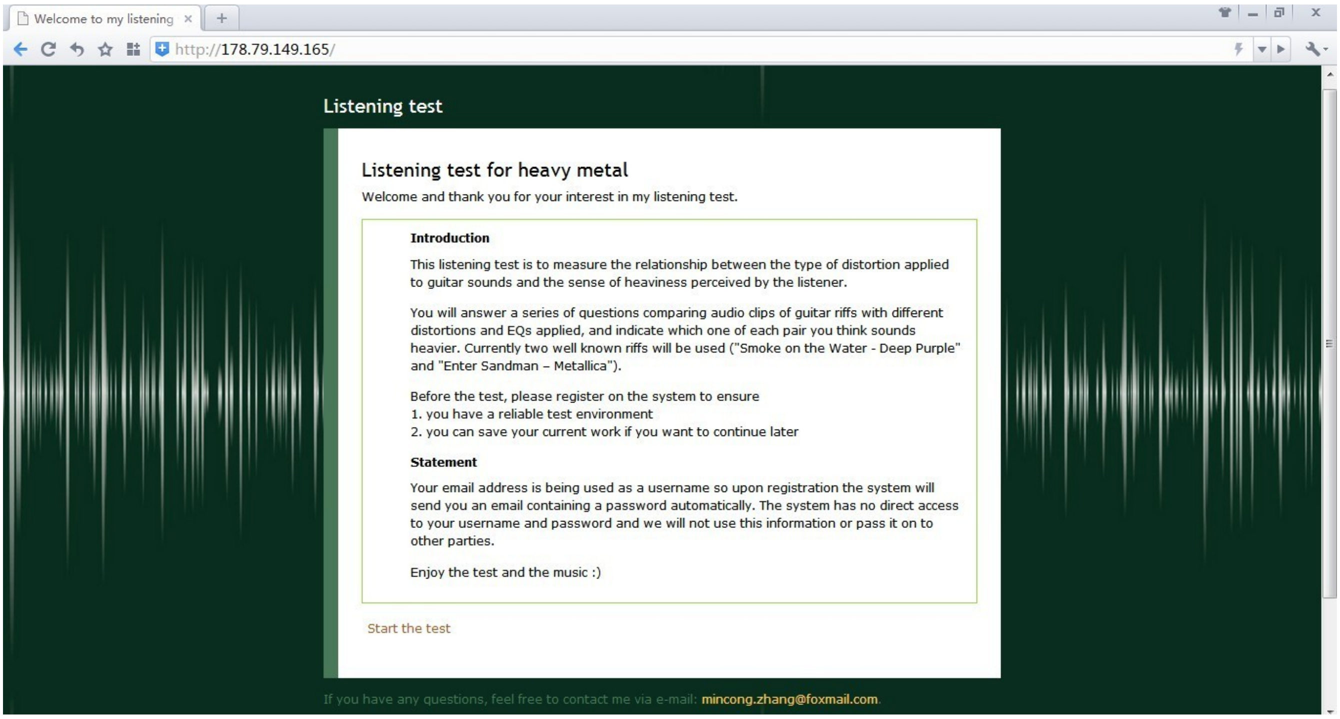

Figure 16: Listening Test website

Filename	Source 1	Weight 2
5919b.mp3	source1	62
8919b.mp3	source1	61
1919b.mp3	source1	60
3919b.mp3	source1	59
8993b.mp3	source1	58
3911b.mp3	source1	57
8911b.mp3	source1	57
1939b.mp3	source1	57
8999b.mp3	source1	56
8939b.mp3	source1	55
8979b.mp3	source1	54
8915b.mp3	source1	54
8719b.mp3	source1	54
3991b.mp3	source1	54
8957b.mp3	source1	54
8917b.mp3	source1	54

Figure 17: Information of the sorted audios

Chapter 4: Results and Discussion

4.1.1 Results: derivation of objective measure

After collecting enough data from the listening test, the visualized four dimensional space data was drew in Figure 18 on page 30, in which the weight was represented as the intensity of greyscale: the settings with the greatest weight was drew in black ball (greyscale=100%); whilst the settings with the lowest weight was drew in white ball (greyscale=0%). The data in four sub graphs showed a pattern: with the increasing of drive value the number of dark points was increasing. For each graph the relative heavier settings focused on the field that have high bass and treble values but low mid values, or the field that all the bass, mid and treble values were very high.

