University of Virginia

Project for CPE 7993: Independent Study

Generation of Non-repetitive Sound Effects

Zhidan Luo

September 2019

1 Introduction

1.1 Background

Game players nowadays have had an increasingly fastidious attitude towards video game in many aspects, such as game audio, graphics, diegetic background, innovation and so on, due to the significant development of internet communication, hardware support and game platform, and other factors. This phenomenon inspires the game companies' motivation to manufacture delicate and high-quality video games. Additionally, the fierce competition between various game firms also contributes them to achieve the ultimate goal, producing as elaborate video games as possible, in order to gain favour among the numerous game players.
One important and dominant respect that a game producer must think about when designing a game is the game audio. Game audio plays a so important role in a video game. When a player is enjoying a video game, the audio would directly influence a player's acoustic perception and thusly, also the game experience. If the audio is perfectly impressive to the player, perhaps he would deeply immerse in playing the game, feeling that he is likely in the atmosphere the game constructed, as well as give a high appraisal to the game later, also possibly recommending it to his friends. Otherwise, the defective or even annoying audio might stimulate the player to turn off the built-in audio and turn on the music of his own, which would generally diminish the effectiveness of the game's impact and dominance. In short, the importance of game audio cannot be ignored when designing a game.

1.2 Motivation

Game audio have three main categories, which are sound effects, background music and character dubbing respectively. Here we primarily focus on the solutions to solve the problems for the sound effects.
With the rapid improvement of storage capacities of console, personal computer and mobile phone, game designers are now able to put high quality or even CD-quality sound effects in their games in order to obtain the most realistic experience for game players. In such a case, pre-recorded versions of sound samples are usually adopted by game developers and therefore, a large amount of pre-recorded ones are required in order to cover situations as many as possible, giving the players as various experience as possible when running a game. However, it is almost impractical to take in pre-recorded sound samples for all events in a game because of memory constraints, especially for a large game. In this way, frequent playback of the limited sound samples has now become a normal phenomenon.
However, one of the most serious problems resulted from by employing such approach is the endless repetition of the same sounds. Such inflexible and annoying repetition would generally lead a player to be tedious of the game sound effects, which certainly, would cause the abatement of his or her game experience.
Fortunately, the problematic issues have now drawn increasingly attention of designers and multifarious technologies and researches have applied to it to solve the problem, sound effects having a very repetitive and limited substance.

1.3 Project overview

The aim of this report is to develop new algorithms that will generate non-repetitive sound effects using parameters from user self-defined. This approach aims to use audio grains to create finely-controlled synthesised sounds, which are based on recorded sound samples. [1]

[1] The code for testing is available at `https://github.com/Dan-Animenz/CPE7993/tree/master/sound_generation`.

2 Algorithm Overview

The block diagram of the whole processing procedure of the project is shown in Figure 1. This report will be based on the sound sample 'TIC-fing-50.wav' to shown results as an example.

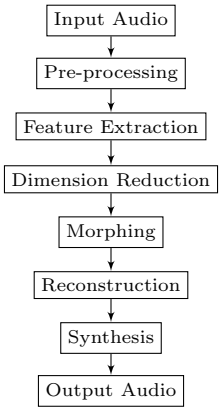

Figure 1: Audio Processing Algorithm

3 Pre-processing

The input audio file is a continuous signal with several sound events that actually are transient signals. Pre-processing is to divide the continuous signal into several short continuous ones with only a single sound event. All sound events are samples to extract their features.
At this stage, peak detection is the key to do segmentation. Figure 2 shows the result of peak detection. Figure 3 shows one extracted sound event.

