A10-3: A multi-resolution sinusoidal model

Audio Signal Processing for Music Applications

Introduction

You have seen through several assignments that the choice of window size is an important tradeoff between time and frequency resolution. Longer windows have a better frequency resolution and can resolve two close sinusoids even at low frequencies, while smaller windows have a better time resolution leading to sharper onsets. So far, in all the analyses, we have only considered a single window length over the whole sound. As we know, analysis of signals with low frequency components needs a longer window as compared to signals with high frequency content. The optimal choice of window length is thus dependent on the frequency content of the signal. In other words, it is better to choose a longer window for analysis at low frequencies while a shorter window is sufficient at higher frequency. In this assignment, you will explore the use of multiple window sizes for analysis in different frequency bands of the signals.

Gudelines

You will implement a multi-resolution sine model compared to the one in `sineModel.sineModel()` function in sms-tools (sine model version without tracking). For each audio frame (at each 'pin') you will compute three different DFTs with three different window sizes (which are input parameters). You then compute the sinusoid peaks for each of the DFTs. Given three frequency bands spanning the whole range of 0-22050 Hz, you will then choose the peaks from these three DFTs depending on the band to which the frequency of the peak belongs, low frequency peaks from the DFT computed on a larger window.

e.g. Given three frequency bands `B1: 0 <= f < 1000Hz, B2: 1000 <= f < 5000, B3: 5000 <= f < 22050`, and three window sizes `M1 = 4095, M2 = 2047, M3 = 1023`, we generate three windows w1, w2, w3 with sizes M1, M2, M3 respectively. For every frame of audio x1 = x[pin-hM1:pin+hM2], we compute three DFTs `X1 = dftAnal(x1, w1, N1)`, `X2 = dftAnal(x2, w2, N2)` and `X3 = dftAnal(x3, w3, N3)`. Choose N1, N2 and N3 as needed. We then compute the peaks of the magnitude spectrum in each case and interpolate them. For reconstruction, we choose the peaks computed from X1, X2, X3 for reconstruction in frequency bands B1, B2 and B3, respectively. With peaks thus chosen, we reconstruct the frame. This ensures that we have a longer analysis window at low frequencies and a shorter analysis window at high frequencies.

Modify `sineModel.sineModel()` in `sineModel.py` (make a copy before changing the code) so that it can take three windows with different sizes, three FFT sizes, and band edges as input parameters, e.g., `sineModel.sineModelMultiRes(x, fs, [w1, w2, w3], [N1, N2, N3], t, [B1, B2, B3])`. Implement the multi-resolution analysis as described above.

Choose two different real world polyphonic recordings from freesound.org that have both a relevant melodic and percussion components. An example could be orchestra.wav in the sounds folder of sms-tools. Visualize each sound and choose a suitable set of parameters for its analysis. Experiment with different window lengths and band edges for the two sounds such that you get both crisp onsets and good frequency resolution.

You will be mainly evaluated for your efforts in the code you write and the contents of the report. You will evaluate a minimum of three other peers in this assignment.

Part 1

Write a short report (1 page) with the `freesound.org` links to the sound used, mentioning the band edges and the window sizes you used for each sound. Write a short explanation to justify your choices.

In the report, comment upon the advantage (or the lack of it) of using such a multi-resolution analysis in terms of obtaining better time-frequency resolution, computational complexity and extending the technique to HPR and HPS models. Specifically, if you were to improve the multi-resolution analysis and extend it to HPR and HPS model, comment on the challenges you would face for sinusoid tracking and F0 estimation. Additionally, you can suggest further methods to improving the time-frequency resolution trade-off (optional).

Make sure your report contains the following:

- Freesound link to the two sounds chosen.
- Explanation and justification of the band edges and the window sizes for each sound.
- Observations about the advantages of a multi-resolution analysis (comment on the time-frequency resolution, computational complexity and extensions to HPR and HPS models).
- Challenges you might face if you were to extend it to HPR and HPS models (mainly in sinusoid tracking and F0 estimation).
- (Optional) Further methods to improving the time-frequency resolution trade-off.

Upload the report as a PDF with this question.

Part 2

Upload the output sound of the multi-resolution sine model. You can name the files `A10-b-1.wav` and `A10-b-2.wav`. You should compress them into one zip file which you upload.

Part 3

Upload the code for the `sineModelMultiRes()` function that you implemented (if you forked sms-tools and modified it, a github link would suffice).