520.445/645 - Audio signal processing - Project1

Meili Liu

October 23, 2022

1 Introduction

There are two part in this project: signal reconstruction from STFT magnitude spectra and Time-Scale Modification (TSM). In the first part, we implement 3 algorithm: GL, RTISI, and RTISI-LA, followed by comparison of their performance based on SER function mentioned in the paper [3]. In the second part, we employ the RTISI-LA method on Time-Scale Modification, and compare its performance with classic WSOLA method on the given sample data.

2 Description of GL,RTISI, and RTISI-LA

Reconstruct a time-domain signal from a short-term Fourier transform magnitude (STFTM) spectrum is a challenging task, because the magnitude spectrum contains no phase information. In the past 20 years, several methods has been proposed to tackle this problem.

GL algorithm, proposed by Griffin and Lim [2] in 1984, is an iterative algorithm to estimate a signal from its STFTM. During the forward and inverse Fourier transform iteration, the mean squared error between given and estimated signal STFTM is guaranteed to decrease.

RTISI algorithm [1] is an variation of the classic GL algorithm by using previous partial frame to generate an initial estimate for the phases of current frame. This can provide a good start point for the iteration, thus reducing the number of iterations dramatically. Also important is that the algorithm can performs in real-time, since the current frame only rely on past estimated frame signal.

RTISI-LA algorithm [3] is an improvement of RTISI algorithm by employing the look-ahead strategy. Specifically, when reconstructing frame m, we also consider the k future frames because they also carry the phase of current frame through overlapping.

3 Evaluation of GL,RIISI, and RTISI-LA

In the this section, we will compare the SER result of these algorithms on given test samples. our test samples consist of 12 audio segments, we classify them into two categories, speech signal (monophonic) and music signal (polyphonic), shown on Table 1. We then compute the average SER scores for various algorithm using various settings on these two types of data.

3.1 Evaluation of Different Amount of Look-Ahead in RTISI-LA

First, we compare the SER scores on RTISI-LA algorithm with various number of look-ahead frames. The step size is fixed at $S = L/4$ where L is 1024, and the number of iterations per steps is set to 10. The SER results on various look-ahead frames are shown in Table 2.

From Table 2, we can see that the performance increase slightly with the number of look-ahead frames. However the computational time also increase with the number of look-ahead frames. For

audio	category
audio1	music
audio2	speech
audio3	speech
audio4	speech
audio5	speech
audio6	speech
audio7	speech
audio8	speech
audio9	music
audio10	music
audio11	music
audio12	music

Table 1: details of test samples.

look-ahead frames	Speech SER(dB)	Music SER(dB)
3	13.04	12.83
5	13.29	13.08
7	13.36	13.09
9	13.41	13.12

Table 2: RTISI-LA with different look-ahead numbers, The number of iterations is 10.

example, for k=3, the total iterations per step is (3+1)*2*10 = 80, while for k=9, the total iterations increase to (9+1)*2*10 = 200. Therefore, since the improvement from increasing k is small, we fix the look-ahead frames at 3 in the following experiments in order to reduce computation time.

3.2 Evaluation of Different Number of Iterations in RTISI-LA

We then compare the effect of number of iterations per step with fixed look-ahead frame at 3. As the number of iterations per step increases, the performance of RTISI-LA should also increase, since the phase information become more and more accurate during iterations . Table 3 shows this trend.

3.3 Evaluation of Different amount of Window Lengths in RTISI-LA

In addition, we evaluated the effect of window length on RTISI-LA. By changing the length of the analysis window when keeping the step size at 256, we actually change the amount of overlap between adjacent frames. The result is shown on table 4. The performance of using smaller window size is worse than that of using bigger window size. The possible explanation is that when we have larger window size, S/L ratio is smaller,which means we have more known part and less unknown part in a frame. This can provide a better initial phase for the algorithm and help it to converge better.

3.4 Comparing GL, RTISI and RTISI-LA

In this section, we compare the SER performance of GL, RTISI and RTISI-LA. The window length is fixed at 1024 with step size at 256. The number of iteration is set to 10 and look-ahead frame is set to 3.The SER result is shown on table 5.

From Table 5, we can see that the original RTISI algorithm is better than GL when the number of

look-ahead frames	number of iteration per step	total iterations	Speech SER(dB)	Music SER(dB)
3	2	8	11.65	11.53
3	10	40	13.04	12.83
3	20	80	13.21	12.92

Table 3: SER Evaluation of RTISI-LA with different iteration number. The look ahead frame is 3

window length	Speech SER(dB)	Music SER(dB)
512	11.51	11.02
768	12.78	12.43
1024	13.04	12.83
1256	13.21	12.92
1536	13.45	12.98

Table 4: SER Evaluation of RTISI with different window length. The look ahead frame is 3, and the number of iterations is 10

Algorithm	number of iterations	Speech SER(dB)	Music SER(dB)
G&L	1	6.15	6.03
G&L	5	10.34	9.34
G&L	10	12.21	11.50
G&L	20	14.32	13.49
G&L	20	15.58	14.74
RTISI	1	13.80	13.22
RTISI	5	14.66	13.54
RTISI	10	12.71	10.95
RTISI	20	11.49	10.06
RTISI	30	9.80	10.95
RTISI-LA	1	10.68	9.27
RTISI-LA	5	12.68	12.12
RTISI-LA	10	13.04	12.83
RTISI-LA	20	13.21	13.04
RTISI-LA	20	14.12	13.74

Table 5: SER Evaluation of GL, RTISI and RTISI-LA

iterations is less than or equal to 10. However, as the number of iterations increases, the performance of GL continues to increase, while the SER of RTISI goes down. We are not surprised that RTISI performs better than GL on a smaller number of iterations because previous partial frame can provide it with a good initial state, so RTISI can converge faster. However, the drop in SER is quite strange when the number of iterations keeps increasing. A possible explanation is that RTISI may learn wrong phase information during the iterations after convergence, because it only repeated estimation on itself. This can leads to the performance degradation.