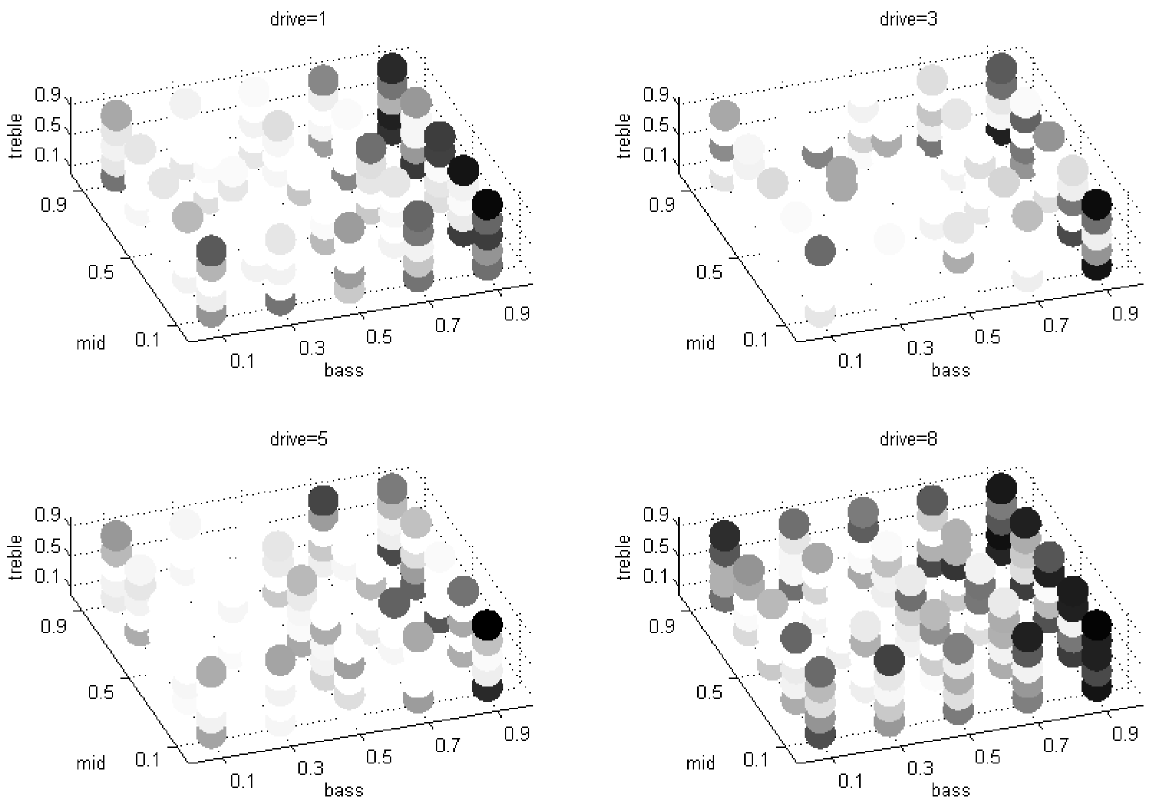

Figure 18: Visualization of Backend Data

Afterwards, a further statistical analysis was done. The correlation between heaviness rating and each parameter was analyzed in Figure 19 on page 31. For each parameter, the general weight distribution in different values is plotted. A black dot inside a white circle represents the median weight value and the median points are linked with whiskers. Filled boxes show the first and third quartiles, respectively. While the ends of the lines represent the adjacent weight values. Outliners are demonstrated as mark 'o'.

The first diagram in Figure 19 on page 31 shows that with the increasing of drive value, the general weight values increase steadily. Therefore a positive correlation between drive parameter and heaviness can be derived. For the bass parameter, apart from a relevant high weight locating on 0.1 bass values, the diagram shows a strong positive correlation between an increasing bass and increasing heaviness. The curves for mid and treble parameters, however, make U-turns, showing a high level of heaviness at two extreme points (0.1 and 0.9), but a low level of heaviness when their values are around 0.5.