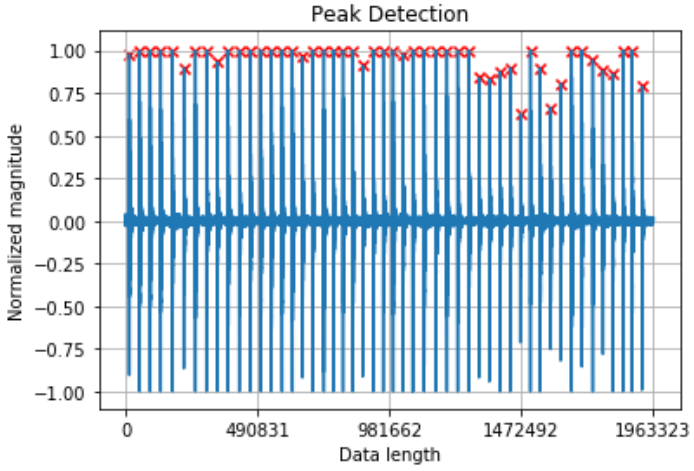

Figure 2: Peak Detection

The most significant point to notice is that every sound event must be exactly the same length. In this way, sound events are able to be represented in matrix form. If divided sound events are not in the same length, zeros should be added in order to equal to the largest length of the sound event.

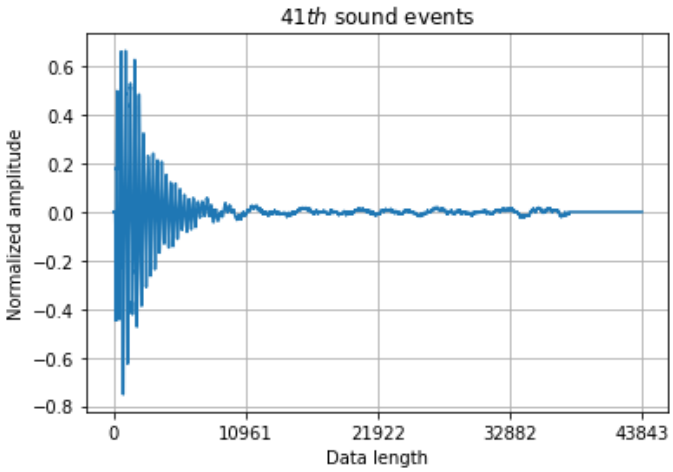

Figure 3: Extracted Sound Event

4 Feature Extraction

This step is to extract the main features of sound events. The characteristic information is of great help to find the similarities and differences between the sound events. Besides, the features are the keys to morph the audio.
The main technique adopted in the step of feature extraction is discrete wavelet transform (DWT for short). It is a new method of transformation analysis, developed from Fourier transform and short-time Fourier transform. More details about algorithms of multi-level DWT are shown in Figure 4.

For example, starting from the signal $x[n]$ with length N in the first level, convolve it with two filters, high-pass filter H_{iD} and low-pass filter L_{oD} respectively. Then do down-sampling on both of the convolved signals to keep the even indexed elements. Finally, level-1 approximation coefficient cA_1 and level-1 detail coefficient cD_1 are computed. Level-2 coefficients cA_2 and cD_2 are computed by starting from level-1 approximation coefficient cA_1, convolving with the same filters and doing down-sampling. Level-3 or higher-level coefficients are computed in the similar way. Generally, the approximation coefficient of the last level and all of the detail coefficients are the needed signal features. For example, if it is a level-4 DWT, then cA_4, cD_4, cD_3, cD_2 and cD_1 are the feature information to morph.
There are two main reasons to choose DWT.

- On one hand, the sound samples are all from our real world. They are all nonstationary signals. Wavelet transform are more capable than Fourier transform, short-time Fourier transform or other similar transform at this stage.
- On the other hand, it is noticeable that one sound event is actually a transient signal. There is Gibbs effect when applied Fourier transform on a transient signal. A transient signal decays so quickly that numerous sinusoids are needed to fit it. When applied wavelet transform to it, Gibbs effect will never occur because it is wavelet transform that provides the decayed wavelet to convolve with the transient signal.

