Music Analysis and Synthesis (ECS731P) - Assignment 3

Kranthi Yanamandra - 170797647

Overview:

The aim of the assignment is to loop a selected region in an audio (single note) by specifying the start and end points (samples) of the loop based on the results of calculating relevant attributes of the audio. To achieve this, the approach taken was as follows:

1. Relevant features were extracted from the audio.
2. Stable sections were identified from these features.
3. Loop boundaries were defined.
4. Audio to be output was synthesized based on the best segment found.

The following sections illustrate the implementation and evaluation of the process mentioned above.

Extracting features:

Fundamental frequency, spectral flux and Root Mean Square (RMS) were the three features chosen to be the measurements to determine the overall stability of the audio signal at a given time. These were calculated as follows.

An implementation of the YIN algorithm which was explored as part of the second coursework assignment, was used to identify the fundamental frequency of the audio. Parabolic interpolation was included in this for increased accuracy. As the analysis would involve specific and relatively static pitches (single musical note), the fundamental frequency was wrapped and centred, which would result in less octave errors. The estimated frequencies were then mapped to the semitone pitch spectrum by using the following equation[1]:

```
f(mask) = 12*log2(f(mask)/440.0)+69.0;
```

where *mask* is a Boolean vector containing the estimated frequencies that are greater than 0 and *f* is the estimated pitch. Pitches were wrapped around one octave and centred on the spectrum by subtracting the median and adding half an octave. All of this yielded in an improved measurement of pitch.

Spectral flux, which is used to measure the salience of spectral content, was calculated by adapting the approach presented by Alexander Lerch[2] in his book ‘An Introduction to Audio Content Analysis’. This was calculated as follows:

```
Spectralflux(n) = sqrt(sum((mag-mag1).^2))/(wsize/2);
```

where *n* is the current sample, *mag* is the vector containing absolute values of the FFT of the grain to be analysed, *mag1* is the vector to store previous window’s magnitude during analysis, and *wsize* is the window size.

Root Mean Square (RMS) was chosen as a simple measure to calculate the loudness of the signal. Lerch[2] presented an approach to calculate the RMS value of a signal as:

```
RMS(n) = rms(x(start:stop).^2);
```

where *n* is the current sample, *start* and *stop* are the indices in the current frame of audio, and *x* is the frame of audio. The same was adapted in the code to calculate the RMS value of the samples of the signal. A silence threshold of 25 dB below the maximum RMS value was set, to mask values lower than that. This was chosen as it seemed to yield the best results out of all the values experimented with.

Loop points and synthesis:

With the relevant features extracted, the next step was to perform analysis to know about the stability of the audio signal. To do this, a moving standard deviation was calculated for all the three features, hence creating 3 vectors to know how these were spread over time. An adaptive threshold that increases with each iteration was applied to these standard deviations, so that segments in consecutive frames could be searched to know if looping could be applied.

A frame was considered to be a potential candidate for looping if its length was greater than its mean fundamental period multiplied by the number of periods (set as a parameter). Out of all these frames, the one with the lowest accumulated standard deviation was chosen to be the best match, since it would mean the signal has changed the least here/ is relatively steady. To select the start and end points for the loop, positive zero crossings are searched for around the frame’s starting index and the closest multiple to the mean fundamental period at the end point[6]. This was done to account for any improper estimations of that segment's pitch.

The output audio was synthesized by dividing it into start, loop and end sections. An integer number of loops was specified in the place where a loop segment was found. This was calculated by using the duration provided in samples. The number of loops closest to the desired total length of the sample was used.

The following section describes the evaluation of the system.

Evaluation:

The code was tested on four different instrument samples – guitar, piano and saxophone and violin. Since the code assumes the audio to have a single note, it is unlikely to perform well for polyphonic recordings or very low frequency drum sounds. This is because the minimum frequency that a segment was allowed to have would be the main parameter used for tuning the system. Smaller values meant that the resulting loop would also be smaller with the content not varying much over time. If the value was too small, the loop sounded a lot less natural and more synthetic without much variation. On the contrary, high values resulted in loop that were longer and sounded much more natural with more variation. However, this meant that there would be a higher probability of discontinuities that arise due to sudden changes at loop points.

Longer samples would change/ evolve as the audio progresses, hence not able to loop very well due to lack of continuity at the start and end of the segment. Values between 14 to 30 periods were found to provide a good trade-off between these two characteristics. The audio files were tested with stretch ratios of 1.5 and 2.0 times of the original signal. Lesser values were not tested since the code was designed to handle no less than the original duration of the sample. Also, anything more than a ratio of 1.5 would only add more loops.

The design worked best for the saxophone piece with few artefacts. This was expected because the variation of harmonic content and energy over time would be the least for such instruments. Relatively percussive instruments such as the piano and the guitar performed less well but were nevertheless synthesized to some extent properly. The unnatural sustain of the notes was clearly noticeable. The violin sample performed the least well, with the vibrato not being preserved which made the loop sound completely synthetic.

Further work:

The following factors could be improved to build a more robust system to synthesize the audio:

1. Effects of amplitude variation and vibrato could be taken into account while synthesizing the output, so as to make it sound more natural.
2. The system could be made to work with polyphonic music and very low frequency percussive instruments.
3. Segment selection could be varied by enabling use of parameters to adjust moving standard deviation window size.
4. Currently, the output audio might not be the exact duration as specified. This is because the closest integer multiple of loops are used.

References:

[1] Dr. Rachel Hall. Pitch and Frequency. [online]
Available at
http://people.sju.edu/~rhall/SoundingNumber/pitch_and_frequency.pdf
[Accessed 26 March 2018].

[2] Lerch, A. (2012). An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics.

[3] Wikipedia, Cent (Music). [online]
Available at https://en.wikipedia.org/wiki/Cent_(music)

[4] Zolzer, U. (2002). DAFX – Digital Audio Effects. [online].
Available at http://www.music.mcgill.ca/~ich/classes/dafx_book.pdf

[5] freesound.org. (2012). Violin G-5 Tenuto Vibrato. Sax Alto - G5. Plucked Guitar Notes > A4.wav. [online].
Available at https://freesound.org/people/Carlos_Vaquero/sounds/153598/
https://freesound.org/people/MTG/sounds/358440/
https://freesound.org/people/Skamos66/sounds/399466/

[6] Zero Cross Detection. [online]
Available at
https://groups.google.com/forum/#!searchin/comp.soft-sys.matlab/messageid$3Aeebd047.-1@WebX.raydaftYaTP/comp.soft-sys.matlab/QyPmPQcUc1g/4GVrKJctqQgJ