Priyanshi Rastogi
1/28/2020

Gabor Transforms

By: Priyanshi Rastogi

1. **Abstract:** In this report we are finding the time frequency signature of a portion of Handel's Messiah. We are able to analyze the time frequency signature by using filters such as the Gabor transform, Mexican Hat Wavelet and more. We explore how different widths of the windows affects the quality of the produced spectrogram as well as the different filter functions themselves to get us the best resolution for time and frequency. In the second part we look at filtering out the overtones in the given song (Mary had a little lamb, recorder and piano version) by again using the Gabor filter and analyze the time-frequency signature to determine the difference between a recorder and a piano and which notes are being played in each.

2. **Introduction/Overview:** We are given the time signal of Handel's Messiah and we want to be able to produce an ideal spectrogram of his work, one of which shows the best of the time resolution as well as frequency. In order to get the best of the two, we looked at oversampling (which includes having a small filter width) so that the data isn't lost when we translate the window over the time. By looking at spectrograms with different widths of translation and filter functions we were able to produce a spectrogram that gave us an ideal spectrogram in which we were able to get most of the time and frequency resolution. Through the use of Gabor filtering again, we were able to see how the song Mary had a little Lamb played through a piano vs. a recorder differed by filtering the overtones which occurred at multiples of the center frequency and helped in producing a clean music score for the piece. We were then able to compare it to the music scale to see exactly which notes were being played (by relating them to the corresponding frequencies).

3. **Theoretical Background:** From our previous knowledge, we were only able to tell either a time series analysis of a signal or the frequency analysis of a signal. In this case, we are given time signals of the different songs and analyze them using the ***Gabor Transform***. The Gabor filter is able to convert the given time signal and give us a time and frequency resolution. The Gabor transform, also known as the short-time Fourier transform (STFT) is then defined as the following: (Kutz)

$$\mathcal{G}[f](t,\omega) = \tilde{f}_g(t,\omega) = \int_{-\infty}^{\infty} f(\tau)\bar{g}(\tau - t)e^{-i\omega\tau}d\tau = (f, \bar{g}_{t,\omega}) \qquad \ldots ①$$

Here $f(\tau)$ is just the function of the signal. τ slides the time-filtering window down the entire signal in order to pick out the frequency information at each instant of time (Kutz). We also have $e^{-i\omega t}$ because it can be written as sines and cosines. These values also heavily depend on our given time signal and what information we want to capture based on that. From our prior knowledge we know that the Gaussian filter is able to give us a clearer signal of what we already have around some center frequency but here the Gabor transform is able to use the Gabor function to slide it over the time domain and focus on certain areas where it needs to pick up better time resolution and certain areas where it needs to pick up better frequency resolution in its filter.

The ***Mexican Hat Wavelet*** is another filter function that helps us grasp better information about our spectrogram. In particular, the Mexican Hat Wavelet has excellent localization

properties in both the time and frequency domain and can be dilated and translated easily, in contrast to the Gaussian filter which has minimal time-bandwidth (Kutz).

$\underline{Mexican\ Hat\ Wavelet = (1 - t^2)e^{-t^2/2}}$.... (2)

In our case, since we want to be using this filter to slide over increments of our time domain, we also want to have a scaling factor α that scales the filter at each increment. Here, the parameters of the wavelet are t, the time domain for which we want to slide the filter across, choosing either a specific time value or a range of time values that we can then slide the filter across accordingly.

4. Algorithm Implementation and Development:

For finding the "best" spectrogram of Messiah we were able to break it down by first using the Gabor Transform to create a general spectrogram and then were able to explore how different width and increments of our 'tslide' (by what increments we wanted to translate our filter function over the time domain).

