REASSIGNED SPECTROGRAM AND ITS TIME-DOMAIN CONVERSION

BY

TIANYI HE

THESIS

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Computer Engineering in the Undergraduate College of the University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

P. Smaragdis
Audio information accounts for a large portion of existing digital data, and numerous researchers are constantly developing new methods and tools for audio analysis. Short-time Fourier transformed based analysis is one of the most prominent analysis tools because of its linearity; however, this mechanism cannot resolve issues with inaccurate localization of energy in both time and frequency, especially for audio editing types of applications. Thus, the reassigned spectrogram method, developed by Auger and Flandrin in 1995, has won its place due to higher accuracy in energy localization by turning discretely sampled signal into continuous domain, and by reassigning the energy to the center of mass for each analysis bin. This method, nevertheless, is non-linear, and it is very difficult to synthesize analysis data back to time-domain representation. This thesis introduces mechanisms to solve this problem and summarizes the result of testing.

Keywords: Signal Processing, Audio Processing, Audio Analysis, Short-time Fourier Transform, Reassigned Spectrogram, Audio Synthesis
TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION ........................................ 1
  1.1 Background ................................................. 1
  1.2 Motivation .................................................. 2
  1.3 Contents .................................................... 3

CHAPTER 2 REASSIGNMENT ....................................... 4
  2.1 Time to Reassignment ...................................... 4
    2.1.1 Method ................................................. 4
    2.1.2 Result .................................................. 7
  2.2 Reassignment to Time ...................................... 8
    2.2.1 DFT Method ............................................. 8
    2.2.2 Additive Method ...................................... 9
    2.2.3 Interpolation Method .................................. 10
    2.2.4 Result .................................................. 12

CHAPTER 3 APPLICATION ........................................ 16
  3.1 Data Compression ......................................... 16
  3.2 Source Separation ........................................ 18
  3.3 Other Applications ....................................... 20

CHAPTER 4 CONCLUSION .......................................... 21
  4.1 Future Work ............................................... 21
  4.2 What I Learned ............................................ 22

CHAPTER 5 REFERENCES .......................................... 23
CHAPTER 1
INTRODUCTION

1.1 Background

As audio signal becomes more abundant, researchers are continuously developing new tools and techniques to more efficiently and accurately analyze and visualize signals. Short-time Fourier transform (STFT) evolved in the 1960s after the development of the phase vocoder, a filter-bank communication scheme to visualize energy in different frequency bands. STFT, \( S(t, \omega) \), analysis allows us to visualize signals in a more intuitive way, as music scores are structured, that is, with the magnitude of the frequency component plotted over time. As a result, we can visualize the strength of the frequency component as a function of time. The STFT based method is one of the most prominent methods, mainly due to its linear nature, which allows users and researchers to analyze, modify, edit, and synthesize the signal back into the time domain.

This method first divides signal \( x[n] \) into \( m \) chunks called “frames”, represented as \( \bar{x}[n,m] \), where \( m \) is the frame number and \( n \) is the sample index at each frame. If we define “hop size” to be \( h \), and “DFT size” to be \( d \), then we have equation (1.1) for each frame \( m \).

\[
\bar{x}[n,m] = x[m* h + n] \quad (1.1)
\]

We usually also multiply each frame by a choice of window function with length \( d \), \( w[n] \), so we define the windowed buffered signal as \( \bar{w}[n,m] \) in equation (1.2).

\[
\bar{w}[n,m] = \bar{x}[n,m] * w[n] \quad (1.2)
\]

Next is to transfer the vectorized signal into Fourier space by doing a
Fourier transform for each frame as in equation (1.3).

$$F^\mathbb{w}(\omega, m) = \sum_n \bar{w}[n, m] e^{-j\omega t} = F^x_w(\omega, m) \quad (1.3)$$

While the above representation was shown in continuous space, the discrete space result can be represented in the same way as:

$$F^\mathbb{w}(\omega_j, m) = \sum_n \bar{w}[n, m] e^{-j\omega_j t} = F^x_w(\omega_j, m) \quad (1.4)$$

where $\omega_j = \frac{2\pi j}{d}$ for $j \in [0, d - 1]$

1.2 Motivation

This characteristic of linear transformation of STFT allows music producers, electronic musicians, and researchers to widely use this method. With the $n\log(n)$ complexity of fast Fourier transform (FFT) algorithm, the STFT based analyze-synthesize scheme can now even be utilized in real time to achieve practical applications such as source separation, pitch detection, pitch correction, and more.