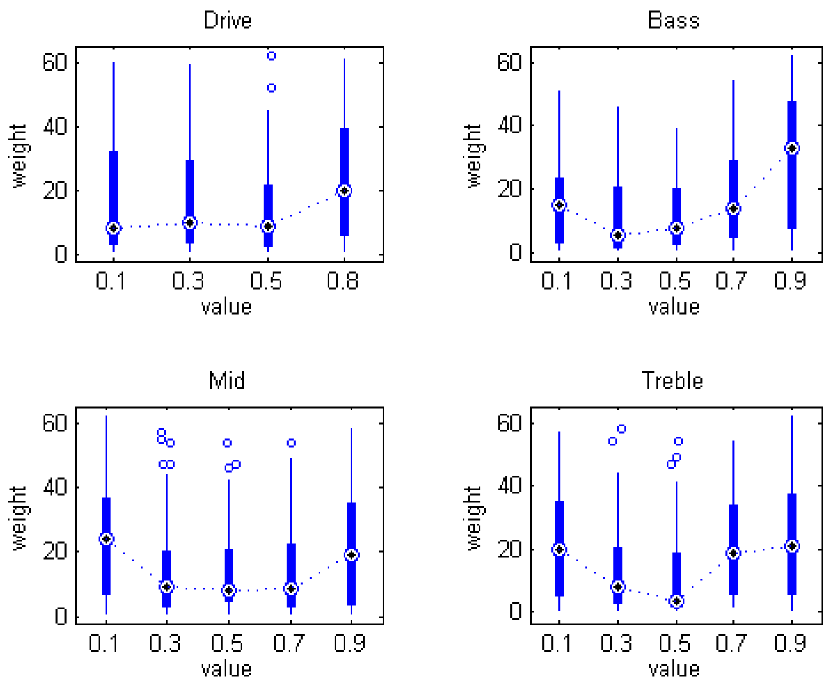

Figure 19: **Correlation between Heaviness and Each Parameter**

4.1.2 Discussion

With the analysis, the results could contribute to the field of music information retrieval systems. Also by applying further analysis, the data could be applied in the fields of emotion or mood detection in music and genre detection. Although two experiments support me finish the measurement of the relationship between parameter values and "heaviness", there is still room for improvement.

Firstly in the JND experiment, for each target parameter and each listener, I have set the other three parameters in Gaussian random values (between 0 and 1) settings instantly. Even the results were good enough to support the next experiment; this strategy didn't maintain one factor different at a time, so the effect of a single factor (target parameter) couldn't be fully determined. The solution is for each target parameter and each listener, set the other three parameters in the same Gaussian random values, or just set all of them in 50% values.

Secondly feedbacks from the web-based listening test shows that sometimes the volume of two guitar samples were not in the same level. The reason lied in the change of equalization (EQ) settings. When the values of three EQ settings (bass, mid and treble) are very low, the volume of sound is lower than the sound with high values of EQ settings. Therefore, the improvement is to set all the audios in the same volume.

Another shortcoming is in my listening test website. Because two riffs were selected to carry out the listening test, I had two sources for web users to do the test. However, first ten questions provided the first riff and the next ten questions provided the second riffs. In terms of user experience it was not good enough because listening to same riff several times would make them boring. One improvement can focus on the intersecting of different riffs in twenty questions. Besides, because there were two source sound data with the same parameters' combinations, actually I should merge the results together. But current algorithms made users pay double efforts to sort all the sounds. Besides, due to the remedial avoiding pair algorithm, the calculation speed slowed down when the data quantity became larger. Thus to test the algorithms and design them earlier is very important.

Lastly, due to the huge sound database in the listening test website, all the audios were not fully sorted. That's to say, there were still some audios had zero weights. As Section 3.3.2 shows, in totally 500 combinations of four parameters, even comparing any two of these required C_{500}^{2}=124750 comparisons. Therefore one way to solve this issue is to optimize the algorithms into a more efficient level so that to fully sorted all the audios in the database. However, the more realistic way is to reduce the sample space. For example if drive parameter was removed, then the rest three parameters (bass, mid and treble) had $5\times5\times5=125$ combinations. It will be easier to sort these settings' combinations and do further analysis.

Chapter 5: Conclusion and Further Work

Aiming to derive an objective measurement of "heaviness" in guitar timbre, I scripted a digital audio workstation to carry out my first Just Noticeable Difference (JND) experiment. According to the JND results, I determined the step-sizes of four amplifier parameters (drive, bass, mid and treble), which mainly affected the perceived heaviness in guitar tone. Afterwards, I programmed the digital audio workstation to produce the sound database automatically.

Then I constructed a website for the online listening test. Specific algorithms were designed for rating the heaviness. This web-based listening test was widely spread through the Internet and got 893 registered users. Therefore sufficient data from the website were acquired for further analysis.

The pattern was shown when visualized data was plotted. Further statistical analysis illustrated that both drive and bass parameter had a positive correlation with heaviness. However, for mid and treble parameter, U-turn curves appeared. At two edge points (0.1 and 0.9), the level of heaviness was considered to be higher than that in the middle (around 0.5).