5 Dimension Reduction

This step is to use principal component analysis (PCA for short) to reduce the dimension of the sample features.
PCA is one of the statistical methods, usually used for extracting the common features of multi-level samples. The key ideal of it is to reduce the dimensionality of the samples, selecting and keeping the important information whose features are obvious.
PCA constructs an orthogonal base to ensure that the relationship between the features is as few as possible. Every sample is represented by the linear combination of the base. The remainder of this section will explain how to construct the orthogonal base. Besides, more basics and details about PCA could be found at another tutorial I

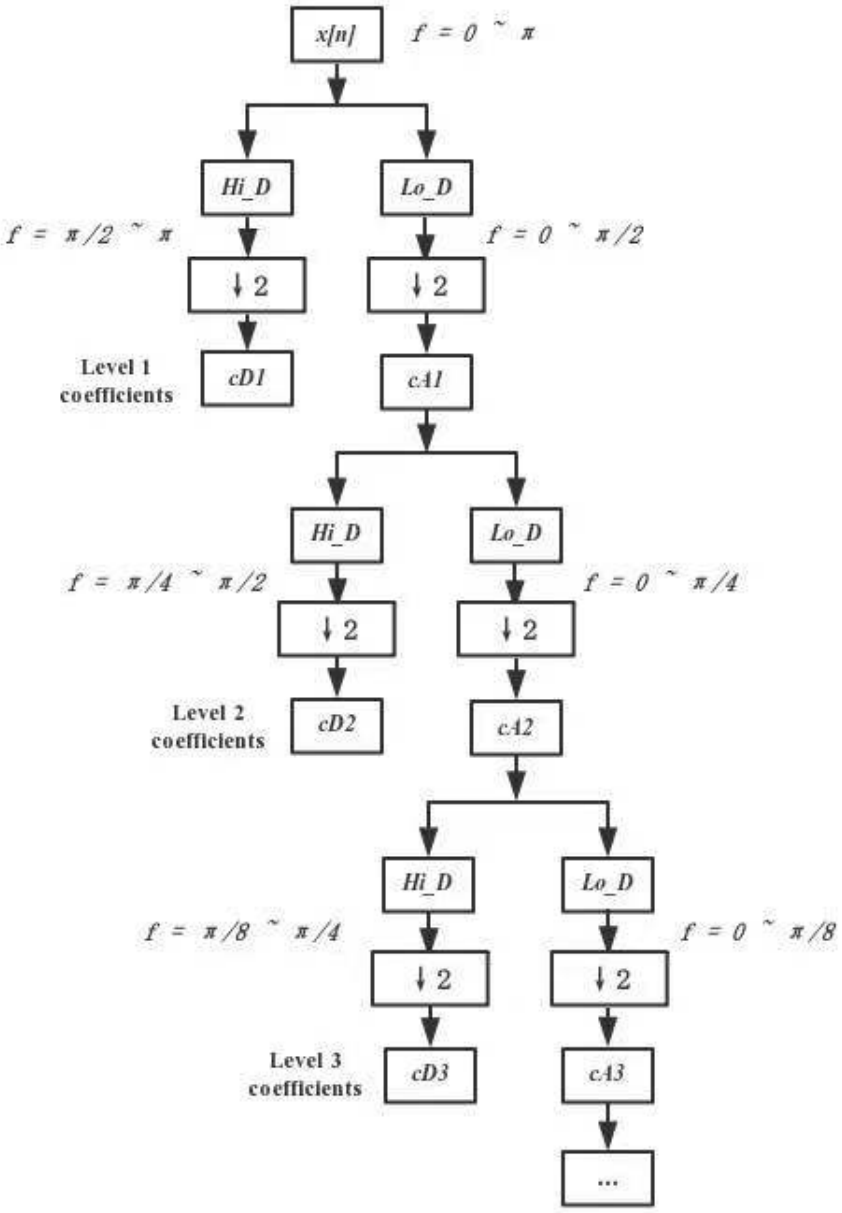

Figure 4: Algorithms about Multi-Level DWT

organised. [2]
PCA constructs an orthogonal base to ensure that the relationship between the features is as few as possible. Every sample is represented by the linear combination of the base. The remainder of this section will explain how to construct the orthogonal base.

