Modification of duration and pitch of audio

Mathieu Lagrange

February 4, 2020

This document, the Matlab scripts and the datasets are available here:
`https://github.com/mathieulagrange/datasim/tps/lagrangeTpTsm`

The example code provided is given in Matlab, but the practical can be done in Python too. If using python, you may consider using the librosa library (`pip --user install librosa`) to manipulate audio files and `IPython.display.Audio` that lets you play audio directly in an Jupyter notebook.

1 Introduction

Several signal processing techniques can be used to modify an audio signal in a meaningful way. We focus here on the time and pitch modification by a factor of two in a first section.

2 Resampling

Take one excerpt from the sqam dataset : `https://tech.ebu.ch/publications/sqamcd`. You can load a shorter version, like twenty seconds of its data stream to debug. A simple sound, like `frer07_1.wav` can assist you to detect and fix more easily glitches. Please consider only one channel if the file you considered is stereo.

Extend and reduce its duration by a factor of two by resampling the data vector.

What happens to the pitch of the sound ?

3 Time blocking

We want to be able to modify the duration without affecting the pitch. For this, implement the time blocking method that splits the signal into overlapping frames (50 percents overlap). In order to limit discontinuities at boundaries of frames, window the frame using energy preserving windows (triangular, hanning, ...). Implement two functions. The first function called **blocking**, convert an audio signal into a matrix containing the windowed frames. The second function called **inverseBlocking**, convert the matrix into an audio signal.

Once done, you can extend the sound duration by a factor of two by duplicating the frames, and reduce the duration by removing half of the frames. You can consider the function **buffer**.

Experiment with frames of $512, 1024, 2048, 4096$ samples.

What is the frame size that gives the best listening quality of the modified sounds ?

Propose an algorithm for changing the pitch without affecting duration.

4 Short term Fourier Transform (STFT)

As you may have noticed, some artifacts occurs. In order to potentially mitigate them, we will consider the STFT approach. This approach describes each frame in the spectral domain, allowing us to manipulate more finely the signal and preserve some important continuity properties.

To this aim, consider the **specgram** (matlab built-in) and the **ispecgram** (provided) in order to implement a processing chain that transform the input signal into its complex spectrogram and revert the transformation. Use the frame size you selected in the previous experiment.

Display the magnitude spectrogram.

Verify that the resulting signal is the same than the original signal (up to some frame padding at the end).

Set the phase information to zero and revert to the time domain. What is the perceptual result ? Use a speech sample for this and a short frame size (512).

Use the Fourier analysis / synthesis processing pipeline in order to extend / reduce the duration of the signal. Is the quality any better ? Why ?

5 Griffin and Lim

Considering the redundancy of the information of the magnitude spectrogram due to the overlap between frames, the Griffin and Lim algorithm allows us to reconstruct a phase information that is more coherent with the one of the magnitude spectrogram.

The complex values of the spectrogram S of an audio signal s can be decomposed into polar coordinates, the magnitude spectrogram $|S|$ and the phase spectrogram $\angle S$ under the following relation:

$$S(j,k) = |S(j,k)|e^{i\angle S(j,k)} \tag{1}$$

Assuming an unknown phase set to random values $\angle S_0 = rand$, the algorithm proceeds as:

1. compute the time domain signal s_{m-1} as the ISTFT of S_{m-1}
2. compute the frequency domain representation $S' = \text{STFT}(s_{m-1})$ while enforcing the magnitude spectrogram : $S_m = |S|e^{i\angle S'_m}$

Repeat until convergence.

Experiment with the number of iterations $50, 200, 500, 1000$ considering a test sample where the duration is not changed but the phase is randomized.

Compute the Root Mean Square Error (RMSE) between the original signal and the resulting signal.

Does the quality improves with respect to the number of iterations ?

Does the perceptual quality improve also ?

6 Phase Vocoder

Another way to mitigate the phase issues is to enforce phase continuity for each frequency band of the STFT. To do so, the phase vocoder approach starts with an initial phase for the first frame (can be the one of the original first frame), and propagate the phase to the next frame assuming phase stationarity for each frequency band:

$$\angle S(j,k) = \angle S(j,k-1) + 2pif^j(t_k - t_{k-1}) \tag{2}$$

where j and k are respectively the frequency and time indexes. The values f^j and t_k are respectively the frequency and time in Hertz and seconds.

First consider the implementation of the phase vocoder provided (pvoc.m) to modify the sound. Use the phase vocoder to initialize the phase and use the griffin and lim algorithm to further improve the phase estimate.

Compare the quality of the 3 approaches:

1. griffin and lim only
2. phase vocoder only
3. both

both in terms of RMSE and perceptual quality.

Bonus: implement the phase vocoder.

7 Report

Please write a report using your favorite word processing tool and output a pdf file. The report shall have for each question a brief description about the way things have been done and some discussion about the resulting behavior.

Send an archive containing the report, the code no later than an hour after the end of the session to `mathieu.lagrange@cnrs.fr`, with the `[ECN]` flag within the title of the message.