However, this analysis method has certain drawbacks. Usual STFT analysis does not take advantage of the sparsity of the signal. More important to researchers, this method is physically limited, to a certain degree, to localize energy in both frequency and time. Largely depending on the analysis window, STFT localization accuracy is one of the main problems in applications like de-noising and source separation. For instance, for an audio signal sampled at 44 kHz, if taking $d = 1024$, then we would have each analysis bin to be as wide as

$$44 \text{ kHz} / 1024 \approx 40 \text{ Hz} \quad (1.5)$$

of uncertain region. Thus, with the windowing smearing effect introduced by the window function, the energy is not localized in one single bin. As a consequence, Auger and Flandrin [1] introduced the reassignment method in 1995 to remedy problems outlined above.
1.3 Contents

This thesis focuses on the method of reassignment and results. Chapter 2 covers analysis and synthesis methods and results. Chapter 3 mainly covers applications of the reassignment method. Chapter 4 offers conclusions and addresses possible future work, in addition to discussing what I learned throughout the process of this research.
CHAPTER 2
REASSIGNMENT

This chapter will first discuss transformation of signal from time-domain to the reassignment space, and then methods to inversely transform reassignment analysis back to time-domain signal.

2.1 Time to Reassignment

2.1.1 Method

The method of reassignment is to localize the center of mass of each bin in spectrogram with respect to its frequency and time [2, 3], rather than keeping the fixed analysis location. So each point at the normal STFT grid, \( S(\omega, t) \), is mapped to a point \( \hat{S}(\omega_{ins}, t_{ins}) \), with the complex weight of \( S(\omega, t) \). Points with the same instant time and instant frequency can be mapped to the same point additively.

This method is analogous to finding the center of mass \((\bar{x}, \bar{y})\) in a 2-dimensional space for weight function \( f(x, y) \), which can be found with:

\[
\bar{x} = \frac{\int x f(x, y) dx}{\int f(x, y) dx} \tag{2.1}
\]

\[
\bar{y} = \frac{\int y f(x, y) dy}{\int f(x, y) dy} \tag{2.2}
\]

Thus, for a spectrogram, instead of having \( f(x, y) \), \( x(t, m_j), S(\omega, m_j) \) is being used. For each frame \( t = m_j \), if we have window function \( w(t) \), and signal \( x(t) \), we would have FFT of each frame to be:

\[
\mathbb{F}^x_w(\omega, m_j) = \int x(t, m_j)w(t)e^{-j\omega t}dt \tag{2.3}
\]
If we have the window instead to be $tw(t)$, then the above calculation of $\mathcal{F}_w^x(\omega, m_j)$ would be changed to $\mathcal{F}_{tw}^x(\omega, m_j)$, as follows:

$$\mathcal{F}_{tw}^x(\omega, m_j) = \int t x(t, m_j) w(t)e^{-j\omega t} dt \quad (2.4)$$

To combine equations (2.3) and (2.4), we define instantaneous time shift as the centroid position in time axis as:

$$\bar{t}(\omega, m_j) = \frac{\int t x(t, m_j) w(t)e^{-j\omega t} dt}{\int x(t, m_j) w(t)e^{-j\omega t} dt} \quad (2.5)$$

Then we obtain the instantaneous time shift for a specific frame for each specific frequency, with the same idea of gravity centering. And the calculation only involves calculating another STFT with different window, $tw(t)$. Although the calculation done here is shown as a continuous-time result, a discrete time result would be very similar.

As a result of equation (2.5), we can represent time-shift for each point as a division of two STFT analysis results with different window functions $w[n]$ and $tw[n]$,

$$\delta \bar{t}(\omega_i, m_j) = \text{real}\left(\frac{\mathcal{F}_{tw}^x(\omega_i, m_j)}{\mathcal{F}_w^x(\omega_i, m_j)}\right) \quad (2.6)$$

and the instantaneous time is then $t_{ins} = m_j + \delta \bar{t}$.

