



SINGLE NOTE FINGERPRINT ANALYSE AND RECONSTRUCTION FOR CREATING HIGHLY DYNAMIC SOUND LIBRARIES

Paper

for defining a new way of creating dynamic
and customizable sound library

Karsten Dehren

April 2017

Contents

1	Motivation	3
1.1	State of the art	3
1.2	Benefits	3
2	Definitions	4
2.1	Basic Definitions	4
2.1.1	Sampling Rate	4
2.1.2	Concert pitch	4
2.2	Musical definitions	5
2.2.1	Amplitude	5
2.2.2	Decay function	5
2.2.3	Harmonic Series	5
2.2.4	Frequency calculation	6
2.2.5	Tone	6
2.2.6	Modulated Tone	6
2.2.7	Note	6
2.2.8	Complex note	7
2.2.9	Sound	7
2.3	Sound fingerprint	8
2.3.1	Contained information	8
2.3.2	Data composition	8
2.3.3	File outlining	9
2.4	File format .DTSTF	10
2.4.1	File allocation	10
2.4.2	File Header	10
2.4.3	File Body	10
2.4.4	Extended file outlining	10

1 Motivation

During the development of a note editing software we had to decide between using a standard sound library based on wave (.wav) files or developing an own format.

The benefits of using a sound library based on standard wave files are first of all the easy set up of these files and then that for basic notes no additional calculations are necessary. But the disadvantages are that 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 instrument 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 – data of one note. After being loaded the data can combined with existing informations by adding to overlay notes and concatenation to create a 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 Note Fingerprint file system and the Cloud Repository data structure offers developers an interface to build a flexible and composite sound library which enables users and developers to add own sounds to the application by recording only one note. The fingerprint – extracted from the recorded note – can be transferred into every note and in addition to that tone and note modulations cause just a minor increase of the calculation effort. Apart from that developers are able to create and add plug-ins to customize the functionalities and overall behaviour of the underlying system.

2 Definitions

2.1 Basic Definitions

2.1.1 Sampling Rate

The Sampling rate SR defines the number of data units played every second. It will have the default value – like a wave file sampling rate – of 44100 data units per second.

2.1.2 Concert pitch

The concert pitch is a reference frequency to tune a group of instruments. From now on it will be called base frequency f_0 with the default value of

$$f_0 = 440Hz$$

2.2.4 Frequency calculation

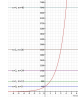
$n \in \mathbb{Z}$

$f_0 = 440\text{Hz}$ Octave:

$$f_n = f_0 * 2^n$$

$m \in \mathbb{Z}$ 12 half tone steps per octave

$$f_m = f_0 * (\sqrt[12]{2})^m \quad (2.14)$$



2.2.5 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.

$$T(x) = \mathcal{A} * \gamma(x) * \sin(x) \quad x \in \mathbb{R} \quad (2.15)$$

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

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

$$A, B \in T \quad f(A, B) = \sum_{i=0}^{\max\{|A|, |B|\}} A(i) + B(i) \quad (2.16)$$

2.2.6 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.

$$T(x) = \mathcal{A} * \gamma(x) * \sin(f(x)), \quad x \in \mathbb{R} \quad (2.17)$$

2.2.7 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 note. Beside of that the note defines the starting amplitudes of each tone.

$$\phi = \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{pmatrix} \quad (2.18)$$

2.2.8 Complex note

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

$$\hat{\phi} = \sum_{i=1}^n T_i, \quad T_1, T_2, \dots, T_n \in \phi \quad (2.19)$$

2.2.9 Sound

A Sound Φ is matrix of Notes which defines their concatenation.

$$\Phi_{m,n} = \begin{pmatrix} \phi_{11} & & \phi_{1n} \\ & \ddots & \\ \phi_{m1} & & \phi_{mn} \end{pmatrix} \quad (2.20)$$

It can also be defined as vector of complex notes. In this case it will be called $\hat{\Phi}$.

$$\hat{\Phi}_n = (\hat{\phi}_1 \quad \dots \quad \hat{\phi}_n) \quad (2.21)$$

2.3 Sound fingerprint

Sound fingerprints are the base for .DTSTF files. It contains the result of a wave analyse which will be defined later on. As a result of that it contains neither the analytical calculations nor a reconstruction definition.

2.3.1 Contained information

A sound fingerprint contains several informations. First of all the number of required sine waves what defines how many sine waves will be needed to create an appropriate note. In second place the wave definitions including a frequency and amplitude factor for each sine wave. And last of all the decay functions which will be defined by a function type and a decay factor.

2.3.2 Data composition

Number of sine waves:

This value will be defined by three numbers which declare how many sine waves – with priority high, medium and low – are required to create an appropriate note.

Frequency factor:

The frequency factor f defines a value to modify a predefined base frequency \mathcal{F} .

$$\begin{aligned} & \sin(f * \mathcal{F}) \quad f \in \mathbb{N}, \mathcal{F} \in \mathbb{R}_{>0} \\ (2.15) \Rightarrow \quad T(x) &= \mathcal{A} * \gamma(x) * \sin(f * \mathcal{F}) \quad x \in \mathbb{R} \end{aligned}$$

Amplitude and decay function

These values correspond to the definitions (2.1) and (2.2).

2.3.3 File outlining

Section	Priority	Data
Number of required sine waves	High	4
	Medium	3
	Low	6
Section	Frequency	Amplitude
Frequency and amplitude factor	2	0,8
	4	0,6
	5	0,7
	8	0,6
	1	0,55
	3	0,5
	9	0,4
	6	0,3
	7	0,4
	10	0,35
	11	0,2
	12	0,12
	13	0,24
Section	Type	Value
Decay function type and value	Exponential	1.3
		1.5
		1.8
		1.1
		1.9
		1.6
		2.2
		1.9
		1.0
		2.1
		2.6
		1.4
		1.7

(2.22)

The data column shows information from the DTSTF format. The other two columns are for description.

2.4 File format .DTSTF

2.4.1 File allocation

2.4.2 File Header

2.4.3 File Body

2.4.4 Extended file outlining