By contrast, the RTISI-LA generally continues to improve as the number of iterations is increased, because we apply the transform iteration on both current frame and future k frames. However, its performance starts to lag behind GL when the number of iterations exceeds 20. One explanation for this is that GL can leverage the global information from whole frames, while RTISI-LA only have access to local information, namely previous k frame and future k frame.

In summary, when number of iteration is less than 20, the RTISI-LA performs better than GL and RTISI. After 20 iterations, although RTISI-LA is slightly worse than GL, it still keeps the advantage of running in real-time applications.

4 RTISI-LA on Time-Scale Modification

Time-scale Modification (TSM) is a subject for stretching or compressing the duration of audio signal without changing other features such as pitch and timbre. In the second part od this report, we first employ ATISI-LA algorithm on this task, then compare it with classic WSOLA algorithm.

Waveform Similarity Overlap-Add (WSOLA) is a variant of Overlap-Add (OLA) by introducing tolerance in the position of analysis or synthesis frame position. So we can select a synthesis frame whose waveforms are aligned with previous frame in the overlapping region. The similarity between two

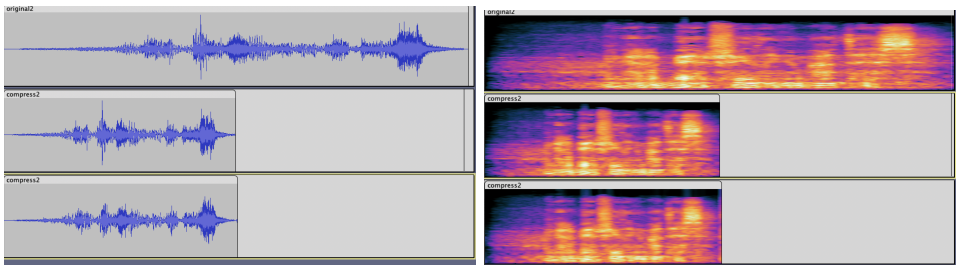

Figure 1: Comparison of synthesised compressed signal with original signal on time-domain and frequency. The first picture is from original signal. The second one is from WSOLA algorithm, and the third one is from RTISI-LA

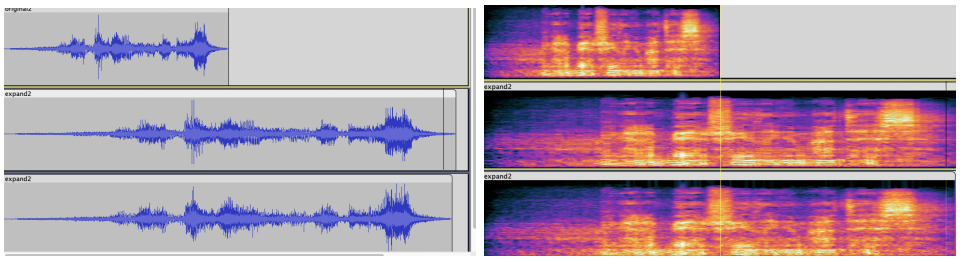

Figure 2: Comparison of synthesised stretched signal with original signal on time-domain.The first picture is from original signal. The second one is from WSOLA algorithm, and the third one is from RTISI-LA

frames can be computed by cross-correlation function. This strategy can help to reduce jump artifacts between frames in OLA.

In contrast with WSOLA, which is a time-domain method, RTISI-LA try to tackle TSM on frequency-domain. We first use an analysis step size S_a to obtain the short-time Fourier transform magnitude spectra. Then try to reconstruct signal using a synthesis step size $S_S = S_a * \alpha$, where α is the modification rate.

4.1 Comparison of RTISI-LA and WSOLA on TSM

In this section, we apply RTISI-LA and WSOLA on TSM task and compare their performance based on checking spectrum. We chose 0.5 and 2 as the modification rate. When $\alpha = 0.5, S_S = 2 * S_a$, the synthesis step size is double of analysis step size, the result is time-stretching. While for $\alpha = 0.5$, the result is time-compressing by a factor of 2.

For RTISI-LA, we set the look-ahead frame at 3 and number of iterations at 10 to achieve a reasonable balance between SER score and computational load.

In subjective listening tests, the perceptual quality of reconstructions using RTISI-LA and WSOLA is similar. We also compare their waveform and spectrum on audio2, the result is shown on figure 1 and figure 2. As we can see from these figure, although the duration is changed, the pitch and timbre are well preserved on both algorithm.

References

[1] Gerald T Beauregard, Xinglei Zhu, and Lonce Wyse. "An efficient algorithm for real-time spectrogram inversion". In: *Proceedings of the 8th international conference on digital audio effects*. 2005, pp. 116–118.

[2] Daniel Griffin and Jae Lim. "Signal estimation from modified short-time Fourier transform". In: *IEEE Transactions on acoustics, speech, and signal processing* 32.2 (1984), pp. 236–243.

[3] Xinglei Zhu, Gerald T Beauregard, and Lonce L Wyse. "Real-time signal estimation from modified short-time Fourier transform magnitude spectra". In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.5 (2007), pp. 1645–1653.