A similar calculation is done in frequency domain for frequency centroid; we can alternatively represent multiplication in time domain with convolution in frequency domain. For $W(\omega)$ being FFT of window $w(t)$, and $X(\omega, m_j)$ being FFT of signal $x(t)$, then we can specify each FFT frame $m_j$ as:

$$\mathcal{F}_h^x(\omega, m_j) = \int W(v) X(\omega - v, m_j) dv \quad (2.7)$$

The Fourier transform property states that $\mathcal{F}(\frac{dw(t)}{dt}) = j\omega W(\omega)$; therefore, if using $\frac{dw(t)}{dt}$ as the window instead of $h(t)$, each frame can be represented as:

$$\mathcal{F}_{\frac{dw(t)}{dt}}^x(\omega, m_j) = \int jvW(v) X(\omega - v, m_j) dv \quad (2.8)$$

So if combining equation (2.7) and (2.8):

$$-j\bar{\omega}(\omega, m_j) = \frac{\int jvW(v) X(\omega - v, m_j) dv}{\int W(v) X(\omega - v, m_j) dv} \quad (2.9)$$
This result is almost the instant frequency shift $\omega_{\text{ins}}$, and can be calculated for each bin with only another calculation of STFT with a different window, $\frac{dh(t)}{dt}$.

$$\delta \bar{\omega}(\omega_i, m_j) = -\text{imag} \left( \frac{\mathcal{F}_{\omega}(\omega_i, m_j)}{\mathcal{F}_{\omega}(\omega_i, m_j)} \right)$$  \hspace{1cm} (2.10)

Knowing the frequency shift, then the instantaneous frequency is calculated as $\omega_{\text{ins}} = \omega_i + \delta \bar{\omega}$.

Having determined the corresponding instantaneous frequency and time for each analysis bin with the method above [4], the last thing is to correct the phase for each analysis point $\hat{S}(\omega_{\text{ins}}, t_{\text{ins}})$. Before assigning the same magnitude to $\hat{S}(\omega_{\text{ins}}, t_{\text{ins}})$ as the magnitude, we have to account for the phase modification that happens when both time and frequency change [5]. As phase can be represented in equation (2.11),

$$\phi = \int \omega(t) dt$$  \hspace{1cm} (2.11)

the change in phase from $t_0$ to $t_{\text{ins}}$ can be represented as:

$$\delta \phi = \int_{t_0}^{t_{\text{ins}}} \omega(t) dt$$  \hspace{1cm} (2.12)

where $t_0$ is the original time of the analysis bin, $t_{\text{ins}}$ is the reassigned instantaneous time, and $\omega(t)$ is frequency as a function of time. However, such operation is not feasible in discrete analysis; as a result, an approximation is performed with linear interpolation as:

$$\Delta \phi = \frac{\omega_0 + \omega_{\text{ins}}}{2} \cdot (t_{\text{ins}} - t_0)$$  \hspace{1cm} (2.13)

This phase would be added to the original phase from the original STFT, $\angle S(\omega, t)$, which completed the reassignment calculation.

Thus, for each analysis bin that exists in the STFT matrix, there is a corresponding point that each bin warped to with frequency $\bar{\omega}$, time $\bar{t}$, amplitude $A$, and phase $\phi$.

Having generated the reassigned warping, the analysis result is filtered based on perceptual criteria to force the spectrogram make sense. For example, if a point is warped to far away, it is reasonable to believe there is a calculation error and thus to ignore that point [2, 3].
The last step is to perform selection based on the energy distribution in the reassigned space, exploiting the sparse nature of the audio signals; most of the time, keeping 80% or 90% of the energy is more than necessary. This step largely reduces the amount of storage space required for the analysis result.

2.1.2 Result

The example is given by an audio file about 2.5 s long, sampled at 11025 Hz. Figure 2.1 shows the reassignment result of this audio file keeping 90% of energy. This reassigned spectrogram is plotted using a scatter plot fashion, with weight $|\hat{S}(\omega_{ins}, t_{ins})|$, indicated by the color. Knowing the piece is a short piece of a music from 1980s, readers can identify the low-frequency component as bass drum, the wide-band component as snare drum, and the harmonic component as a bell.