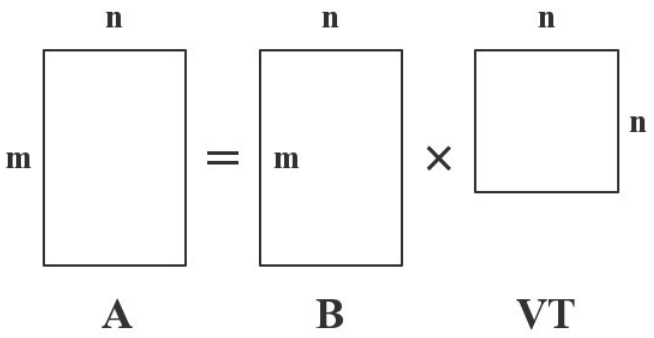

Figure 5: Algorithms about PCA

Suppose the sample matrix is A with size m-by-n. Every row is a sample with m samples in total and every column is a feature with n features in total. If two variables are unrelated, their covariance have to be 0. That is,

$$\text{cov}(X, Y) = E([X - E[X]][Y - E[Y]]) = 0 \tag{1}$$

At this stage, the sample matrix could transform to a new matrix whose every column is unrelated through a certain linear transform. To simplify the problem, consider two random variables X and Y with their mean values of 0. Then their covariance is,

$$\text{cov}(X, Y) = E([X - 0][Y - 0]) = E(XY) \tag{2}$$

This means their dot product has to be 0 to if they are unrelated. Therefore, means of every column are firstly computed. Besides, all elements in each column are subtracted by the mean of the current column to confirm the means are 0. In this way, every column of matrix B obtained from matrix A through a certain linear transform is orthogonal. Represent B in matrix form is,

$$B^T B = D, \text{ where } D \text{ is a diagonal matrix} \tag{3}$$

Suppose the transform is $AM = B$ and substitute this for the above equation. That is,

$$\begin{aligned} &(AM)^T(AM) = D \\ \Longrightarrow &M^T A^T AM = D \\ \Longrightarrow &A^T A = \left(M^T\right)^{-1} DM^{-1} \end{aligned} \tag{4}$$

In the last formula of equation (4), $A^T A$ is a diagonal matrix. Therefore, if apply eigen-decomposition on it, $A^T A = VDV^{-1}$ will be obtained and V will be an orthogonal unit matrix. Then $V^T = V^{-1}$ will be obtained and thus this matrix V meets the requirement for M. In short, the steps to apply PCA on matrix A are:

1) To normalize every column of A so that every column's means of the normalized matrix A are 0;

[2]The tutorial is available at `https://github.com/Dan-Animenz/SYS6018-Data_Mining/tree/master/final_tutorial`.

2) To calculate the eigenvalues of $A^T A$, $D = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ and eigen-matrix V;

3) To make matrix $B = AV$.

This new matrix B is the needed matrix where means of every column are 0 and every column is orthogonal. At this stage, $A = BV^T$ where every row of V^T is the needed feature and every row of B is the combination coefficients of features. B is the presentation of X in the principal component space. V is called transformation matrix or projection matrix.
In this project, PCA will be applied on detail coefficient sets and approximation coefficient from the set. The purpose is to reduce the number of sound events of every signal feature extracted by DWT and it does not lost the integrity and accuracy of the sound event sets.

6 Morphing

This section details the algorithms of morphing of sound events. Figure 6 shows the basic steps of morphing of the wavelet coefficients.

