FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

ELECxxxx Final Report

Using Spectral Mapping to Exploit and Enhance an Object's Latent Harmonicity

Matteo Fabbri
SID 201071767
el16mf@leeds.ac.uk

Reuben Frankel
201042778
el16rf@leeds.ac.uk

Zai'En Pan
201072683
el16zep@leeds.ac.uk

Rhun Gwilym
SID
el16rpg@leeds.ac.uk

Abstract

Spectral mapping is a technique well-documented in the field of audio signal processing, in which inharmonic sounds can be made harmonic and vice-versa. The remapping of sonic information has been used as a compositional tool, where existing musical instruments have been morphed into unusual timbres without affecting their sonic identity. Since the reverse is also possible, this project builds upon the concept of spectral mapping to transform ordinary objects into musical instruments. Existing systems such as Mogees [1] have achieved the conversion of objects into instruments, where sounds are produced as the result of the user executing physical gestures on a surface. It is important to note that interaction with these objects simply triggers user-defined physical modelling synthesiser patches, of which the output sound bears no acoustic resemblance to the input. Spectral mapping provides the possibility of creating an instrument from any object, with a heavy emphasis on the reflection of its original tonal character at the output. Its sonic properties will be analysed, remapped, and reconstructed into a sound that is more harmonic (see section in background research), whilst still retaining qualities of its original form. The result will be a unique instrument that is made consonant through the object's own properties.

Acknowledgements

We would like to thank our project supervisor Mr. David Moore for his unrelenting support on this project. Additional thanks to Dr. William A. Sethares and Dr. Andrew J. Milne for providing many helpful spectral mapping resources, as well as Dr. Gabriele Bunkheila for assisting with MATLAB Audio Toolbox.

Table of Contents

(make sure to update before final submission)

1. Introduction

Background

There are hundreds of tuned musical instruments that exist around the world which have a range of pitches and timbres but in truth, what surrounds us in our daily lives are potential, untapped inharmonic objects whose sounds can similarly be manipulated. The harmonic equivalents of these objects could open up possibilities for new types of instruments. This has been explored by several works including William Sethares' *TransFormSynth* and Mogees Ltd's *Mogees* but do not strike the balance between keeping the original sound's timbre and harmonicity respectively.

Aims and Objectives

Our motivation was to explore and discover new exciting never-before-heard sounds made from inharmonic sources that could be musicalised. This notion was exciting to us as it could open up several possibilities of experimentation and for interesting musical purposes. The primary aim of this project is to explore how the sound signature of an object can be made tonal without major changes to its character, through the use of software processing methods such as spectral mapping.

The main objective of this project is to produce a software system that, for any given input signal, performs frequency analysis and manipulation to form a tonal parallel.

2. Key Literature

Discussion of Key Literature

Harmonicity

Woodhouse [2] states that not any excitated object will sound as pleasant as an musical instrument (e.g. a saucepan or table), despite sharing similar properties as a tuned percussion instrument, such as frequency or amplitude response. This is because the frequency response of everyday objects often include natural or resonant tones, which are not harmonically related.

Harmonic entropy

FROM SETHARES' Tuning-Timbre-Spectrum-Scale (appendix J, p. 371)
"Harmonic entropy is a measure of the uncertainty in pitch perception, and it provides a physical correlate of tonalness, one aspect of the psychoacoustic concept of dissonance."
"It provides a way to measure the uncertainty of the fit of a harmonic template to a complex sound spectrum. As a major component of tonalness is the closeness of the partials of a complex sound to a harmonic series, high tonalness corresponds to low entropy and low tonalness corresponds to high entropy."

Spectral Mapping -

Discussion on Sethares' TransFormSynth and how ours is different

Technical discussion

- Single sine component vs multiple components
- No warping vs asymmetric warping
- Alternative tunings for real instruments vs harmonisation for object's latent harmonicity
- Complex user controls vs dynamic/simple user control
- Changes the fundamental according to user preference vs we find the fundamental and call it as it is
- Max/Java vs MATLAB/VST

Logistical discussion

- Why: what does it offer over these alternatives
- How did we build upon them?
- Can it be used in the same manner?