As for the problems I faced, the first one is scripting the software REAPER. Because this software is not open source, sometimes I couldn't achieve some specific functions in the programming level. So I had to hack into the software and play some tricks to make it work. But overall I learnt how to control it.

Another problem is the balance between the effective classical sorting algorithm and my own algorithms. Some classical sorting algorithm like Quicksort [11] is more efficient than my algorithms. However, the situation in the listening test was different people had different views toward the perceived heaviness. And my algorithms could provide double checks for the sorting and give general results. Besides, multiple users should be supported to do the test at the same time. So the solution was to use my reliable but not so efficient algorithms.

If I could do the project again, I would pay more attention to the critical path of my project. After scripting REAPER and generate my sound database, I would then construct the listening test website. That's because the algorithms in the website were really important to be tested before inviting a number of users to do the test. And when testing the efficiency of website algorithms I could carry out the JND test, meanwhile the website could be modified and the algorithms could be optimized. As soon as the JND test was finished I could generate the sound database and go on the web test. This way could be more time-saving and so I would have sufficient time to do further analysis of the results.

As for the future work, the analysis can go further. Apart from the graphical representation, the statistical analysis in Figure 19 on page 31 can only describe the relationship between one single parameter values and “heaviness”, but the “heaviness” is actually affected by the combination of multiple parameters. For example, as the Figure 18 on page 30 shows, the greatest weight is located on the settings with drive = 0.5, bass=0.9, mid=0.1, and treble=0.9. So an advanced analysis is needed to show the relationship between the combination of several parameters and “heaviness”.

Further signal processing of audios in certain heaviness level can provide the information for pattern reorganization. With the results perhaps some machine learning techniques can be applied. Then combining the data-driven test with appropriate signal processing methods, the results can be used to match a new distorted guitar sound to how 'heavy' it is or can be used to other relevant retrieval systems.

References

[1] Waves Audio Ltd. From Waves: developer of signal processing solutions: http://www.wavesgtr.com

[2] Cockos Incorporated. From Reaper: digital audio workstation: http://reaper.fm

[3] *Weber's Law of Just Noticeable Difference.* From University of South Dakota.: http://people.usd.edu/~schieber/coglab/WebersLaw.html

[4] Berger, H. M. (1999). *Metal, Rock, and Jazz: Perception and the Phenomenonlogy of Musical Experience.* Hanover, N.H.: Wesleyan University Press/University Press of New England.

[5] Fales, H. M. "Heaviness" in the Perception of Heavy Metal Guitar Timbers.

[6] fansite, T. W.-A. In *The Who's Marshall History.*

[7] Guitar amplifier. From Wikipedia: http://en.wikipedia.org/wiki/Guitar_amplifier

[8] Pittmann, A. (2003). The Tube Amp Book. Hal Leonard.

[9] Django. *The Web framework for perfectionists (with deadlines)*. From https://www.djangoproject.com/

[10] Linode - Xen VPS Hosting. From Linode - Xen VPS Hosting: https://www.linode.com/

[11] Sedgewick, R. (1978). *Implementing Quicksort programs. Comm. ACM* **21** (10): 847–857. doi:10.1145/359619.359631

Acknowledgement

Firstly I want to thank my supervisor Christopher Harte, who helped me a lot in my project. Also with his assistance, I gained my confidence in public speaking.

Then I really appreciate Dr. Dawn Black, who gave me lots of advice in this research. She helped me to record many guitar riffs, without these guitar samples my sound database cannot be achieved.

Lastly I want to show my gratitude to 20 volunteers who took part in my JND experiment, and to those 893 web users who did my listening test.

Appendix

北京邮电大学
本科毕业设计（论文）任务书
Project Specification Form

学院 School	International Scho	专业 Programme	Telecommunications	班级Class	2009215107
学生姓名 Name	ZHANG Mincong	学号 BUPT student no	9212915	学号 QM student no	90465589
指导教师姓名 Supervisor	HARTE Christopher	职称 Academic Title	Teaching Fellow		
设计（论文）题目 Project Title	How heavy is heavy?				
题目分类 Scope	Research	Software	Computer Software		

主要任务及目标Main tasks and target:	By
Task 1: produce database of different guitar tones using a number of clean recordings and a number c	December 2012
Task 2: Run a listening test, perhaps as a web-based game to find human ratings for heaviness of gui	March 2013
Task 3: Analyse results to try to find correlations between parameters and heaviness ratings	March 2013
Task 4: Try to find algorithm that can give a close matching result to the percieved values (possibl	April 2013