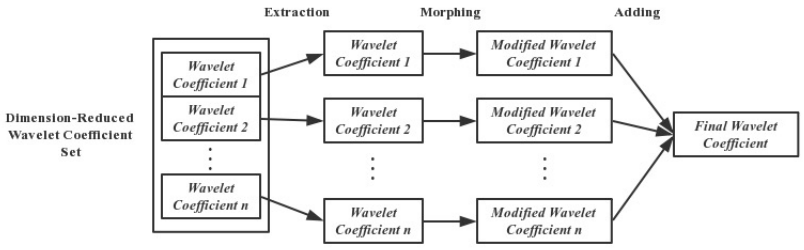

Figure 6: Procedure of Morphing

Starting from one of the dimension-reduced wavelet coefficient sets, the first step is to extract the wavelet coefficients in a set. The next step is to morph each extracted wavelet coefficient.
The specific implementation of morphing is to add random numbers within a controllable interval to every data of a wavelet coefficient. To be specific, suppose n_i is one original data of a wavelet coefficient and m_i is the morphed data of n_i. The definition of morphing is shown below.

$$m_i = n_i + |n_i| \times \text{ Random Number} \tag{5}$$

It is noticeable that the data in latter parts of every wavelet coefficient is greatly close to 0 and they are of actually little influence. If these data has been added with a large random number, it might result in unnecessary noise in the morphed wavelet coefficients. In order to solve this problem, the random number has been multiplied with the magnitude of the data before adding with the data to eliminate the possible noise from latter parts of wavelet coefficients.

7 Reconstruction and Synthesis

This step is to utilize the morphed signal features to synthesise the output signal. The basic theory of this step is Inverse Discrete Wavelet Transform (IDWT for short). If n-level DWT has applied on signals at feature extraction step, then n-level IDWT will be applied on the morphed signal features from the modelling step at the reconstruction step. More details about algorithms of level-j IDWT are shown in Figure 7.

First do upsampling on the approximation and detail coefficients cA_j and cD_j at level-j respectively to insert zeros at odd indexed element. Then the upsampled signals are convolved with the same filters as in the DWT, high-pass filter H_{iD} and low-pass filter L_{oD} respectively. Finally take the central part of the convolved signals with the convenient length to construct cA_{j-1}. Continue to construct cA_{j-2} using cA_{j-1} and cD_{j-1} until cA_0 is computed.

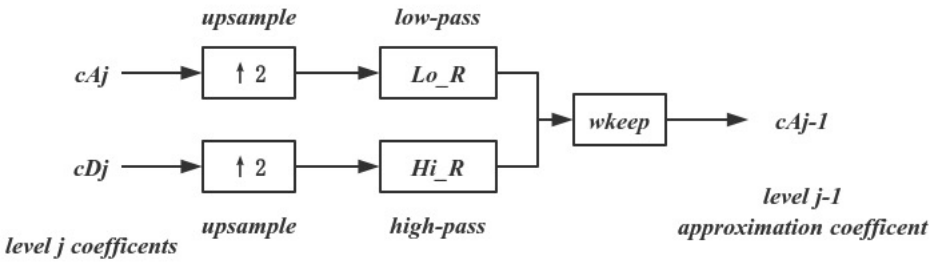

Figure 7: Algorithms about Level-j IDWT

cA_0 is the reconstructed sound event.
After this, it is better to do normalization on the events before storing them. That is,

$$\text{Normalized Signal} = \frac{\text{Reconstructed Signal}}{\max\{|\text{Reconstructed Signal}|\}} \tag{6}$$

Then randomly combine reconstructed sound events, the synthesised sound audio are obtained.

8 Results and Conclusions

Figure 8 shows one synthesised sound audio combined by 5 reconstructed sound events and Figure 9 shows the corresponding spectrum diagram.