Maybe we could just make a table?

	Harmoniser	TransFormSynth

Sine Component	Single Component	Multiple Components

Other Similar Projects

Mogees [], developed by Mogees Ltd. is a way of transforming everyday objects of any material such as metal chairs and wooden tables into a musical instrument. It operates via a dedicated app on the iOS or Windows platform to detect specific pre-made gestures e.g. tapping, scratching in real-time using custom vibration sensors (high-quality contact microphones) which are attached to the desired instrument's surface.

Summary of Research

New instruments created in the digital age often follows a system of an input, mapping, and synthesis [10]. Similarly, the spectral mapping transformation we are interested in makes use of audio analysis and resynthesis, they are recurring fundamental techniques used across existing research to approach the development of new musical instruments.

Mogees proves how inharmonic objects are very much unexplored with new sounds yet to be heard but also how easily accessible they are and around us at all times. We strived to achieve a similar goal in our own project, to encourage the user to experiment and produce different sounds using the software we created. Although, in our case the inharmonic sound is harmonised.

3. Design and Development

System Requirements

Spectral Mapping

While Sethares' *TransFormSynth* performs spectral mapping upon single sinusoidal components only [Spectral Tools for Dynamic Tonality and Audio Morphing - CITE PROPERLY THIS], our system additionally maps all the sinusoidal components belonging to the the peaks' regions of the smoothed amplitude frequency spectrum of each frame (only if that peak comprehends a harmonic template's frequency). This process is achieved via independent linear interpolation of the left and right side of the amplitude peak-regions; each peak-frequency of the source spectrum and the harmonic template are paired with a nearest-neighbor algorithm, similar to the one suggested by Sethares [Spectral Tools for Dynamic Tonality and Audio Morphing - CITE PROPERLY THIS]. This process guarantees a more natural output sound, which presents more fidelity to the input audio file.

Enhancing

Since, by their own nature, inharmonic sounds tend to contain high percentages of floor noise, a amplitude enhancing algorithm has been implemented in order to scale up the mapped peaks with lower amplitude while maintaining fixed the loudest peak in the spectrum, which is taken as ceiling amplitude reference for the enhancing process. Thus, all the mapped peaks can be enhanced up to infinity, but not beyond the amplitude level of the loudest peak. This operation is the best compromise between keeping the original characteristic nature of the input sound (by maintaining constant, at least until the amplitude of the loudest peak has been reached, the ratios between the peaks' amplitude levels in the spectrum) and giving the user the possibility to enhance the harmonicity of the output sound.

- Amplification
- Morphing
- Output Sound

System Architecture (Final algorithm, high-level, low-level - big block diagram)