![Reassignment in Scatter Plot, $S_{bat}$](image)

Figure 2.1: Reassignment Result

The same piece of signal is also analyzed with the ordinary STFT method shown in Figure 2.2. As we can see, both of them capture almost the same amount of information, and the reassigned spectrogram shows the harmonic
component with higher accuracy and less smearing.

![Reassignment Result](image)

Figure 2.2: Reassignment Result

2.2 Reassignment to Time

Having discussed the results and the method of reassignment, we now understand that reassignment is not a linear transformation on the original signal; thus, it is rather difficult to convert the analysis result back to time-domain representation of a signal. The essential part of this thesis is to develop and test different methods to synthesize the analyzed result back to time. The following sections will focus on explanation and discussion of three methods that have been developed and tested.

2.2.1 DFT Method

The first method that I developed was to interpolate the time axis based on frame number by zeroth-order interpolation. This means merging points on the fixed frame by finding the closest frame. There are \( h \) samples between
the start of each frame, and with a zeroth order interpolation, all analysis points can be considered to happen at the closest frame. Although this interpolation is zeroth order on the time axis, it is still linear in the phase, so utilize \( \Delta \phi = \frac{\omega_{\text{ins}}}{2} \ast (t_{\text{ins}} - t_0) \) as the phase shift as a result from time shift,

\[
\hat{S}'(\omega_{\text{ins}}, m) = \hat{S}(\omega_{\text{ins}}, t_{\text{ins}}) \ast e^{j\omega_{\text{ins}}(m-t_{\text{ins}})}
\]  

where both \( m \) and \( t_{\text{ins}} \) are in the same unit of time.

There will be no interpolation on the frequency axis, so each analysis point corresponds to only one single frequency, \( \omega_{\text{ins}} \). The interpolated results \( \hat{S}'(\omega_{\text{ins}}, m) \) would preserve the continuous frequency but will lose the continuous representation of time by substituting time shift with phase shift.

For each frame, synthesis is performed with inverse discrete Fourier transform based on the analysis point in the frame, with the frequency selected at \( \omega_{\text{ins}} \) for each matching analysis point. As a result, the size of the inverse Fourier transform matrix depends on the number of analysis points available in the frame. If the number of points in a frame is \( p \), then the Fourier matrix will be \( d \) by \( 2p \) for the frame, where the factor of 2 accounts for the symmetrical part in the Fourier transform.

Having obtained the transform result for each frame, a normal inverse overlap-add procedure will allow us to synthesize the time-domain signal.

2.2.2 Additive Method

The DFT method mentioned above does a zeroth-order interpolation based on frame, and merges several points in one frame. It is natural to further ignore the limitation of frame, and directly overlap-add all sinusoidal components.

This method, additive synthesis, computes the corresponding sinusoid with given frequency \( \omega_{\text{ins}} \), amplitude \( A \), phase \( \phi \), and starting time \( t_{\text{ins}} \). Given these four parameters just mentioned, construct the \( i \)th component sinusoid

\[
c_i[n] = A_i \cos(\omega_{\text{ins}}n + \phi_i)
\]

and add it up with overlap-add method at the exact time instant to synthesize
\[ \hat{x}[n] = \hat{x}[t_{ins} + n] + c_i[n] \ast w[n] \]  
\[ (2.16) \]

where \( w[n] \) is the analysis window.

However, there is no guarantee that \( t_{ins} \) is integer-valued; therefore, zeroth-order interpolation is done to remedy non-integer shift, and less-than-a-half-sample time shift is accounted for by the phase shift.

\[ \delta \phi = \omega_{ins} \ast (\text{round}(t_{ins}) - t_{ins}) \]  
\[ (2.17) \]

Accounting for \( \delta \phi \) when constructing sinusoids yields equation (2.18).

\[ c_i[n] = A_i \cos(\omega_{ins}n + \phi_i + \delta \phi_i) \]  
\[ (2.18) \]

This method seems like the DFT method, but it actually works very well because it takes into account the different time-domain window and smoothes all sinusoids at different times reducing artifacts and enabling more natural synthesis. However, if the phase information is inaccurate, the method will break down from incorrect phase interference and give a terrible result.

