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

Summary

[Motivation 3](#_Toc480255371)

[State of the art 3](#_Toc480255372)

[Area of usage 3](#_Toc480255373)

[Benefits 3](#_Toc480255374)

[Definitions 4](#_Toc480255375)

[Musical definitions 4](#_Toc480255376)

[Tone 4](#_Toc480255377)

[Note 4](#_Toc480255378)

[Sound fingerprint 4](#_Toc480255379)

[Contained information 4](#_Toc480255380)

[Data composition 4](#_Toc480255381)

[File format .DTSTF 4](#_Toc480255382)

[File allocation 4](#_Toc480255383)

[File Header 4](#_Toc480255384)

[File Body 4](#_Toc480255385)

[Simplifying definitions 4](#_Toc480255386)

[Analyse data regions 4](#_Toc480255387)

[Backgrounds 4](#_Toc480255388)

[Physical Background 4](#_Toc480255389)

[Sinus wave behaviour 4](#_Toc480255390)

[Fourier transformation 4](#_Toc480255391)

[Musical theory 4](#_Toc480255392)

[Notes in general 4](#_Toc480255393)

[Harmonic series 4](#_Toc480255394)

[.WAV file format 5](#_Toc480255395)

[Basic structure 5](#_Toc480255396)

[Data calculation 5](#_Toc480255397)

[Data validation 5](#_Toc480255398)

[Analyse 5](#_Toc480255399)

[Data allocation 5](#_Toc480255400)

[Base frequency calculation 5](#_Toc480255401)

[Amplitude calculation 5](#_Toc480255402)

[Exponential decay definition 5](#_Toc480255403)

[Fingerprint construction 5](#_Toc480255404)

[Reconstruction 5](#_Toc480255405)

[Fingerprint analyse 5](#_Toc480255406)

[Sine wave calculation 5](#_Toc480255407)

[Amplitude calculation 5](#_Toc480255408)

[.WAV file construction 5](#_Toc480255409)

[Programming 5](#_Toc480255410)

[Objects 5](#_Toc480255411)

[Data structure 5](#_Toc480255412)

[Data behaviour 5](#_Toc480255413)

[Reconstructive calculations 5](#_Toc480255414)

[Conclusion 5](#_Toc480255415)

[Sources 5](#_Toc480255416)

# 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 setup of these files and then that for basic tones 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 how to manage and create sound libraries including a file format to define a fingerprints of each tone and an optimised non-redundant data structure.

## State of the art

The probably most common way of creating a sound library is a list of native wave files which will be combined, lined up and added at runtime. After entering a new note the related data will be loaded and combined with the existing files to create one application internal sound file which can be played on demand.

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.

## Area of usage

The Dynamically Transformable Single Tone Fingerprint (.DTSTF) format in combination with the Repository Cloud data structure is meant to be used as sound library system. This system can be used in note editor and sequencing software or any other type of software which uses a sound library.

## 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 the 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 the overall behaviour of the underlying system.

# Definitions

The following part will define some notations including basic musical theoretic terms, a short sound fingerprint definition and Dynamically Transformable Single Tone Fingerprint file format.

## Musical definitions

### Tone

### Note

## Sound fingerprint

### Contained information

### Data composition

## File format .DTSTF

### File allocation

### File Header

### File Body

## Simplifying definitions

### Analyse data regions

# Backgrounds

## Physical Background

### Sinus wave behaviour

### Fourier transformation

## Musical theory

### Notes in general

### Harmonic series

## .WAV file format

### Basic structure

### Data calculation

### Data validation

# Analyse

## Data allocation

## Base frequency calculation

## Amplitude calculation

## Exponential decay definition

## Fingerprint construction

# Reconstruction

## Fingerprint analyse

## Sine wave calculation

## Amplitude calculation

## .WAV file construction

# Programming

## Objects

## Data structure

## Data behaviour

## Reconstructive calculations

# Conclusion

# Sources