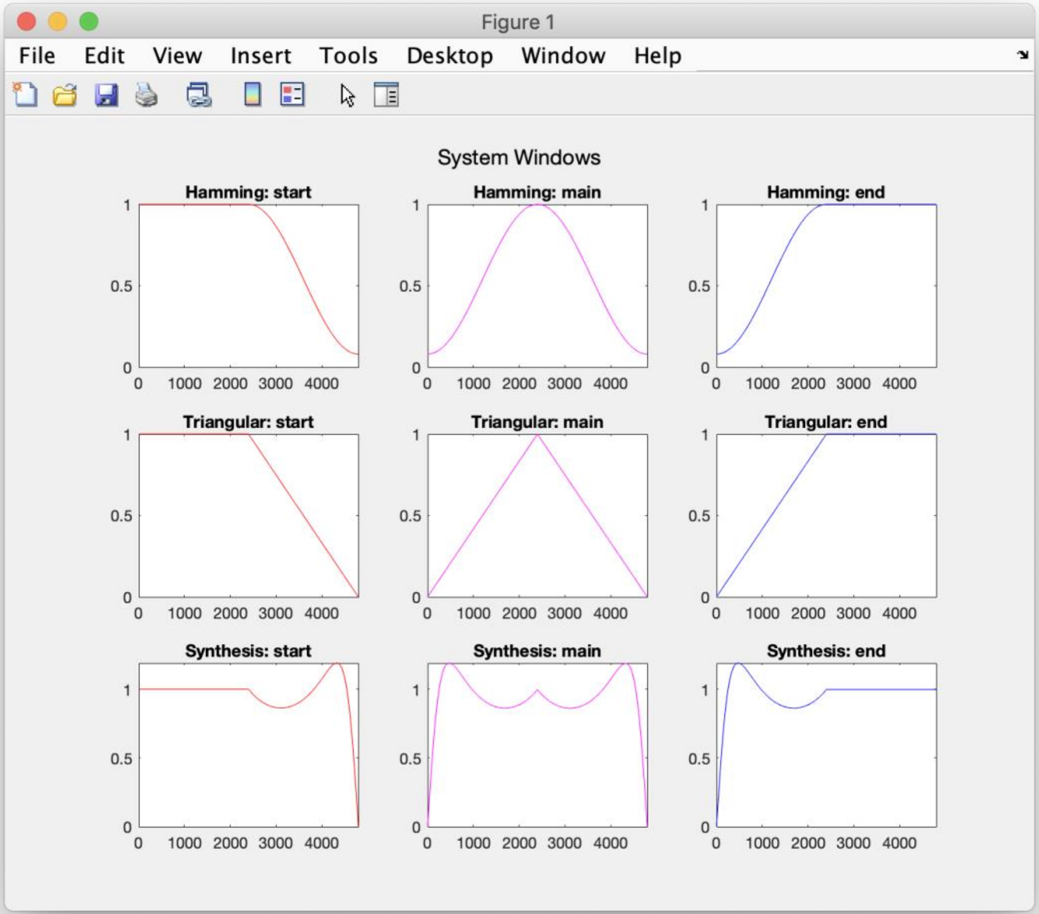
High-Level number 1

High-Level number 2

Deconstruction

In addition to the audio samples, the sampling frequency of the input audio file is required in order to properly calculate the information about each frame (length, in samples; overlap with previous and next frame, in samples). Furthermore, framing is also an inevitable process in conversions from time domain to frequency domain representations, since a single -and, thus, static- FFT representation of an entire audio file does not factor in temporal or spectral change.

Windowing is a consequence of the need of overlapping frames, which in turn are needed in order to avoid a “tremolo” effect in the output audio file. For the overlapping regions to have the same weight as the non-overlapping parts, a hamming window (similar to a bell-shape) is applied. Furthermore, the hamming window makes the frame’s samples more periodic by attenuating both ends of each frame, and this additionally improves the performance of the FFT, which supposes a periodic signal [Sethares’ Tuning, Timbre, Spectrum, Scale, Appendix C, p. 335]. For the FFT calculation process, it is worth mentioning that, due to the discrete nature of it, the frequency bin resolution (how many Hz are all the frequency bins spaced apart, which affects the spectral resolution of the spectral mapping process) depends both on the sampling frequency of the input audio file and on the sample length in samples (entered by the user). For the same sample frequency, a longer frame means a better FFT frequency resolution.



FREQUENCY BIN RESOLUTION FORMULA

$$frequency\ bin\ resolution = \frac{sample\ rate}{FFT\ size}$$

Spectral Analysis

Amplitude Smoothing

Due to the noisy nature of inharmonic signals and to the high number of data points, before calculating the number of peaks of each frame's frequency spectrum, the signal needs to be smoothed in order to get rid of high-frequency fluctuations (noise). Without the smoothing (which is very similar to a low-pass filter), in fact, the number of peaks/minimas obtained would be too high.

Spectrum is smoothed and peak subsequently counted until target number of peaks is met or exceeded - herein lies the fundamental flow in our program.

PICTURES

Furthermore, eliminating the noise makes possible trends, patterns and peaks much more evident. The smoothed frequency spectra are in fact used only to determine the position -as frequency bin number- of peaks, fundamental frequency (that is, the peak with lowest frequency) and minimas in such complex data sets; the actual amplitude values of those are taken from the un-smoothed spectra.

Peak Detection

The system classifies a peak region as the data between any two adjacent minima indices. Neighboring peak regions shared a minima point, such that they successively span the spectrum from the first to last identified minima indices.