### 2.2.3 Interpolation Method

The natural way to explain the smearing effect in ordinary STFT representation is that is the convolution of the signal of interest with the window function in the Fourier space. Thus, if reassignment is the way to relocalize the signal of interest, we can reproduce and synthesize the smearing effect, or re-interpolate the STFT matrix by convoluting the reassigned signal with window functions. The natural choice of such a kernel or window function is to choose exactly the same window we use for analysis, and to use its Fourier space representation to interpolate in frequency axis and time-domain representation to interpolate in time axis.

Thus, with a given analysis window function \( w \), first determine the analytical form of \( w[n] \). Next, designate a region of interest; that is, interpolation only around one point will be performed, and the size of this region will largely affect the computation. For a specific point \( \hat{S}(\omega_{ins}, t_{ins}) \), we would be
interested only in

\[ \{ \omega_m : \frac{L}{2} < \omega_m < \omega_{\text{ins}} + \frac{L}{2}, \omega_m \in \mathbb{Z} \} \quad (2.19) \]

In other words, interpolation is only performed with the integer frequency component in the region of interest, that defined with length \( L \) in frequency axis. The convoluted result, in frequency axis, can be shown as the following equation:

\[
\hat{S}'(\omega_m, \omega_{\text{ins}}, t_{\text{ins}}) = \hat{S}(\omega_{\text{ins}}, t_{\text{ins}})(\sum_n w[n]e^{j\omega_m n}) \quad (2.20)
\]

where the first part of the right-hand side, \( \hat{S}(\omega_{\text{ins}}, t_{\text{ins}}) \), is the signal of interest, or the magnitude in the reassigned space, represented as a delta function, and the second part, \( \sum_n w[n]e^{j\omega_m n} \), is the weight of the kernel at specific frequency \( \omega_m \).

The above mechanism defines interpolation in the frequency axis, but not in the time axis. The time domain convolution is defined with a different kernel function, the time-domain representation of the window \( w(t) \) in the region of interest. Therefore, the process is similar to that of frequency. Within the region defined as:

\[ \{ t_n : t_{\text{ins}} - M/2 < t_n < t_{\text{ins}} + M/2, t_n \in \mathbb{Z} \} \quad (2.21) \]

where \( t_n \) represents the integer frame number, the final result in the region of interest is:

\[
\hat{S}'(\omega_m, t_n, \omega_{\text{ins}}, t_{\text{ins}}) = \hat{S}(\omega_{\text{ins}}, t_{\text{ins}})(\sum_n w[n]e^{j\omega_m n})w(t_n - t_{\text{ins}})e^{j\omega_m \omega_{\text{ins}}/(t_n - t_{\text{ins}})} \quad (2.22)
\]

The first part \( w(t_n - t_{\text{ins}}) \) of the additional two parts on the right-hand side of equation (2.22), compared to equation (2.20), indicates the weighting in the time domain based on difference in frame number, and the second part \( e^{j\omega_m \omega_{\text{ins}}/(t_n - t_{\text{ins}})} \) is the phase correction term, which is a result of linear phase approximation as in equation (2.13).

Keep in mind that the above \( \hat{S}'(\omega_m, t_n, \omega_{\text{ins}}, t_{\text{ins}}) \) was the contribution from one point in the reassigned space \( \hat{S}(\omega_{\text{ins}}, t_{\text{ins}}) \), but the actual interpolated
STFT spectrogram \( \hat{S}'(\omega_{\text{ins}}, t_{\text{ins}}) \) is then the sum of all the regions of interest.

\[
\hat{S}'(\omega_m, t_n) = \sum \hat{S}'(\omega_m, t_n, \omega_{\text{ins}}, t_{\text{ins}})
\] (2.23)

The performance of this method largely depends on two parameters that define the size of the region of interest.

2.2.4 Result

It will be helpful to see the synthesis performance of different methods mentioned above, so the above three methods were performed with the same analysis result, a trumpet tone sampled at 22050 Hz.

The original STFT spectrogram of the audio piece is shown in Figure 2.3, and the reassignment result with only 85% of energy is shown in Figure 2.4.

![Figure 2.3: Original Result](image)

Figures 2.3 and 2.4 show that keeping only 85% of the energy throws out some of the high-frequency component. Next, the synthesized signal resulting from different methods will be presented as a combination of longitudinal waveform and STFT analysis results, and then numerical measurement of distortion, artifact, and interference will be shown.