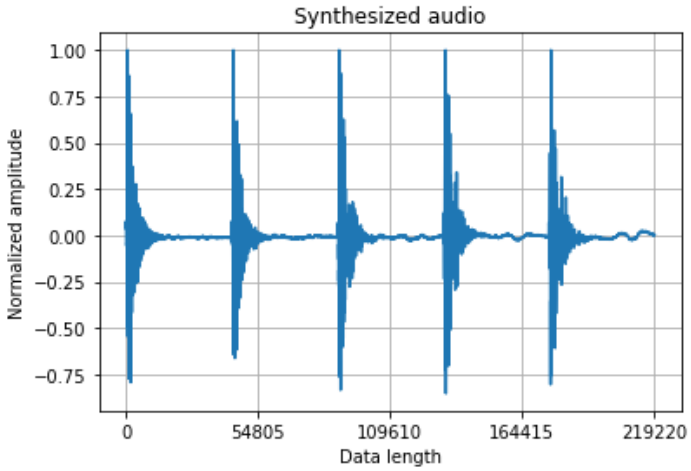

Figure 8: Synthesised Sound Audio

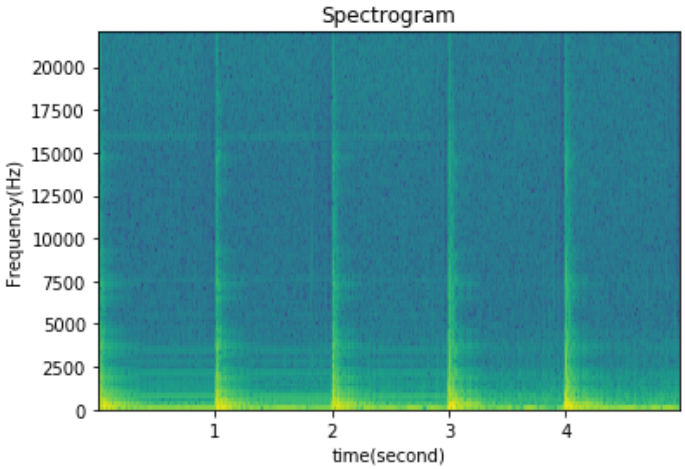

Figure 9: Spectrum Diagram

This project has successfully developed new algorithms in Python that will generate numerous non-repetitive sound effects only based on a small amount of recorded sound samples. Therefore, it has solved two major problems when companies are developing a game:

- monotonous repetition of pre-recorded sound effects
- memory constraints due to a large number of high quality sound effects

Designers are able to generate specified sound effects by morphing parameters of different characteristics.

9 Suggestions for Further Work

In the future, it is the plan to continue the work of synthesizing sound effects. At this stage, not only one kind of sound effects are synthesized, but also several sound effects are reconstructed. For example, Figure 10 illustrates the relationship between four different kinds of sound effects.

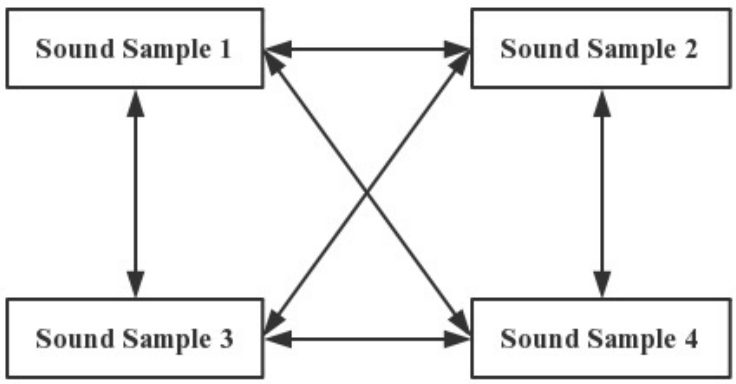

Figure 10: Relationship between Four Different Kinds of Sound Samples

Based on the availability of transformation between them and generation of non-repetitive sound effects of a specified one within itself, it is also able to generate a sound effect between them. If sound sample 1 is the gunshot

and sample 2 is the sound of artillery, it can generate a sound mixed by them and additionally the degree of the two elements from the samples are controllable by parameters.
This can be done by singular value decomposition. In principal component analysis, the sample data matrix is decomposed to the product of two matrixes. In singular value decomposition, it is decomposed to three matrixes, which is the biggest difference between the two models. The ideal is promising. If the algorithms are developed, it will be more flexible and powerful.