Fundamental Frequency Detection

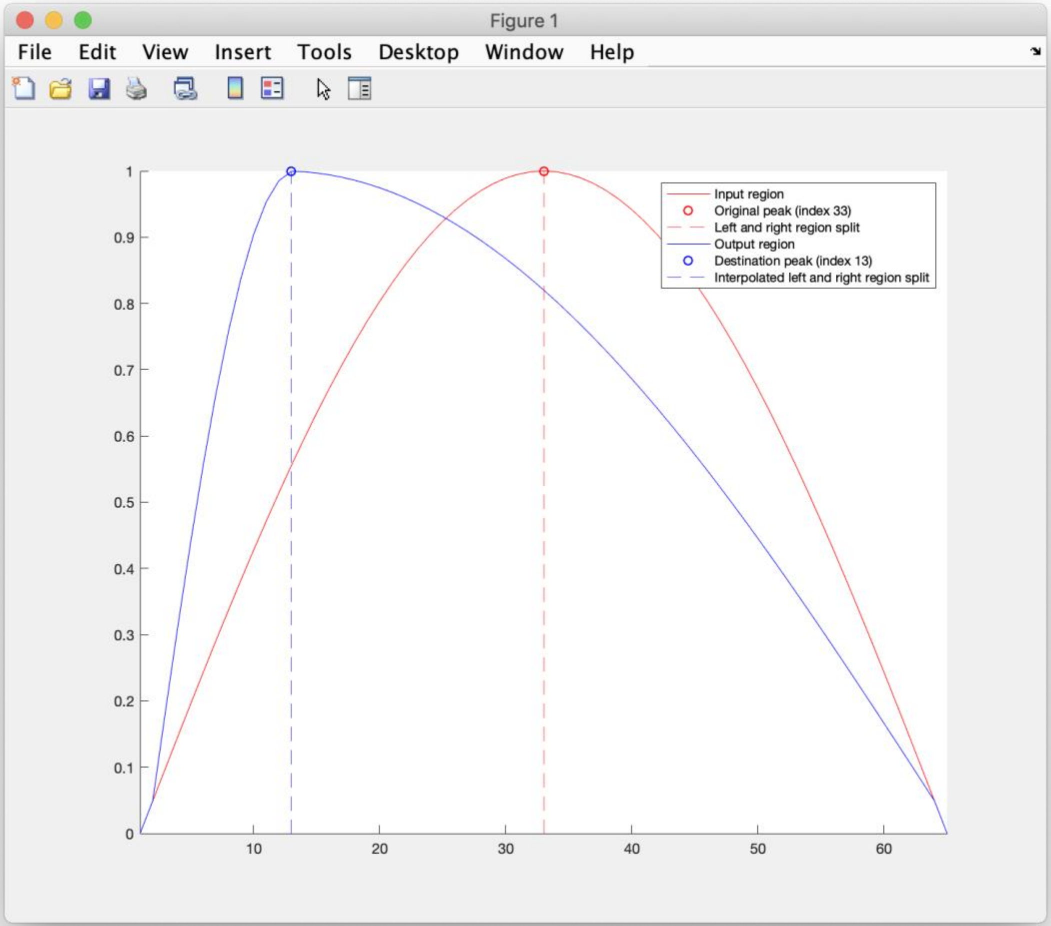
The system uses the first detected peak region's point of maxima as the fundamental frequency, from which the harmonic template is calculated. Importantly, this value is a bin index, and subsequent processes - including harmonic template calculation - operate without converting to frequency equivalents unnecessarily.

Spectral Mapping

Spectral mapping is the core process of the system, responsible for any kind of harmonic transformation. An original warping algorithm is used to morph each detected and valid peak region, such that an existing maxima index - original peak - is quantised to its corresponding harmonic of the fundamental - destination peak.

A peak region is considered valid for spectral mapping if the original and destination peak both fall within of the region bounds by at least three indices. These constraints ensure the minimum requirements for the interpolation processing are met, and the bordering indices are left unprocessed for seamless substitution of the peak region back into the spectrum.