Shown in Figure 2.5 is the result from the DFT method. The spectrogram seems to be acceptable, but the waveform is very different from the original time-domain representation, evincing a problem in phase.

The second result, Figure 2.6, is from additive synthesis. While observing the correct shape of the waveform, a modulation effect can be also identified in the time domain signal, resulting from the imperfect phase overlap.
Figure 2.4: Reassignment Result

Figure 2.5: DFT Method Result
However, this result is much better than that using DFT.

The last result from the interpolation method, shown in Figure 2.7, produces a better looking overall window, but strong modulation effect. The result is synthesized with 1% of DFT size in the frequency axis, and 1 DFT window size in the time axis.

Table 2.1 shows sound quality measurements for different methods. The ordinary STFT result is much better than the reassigned result; however, the measurements were not performed on completely synchronized data for reassignment, whereas the STFT result is completely synchronized.
Table 2.1: Synthesis Result Measurement

<table>
<thead>
<tr>
<th>Method</th>
<th>SDR</th>
<th>SIR</th>
<th>SAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>STFT</td>
<td>51.3</td>
<td>Inf</td>
<td>51.3</td>
</tr>
<tr>
<td>DFT</td>
<td>-16.5</td>
<td>Inf</td>
<td>-16.5</td>
</tr>
<tr>
<td>Addition</td>
<td>-4.3</td>
<td>Inf</td>
<td>-4.3</td>
</tr>
<tr>
<td>Interpolation</td>
<td>-2.4</td>
<td>Inf</td>
<td>-2.4</td>
</tr>
</tbody>
</table>
CHAPTER 3
APPLICATION

With higher fidelity in the location of energy both in frequency and time, it is possible to have better performance of de-noising by localizing noise better, and source separation by localizing the frequency component more accurately for each source. As the reassigned space takes advantage of sparsity, data compression is also a possible application.

3.1 Data Compression

One of the applications may be data compression. Since the reassigned spectrogram can take advantage of the sparsity of the analysis result, it is possible to analyze and perform compression, and then maintain a certain fidelity with the synthesized result.

Certain parameters in the process of analysis can be chosen to consider how much information or energy we wish to keep. In the following example, by keeping 90%, 85%, and 80% of the energy of the piano note, we saved different amounts of the storage space compared to the STFT analysis.

The example was performed on audio data of a piano note sampled at 44 kHz. Synthesis is performed with additive synthesis. With the choice of keeping 90%, 85%, or 80% of energy, the results are shown in Figures 3.1 to 3.3, and Table 3.1 shows the numerical measurements for synthesis with different amounts of energy.

<table>
<thead>
<tr>
<th>Method</th>
<th>Data Reduction</th>
<th>SDR</th>
<th>SIR</th>
<th>SAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reassignment 90%</td>
<td>97.0706%</td>
<td>-1</td>
<td>inf</td>
<td>-1</td>
</tr>
<tr>
<td>Reassignment 85%</td>
<td>97.7573%</td>
<td>-1</td>
<td>inf</td>
<td>-1</td>
</tr>
<tr>
<td>Reassignment 80%</td>
<td>98.2282%</td>
<td>-1</td>
<td>inf</td>
<td>-1</td>
</tr>
</tbody>
</table>
Figure 3.1: Reassignment Result with 90% of Energy

Figure 3.2: Reassignment Result with 85% of Energy
The table and figures above indicate that keeping 80% to 90% of energy would not make much of a difference in terms of sound quality of the synthesis signal and storage space for this signal.

3.2 Source Separation

One of the other applications enabled by the synthesis of reassigned spectrogram is source separation. Source separation is among the most popular topics in audio engineering, and many researchers have pursued it with methods consisting of different levels of complexity on STFT-based analysis tools. A method of supervised source separation applying masking is shown here as an example of the feasibility of this application.

The masking method involves two steps. The first step is to compare the magnitude of the original pieces; if audio A is the signal to be extracted, then comparisons of audio A with all other mixed pieces by the magnitude in the reassigned space are performed in order to generate the mask. To be precise, if audio A has greater magnitude in a certain region than other audio pieces, we will label this region as “True”. Then the second step is to apply the
mask to the mixed signal; the regions labeled as “True” in the mixed signal will stay, whereas the “False” region will be left empty.