Measurable outcomes
1) In depth study of problem culminating in production of database of guitar sounds
2) Perceptual listening study giving human ratings for heaviness
3) Derivation of objective measure based on listening study results

主要内容Project description:

The electric guitar has been with us since the 1950s. Through the 60s, 70s and 80s, as styles developed from rock'n'roll through heavy rock to heavy metal, the tones produced by guitarists became progressively more 'heavy'. This notion of 'heaviness' in a guitar tone depends on a number of features including the style of playing, the equalisation curve and the level of harmonic distortion applied to the sound. In this project the aim is to investigate these parameters and try to derive an objective measure for heaviness that could be used as a feature in music information retrieval systems. A perceptual listening test will be carried out to try to grade the level of 'heaviness" as percieved by human listeners. The results from this can be used as a basis for an objective measure.

Project outline

To start with, I will search the background of the 'heaviness' of guitar sound. Although the 'heaviness' is affected by many factors, mainly it is the distorted guitar sound that makes the music sounds heavy. So I can build a timeline of typical guitar songs and their parameters to find out the development of guitar distortion and 'heaviness'. Also I will look into how the valve amplifier works and how to create distorted sounds. And research in this, will make contributions to music retrieval.

Afterwards, I will use some Virtual Studio Technology (VST) plug-ins to simulate the guitar distortion sounds. I shall input a piece of recorded clean guitar sound, containing typical music paragraph, into VST plug-ins, and generate various distorted sounds by adjusting parameters, such as 'bass', 'middle' and 'treble'. I will use a digital audio workstation called REAPER, which is scriptable with Python, to adjust parameters automatically through accessing internal actions and some APIs.

Obtaining the sound database, I will use Django, a high-level Python Web framework, to design a website. This website is used for classifying the distorted guitar sounds through a well-designed interface, asking the subjects to choose which sound is heavier. Besides, I must make sure every sound track to be shown and compared evenly.

Lastly, I will analyze the results. It is expected to find out some interesting distribution from 'clean' to 'heavy' within the parameters using some machine learning techniques, but it all depends on the experiment results. Then combining the data-driven test with appropriate signal processing methods, the results can be used to match a new distorted guitar sound to how 'heavy' it is or can be used to other relevant retrieval.

Reference link
on REAPER: http://wiki.python.org/moin/PythonInMusic; http://wiki.cockos.com/wiki/index.php/ReaScript;
on Django: http://www.djangobook.com/en/2.0/index.html; https://www.djangoproject.com/

Project ID QCAH10 How heavy is heavy?

	Nov		Dec		Jan		Feb		Mar		Apr		May	
Task 1: produce database of different guitar														
Background searching	■	■												
Learn to control VST in Python		■	■											
Simulate different guitar distortion sounds			■											
Task 2: Run a listening test, perhaps as a														
Setup a website by using Django				■			■							
Design frontend of the website							■	■						
Run the listening test								■	■	■				
Task 3: Analyse results to try to find														
Find correlations in parameters									■					
Pattern analysis									■	■				
Task 4: Try to find algorithm that can give														
Data-driven test										■	■			
Match results										■	■			
Write draft report			■	■			■	■	■	■				
Write final report										■	■	■	■	■

Risk Assessment

This project has no need of hardware implement so the risk mainly focuses on unfamiliar software REAPER. Also this software is unstable so it is possible that some of the functions cannot be achieved. Another risk is the remote web server, because the quality of the web server cannot be predicted until the website is setup. Lastly my laptop might have crushed so the code and important data might be lost. But thanks to my supervisor, we all have backups in the remote repository.

Table 2: Risk Assessment

Description of Risk	Description of Impact	Likelihood Rating	Impact Rating	Preventative Actions
The software REAPER cannot be scripted to achieve the required functions	JND experiment cannot be conducted and sound database cannot be generated	Likely	Serious	Find another scriptable software or use another tools
Web server do not support Django web framework	Website cannot be setup	Likely	Serious	Find method to configure the environment for Django
Unstable Server	Less users to do the listening test	Likely	Serious	Change web server
Computer crush	Files are lost	Likely	Minor	Use the backup files in repository

Environmental Impact Assessment

Since this project is a research and it mainly requires software development without any hardware, the cost of manufacture, as well as waste disposal and recycling, can be neglected. In addition, this project only needs electricity for my laptop and perhaps the web server, this project has no environmental impact.