Initially, the unprocessed peak region is split into left and right parts at the largest value location. Interpolation is then performed to either side consecutively, in order to transform each data set to new horizontal dimensions. The degree of interpolation for the left side is determined by the factor difference between the original and destination peak. The degree of interpolation for the right side is automatically calculated with respect to the left, so that appending both data sets results in an output the same size as the input. Upon insertion of the processed peak region back into its original position, this last step is crucial in preventing unwanted data discontinuities and constraint breaches in the context of the entire spectrum.



Reconstruction

Once the spectral mapping process, whose lifecycle belongs to the frequency domain representation of the audio signal, has been performed, each frame needs to be converted back to the time domain. In order to do this, the IFFT computation operates on both mapped amplitude and phase spectra of each frame.
Prior to recombining the audio waveform of each frame, a re-Synthesis window is applied on each of them. The re-Synthesis window consists in a 2 steps process; the first is a triangular window (whose shape is calculated depending on the frame overlapping percentage; the triangular window weights the overlapping portions of each frame less than the non-overlapping portions, so that linear equal amplitude is guaranteed in the reconstructed final audio output waveform [CITE PROPERLY THIS; http://kom.aau.dk/group/04gr742/pdf/framing_worksheet.pdf]), the second is an inverse

Hamming window (division by Hamming window; since in the Deconstruction stage we performed a multiplication by Hamming window, we now need to perform the inverse of it). Once we obtain an audio waveform of each frame, those need to be recombined and recombined with the other frames.

MATLAB Audio Toolbox (VST Development)

UI (Diagram), Explanation of layout Etc.

Summary of Developmental Processes

4. Testing and Validation

Testing Process

User input validation function (02_05 build)

What Tests were performed? How do they validate the requirements from chapter 3? Link to files.

Could import some graphs and then discuss them. Could also talk about some sounds that were outputted.
Quantitative and qualitative validation of system (How much does it meet the initial criteria?)
Any validation errors

5. Further

6. Conclusion

Overall Summary and Conclusion

Conclusion on final state of the system? Is it stable?

Challenges

Any relevant paths explored along the way? We explored several avenues.

A big challenge was finding a balance between keeping the original sample's characteristics and processing speed.

Sequential processing to avoid memory issues

Another was figuring out which way was best to maintain these characteristics. By morphing, enhancing etc.

Other particular coding problems also occurred such as clipping

Evaluation and Critical Analysis

Maybe somehow we could have made the code even more efficient? There could be a case where a very big file is imported and thus it takes much longer to process.

Phase warping - was this correct in performing interpolation like on amplitude components? Is phase scaleable in the same way as freq?

SMOOTHING PART (unless fixed before next thurs)

Should probably comment on the output sound. How it sounds etc. even if it doesnt sound "good", does it harmonise by our presented defention of harmonicity (yes it does, i promise) - if yes, we couldnt have

Possible Future Directions

A real-time implementation with faster processing speed.

Perhaps turn into a physical product instead of a software one with a microphone attached where one could record a sound and then import it for modification. This would be a portable product which could be taken outside where it would encourage spontaneous recordings with very creative outcomes.

Develop the system further in such a way to allow the user decide which wave characteristic the output sound should have such as square, sawtooth and triangle.

Xentonality

Division of Workload

Team Member	Sections
Matteo	
Reuben	
Zai	
Rhun	

Appendix

Final Algorithm (optional)

Class Diagrams (UML)
Spec of functions Etc.

References

(make sure these are in order of appearance and formatted correctly [IEEE])

[1] B. Zamborlin, B. Caramiaux, and C. C. Emanuele, “Method to Recognise a Gesture and Corresponding Device,” 2017.

[2] J. Woodhouse, “What makes an object into a musical instrument?” *Plus*, 03 Feb., 2011.

[3] Ableton Live, “24. Live Instrument Reference”, *24.2 Collision*. [Online]. Available: https://www.ableton.com/en/manual/live-instrument-reference/. [Accessed: 01-May-2019].

[4]

[5]

[6]

[7]

[8]

[9]

[10]

[11]

[12]

[13]

[14]

[15]