The test is performed with two speech audio signals sampled at 22.05 kHz, and the mixed signal is composed of two speakers speaking different sentences. Figure 3.4 shows the mixed audio file in reassigned space, and the result is shown in Figure 3.5, with the two speakers’ signals separated:

![Reassignment in Scatter Plot, S\textsubscript{hat}](image)

Figure 3.4: Mixed Speech Spectrogram

Numerical measurements of the performance of the method are listed in Table 3.2, with comparison to ordinary STFT masking method. Low SDR and SAR values are results from imperfect synthesis methods, and acceptable SIR value shows the application is possible and feasible.

Table 3.2: Source Separation Result Measurement

<table>
<thead>
<tr>
<th>Method</th>
<th>Source</th>
<th>SDR</th>
<th>SIR</th>
<th>SAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reassignment Additive</td>
<td>Source 1</td>
<td>-0.3</td>
<td>18.4</td>
<td>-0.1</td>
</tr>
<tr>
<td>Reassignment Additive</td>
<td>Source 2</td>
<td>2.8</td>
<td>36.6</td>
<td>2.8</td>
</tr>
<tr>
<td>STFT</td>
<td>Source 1</td>
<td>14.2</td>
<td>34.3</td>
<td>14.3</td>
</tr>
<tr>
<td>STFT</td>
<td>Source 2</td>
<td>14.3</td>
<td>29.6</td>
<td>14.4</td>
</tr>
</tbody>
</table>
3.3 Other Applications

Other applications may include but are not limited to all types of learning algorithms based on normal STFT analysis-synthesis scheme. Musical applications such as de-noising applications, pitch correction, pitch detection, music transcription, and audio synthesis are all possible applications for re-assigned spectrogram.
CHAPTER 4

CONCLUSION

While STFT is a tool largely developed for audio analysis, the transformed version, reassigned spectrogram, has not been developed much. There are two main reasons for this lack of development: first, this transformation is still a linear projection, if the low energy data and the structure are kept; second, while the transformation offers certain benefits in the new space, such as localization of energy, it is not fully equipped with a synthesis method. The results reported in this thesis are acceptable in terms of synthesis, suggesting that the proposed methods can be further improved and more widely utilized, and may even ultimately perform better than STFT. This chapter will discuss the possible future work on these synthesis methods, and what I learned from this research experience.

4.1 Future Work

The most important thing about an audio signal is sound quality; as a result, the quality of the audio being heard is the first and major aspect to work on. The problem still exists in the fact that the phases of the sinusoids do not line up perfectly, which will deteriorate the sound quality. Given the opportunity to further improve my work, I would focus on synthesized sound quality of the audio signal, mainly by investigating the phase of the sinusoids.

Secondly, producing tracking and clustering between points can possibly help both synthesis and analysis. Tracking provides the phase shift as a continuous signal, which is homogeneous among all tracks, and clustering may decrease the chances of multiple sinusoids adding up with incoherent phases. Tracking with the McAaulay-Quatieri method is a prominent musical analysis tool, and I believe implementing the McAaulay-Quatieri method for tracking and some clustering technique in the reassigned space may result in
improve sound quality.

4.2 What I Learned

There are three main things I would like to discuss, the first thing being determination and commitment. This is by far the longest project I have ever worked on, starting out without any knowledge in the field, but ending up with acceptable deliverables. Throughout the process, one needs to always keep up with what one is working on, being determined to read literature, solve problems, and propose different solutions and test them; one must also be committed so as to actually work on the thesis on a daily basis. Determination and commitment are what I found most important throughout the process.

Secondly, I would like to discuss process and result. Most of the time we value results more than the process. Nevertheless, during the course of research it is not always the case that we meet our expectation. Being productive is one thing, but it is never guaranteed to have any result. Learning during the process of research is what research really means for me. A result is an instantaneous product, but process is where we put our determination and commitment.

Apart from determination and commitment, discipline and independence are the last things I would like to focus on. As evinced above the process is important, but how exactly we work is also crucial. Performing under supervision and having disciplines are important aspects for a research project, but they do not count for the entire research work. Most of the time we still have to work alone and perform individually. I believe this is also a part that I need to work on more. This project has forced me to work individually in an unfamiliar field and enabled me to improve my capacity for independent research.
CHAPTER 5

REFERENCES


