Single tone fingerprint analyse and reconstruction for creating highly dynamic sound libraries

Karsten Dehren

April 23, 2017

Contents

Chapter 1

Motivation

During the development of a note editing software I had to decide between using a standard sound library based on wave (.wav) files or to develop an own format.
The benefits of using a sound library which is based on standard wave files are first of all the easy set up of these files and then that for basic tones no additional calculations are necessary. But the disadvantages are these libraries are usually inflexible and hard to edit at runtime and apart from that it takes a relative long time to load the files.
In order to avoid these problems this paper will define a new strategy to manage and create sound libraries including a file format to define a fingerprints of each tone and an optimized non-redundant data structure.

1.1 State of the art

The probably most common way of creating a sound library is a list of native wave files which contain the – in average 3 to 5 second long – information of one sound. After being loaded the data can combined with existing informations by adding to overlay sounds and concatenation to create a sequence of sounds or an entire song.
This strategy works perfectly fine as long as there are no sound modulations necessary. But if the user wants to change the note – for example a bended note on a guitar or a vibrating tone – the software has to recalculate the file or to dynamically drop data segments to adjust the frequency. Neither of these strategies can maintain the sound quality because of a quick recalculation of sound files can't start a detailed analyse what leads to a sound calculation with flawed or maybe even wrong parameters. And beside of that dropping data segments will directly influence the sound quality by ignoring details that could have had influenced the sound.

1.2 Benefits

The Dynamically Transformable Single Tone Fingerprint file system and the Cloud Repository data structure offers developers an interface to build a flexible and composite sound library which enables users to add own sounds to the application by recording only one tone. The fingerprint – extracted from the recorded tone – can be transferred into every note and in addition to that tone and note modulations cause just a minor increase of the calculation efforts. Apart from that developers are able to create and add plug-ins to customize the functionalities and overall behaviour of the underlying system.

Chapter 2

Definitions

2.1 Musical definitions

This section defines common musical terms and the context they will be used in. These might not be the standard definitions – because in normal use this terms have a logical overlapping – but in this context it is important to differentiate between them to understand the following explanations and calculations.

2.1.1 Amplitude

An Amplitude $\mathcal{A}$ is a factor which defines the starting volume of a sine wave. It can only take values between 0 and 1.

$$\mathcal{A} \subset \mathbb{R}, \quad \mathcal{A} = [0, 1]$$

2.1.2 Sine wave decay function

The decay γ is a linear or exponential function which defines how fast a sine wave will decay over time.

$$x \in \mathbb{N}, \quad \gamma(x) = m * (-x) + b \ \vee \ \gamma(x) = e^{f(x)}$$

2.1.3 Tone

A tone is a single sine wave with or without decay function. In addition to that it has an undefined factor or function which defines the wave amplitude.

$$T(x) = \mathcal{A} * \gamma(x) * sin(x) \quad x \in \mathbb{N}$$

It can be combined with other tones by adding to overlap two tones.

$$f : T \times T \to T, \ A, B \mapsto A + B, \quad A, B \in T$$

$$A, B \in T \qquad f(A, B) = \sum_{i=0}^{\max\{|A|,|B|\}} A(i) + B(i)$$

2.1.4 Modulated Tone

A modulated tone is a specialization of standard tones. In addition to the tone properties it can have a frequency which is changing by time. The factor which changes the frequency will be defined in at least one additional function. IF the factor is defined by multiple functions they have to have a starting and terminating index to avoid function overlapping ambiguous information.

$$T(x) = \mathcal{A} * \gamma(x) * sin(f(x)), \quad x \in \mathbb{N}$$

2.1.5 Note

A note ϕ is a vector of standard and modulated tones which will be played at the same time. These tones can be read individually or as unit what will transform them into a complex tone. Beside of that the note defines the starting amplitudes of each tone.

$$\phi = \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{pmatrix}$$

2.1.6 Complex tone

A complex tone $\widehat{\phi}$ is the sum of tones from one note.

$$\widehat{\phi} = \sum_{i=1}^{n} T_i, \quad T_1, \ldots, T_n \in \phi$$

2.1.7 Sound

A Sound Φ is matrix Notes which defines their concatenation. It can also be defined as vector of complex tones. In this case it will be called $\widehat{\Phi}$.

$$\Phi_{m,n} = \begin{pmatrix} \phi_{11} & & \phi_{1n} \\ & \ddots & \\ \phi_{m1} & & \phi_{mn} \end{pmatrix}$$

$$\widehat{\Phi}_n = \begin{pmatrix} \widehat{\phi}_1 & \ldots & \widehat{\phi}_n \end{pmatrix}$$

2.2 Sound fingerprint format

2.2.1 Contained information

2.2.2 Data composition

2.3 File format .DTSTF

2.3.1 File allocation

2.3.2 File Header

2.3.3 File Body

2.4 Simplifying definitions

2.4.1 Analyse data regions

Chapter 3

Backgrounds

3.1 Physical Background

3.1.1 Sinus wave behaviour

3.1.2 Fourier transformation

3.2 Musical theory

3.2.1 Notes in general

3.2.2 Harmonic series

3.3 .WAV file format

3.3.1 Basic structure

3.3.2 Data calculation

3.3.3 Data validation

Chapter 4

Analyse

4.1 Data allocation

4.2 Base frequency calculation

4.3 Amplitude calculation

4.4 Exponential decay definition

4.5 Fingerprint construction

Chapter 5

Reconstruction

5.1 Fingerprint analyse

5.2 Sine wave calculation

5.3 Amplitude calculation

5.4 .WAV file construction

Chapter 6

Programming

6.1 Objects

6.2 Data structure

6.3 Data behaviour

6.4 Reconstructing calculations

Chapter 7

Conclusion

Chapter 8

Sources