1. One of our first steps is to load in the data (see Line 1) to analyze the piece *Messiah*.
2. Our next step is to make sure all of our dimensions are defined correctly, largely significant to make sure our spectrogram that is produced has accurate values. We load in our signal 'v' and define n to be the length of that signal and define our time domain to be going from 1 to n with L = 10 (see line 5). We want to make sure that we adjust our frequency (k and respectively ks) because there are an odd number of frequencies (thus making it go from 0…(n-1)/2 –(n-1)/2 -1 and remembering the time signal domain by $2\pi/2L$ to fit into the frequency domain and later taking the fftshift (to align the domain and frequency values). We can also see what our given signal looks like (lines 3-13).
3. We then make a vector of the different window widths that we want to try to make a spectrogram of and put those values in our Gabor transform function (again creating a different spectrogram for each window width) and defining our 'tslide' (the range of which we will slide the Gabor filter over the time domain with some increment). So we multiply our signal with the Gabor filter (at each width, α), take the Fourier transform of it to get it into a frequency signal and then put that into a vector (taking the fftshift and absolute value because we want to align the domains and can only focus on the positive parts) and plot that vector which produces our spectrogram. We chose the final α values and the increment of the time slide ('tslide') after exploring and experimenting with various values. (Lines 15-36).
4. Additionally, we can also compare using the Mexican Hat Wavelet (as described in section 3, equation 2) and make sure we scale it which is what the α value is doing within the function. We slide the Mexican Hat Wavelet across the time domain of the given signal by multiplying the signal by the filter and then taking the Fourier Transform of it (Lines 38-53) and then plot our spectrogram.
5. In our next part, we analyze Mary Had a Little Lamb and see which notes are being played by producing a spectrogram of the notes. We do this by first loading in our song (played through the piano) and set our dimensions/set up the same way as we did when analyzing Messiah. Here we want to make sure we convert our frequencies into Hertz so we divide our k (frequencies) by 2π and then fftshift them, again to align our domains (see lines 63-64). After experimenting, we set our 'tslide' (used in Gabor transform) by which we are translating the filter over the time domain by 0.1 and have the center frequency of our Gaussian filter that we use to filter out the

overtones at 300 and its width to be 0.00009 since we know the smaller the width's value the wider the filter becomes. We also make the width of our Gabor transform to be 100. All of these values were set after exploring different values for them and plotting the spectrogram to see which final values would make the spectrogram give us the best representation of which notes were being played (comparing them to the frequencies on the music scale) over time (see Lines 56-89).

6. For the second part, with the recorder version of Mary Had a Little Lamb we would want to do the exact same thing as we did with the piano version. Based on its signal and exploring various ranges of values, all of our values are set to be the exact same as for the piano version (see Lines 90-126).

5. Computational Results: (see Figures in bottom of section 6.)

After creating an ideal spectrogram of Handel's Messiah, we were able to see (from Figure 1) that using an a (width) value of 1 in the Gabor transform gave us a pretty decent time and frequency resolution. We are able to see how well the time resolution is defined within each spectrogram by how clearly we can see the vertical lines over the time domain and similarly, we are able to notice how well the frequency resolution by the horizontal lines that are displayed. With width of a = 1 we can see both pretty clearly. Similarly, (from Figure 2) we can see that the Mexican Hat Wavelet does a great job of combining those two resolutions and giving the best of both time and frequency resolution since we are able to see multiple vertical lines and several clear horizontal lines, thus giving us a spectrogram slightly better than of the one that used Gabor's transform.

We created a spectrogram of the piano version of Mary Had a Little Lamb using the Gabor Transform to translate the signal into a time and frequency spectrogram and then further filtered out the overtones using a Gaussian filter centered around 300 (which we were able to tell from the initial spectrogram). We can compare the frequencies of the notes being played to the musical scale given to us and see that the notes that are being played in order are EDCD EEE DDD EEE EDCD EEEE DD EDC each note for about half a second with the very last note being played for 2+ seconds (see Figure 3).

We did the same thing for the recorder version of Mary Had a Little Lamb and see that, after filtering around the center frequency (900), the notes that are being played are approximately BAGA BBB AAA BBBB AGA BBBB AABA GG each for again about half a second and the last note being played for around 2 seconds (see Figure 4).

6. Summary and Conclusions:

In conclusion, in part 1 we were able to explore how various filter widths, increments of our time slide domain, and even different filters themselves were able to help us produce the 'best' spectrogram of Handel's Messiah. We were able to see that having a width between 0.3 and 1 seemed to give us the best time and frequency resolution through the use of Gabor Transform (Figure 1). We then explored even further by using the Mexican Hat Wavelet function instead of the Gabor Transform function to see how that would help produce an even more ideal spectrogram. By sliding over the Mexican Hat Wavelet function over our time domain and scaling it each time by a width of .5 we were able to see that it the spectrogram that produced gave us much higher time resolution (as compared to the one made using Gabor Transform) as well as great frequency resolution (Figure 2). Therefore, we were able to conclude that using Mexican Hat Wavelet can be more useful in certain situations depending on your signal and by adjusting the parameters to fit best to that signal accordingly and similarly with the Gabor Transform.

Priyanshi Rastogi
1/28/2020

In part two, we analyzed the song Mary Had a Little Lamb that was being played on the piano and another version that was played using a recorder. For the piano version, we used the Gabor Transform to get the given signal to reflect both time and frequency. Then we applied a Gaussian filter after that to filter out any overtones and to get a clean score/neat spectrogram of the notes that were being played (EDCD EEE DDD EEE EDCD EEEE DD EDC), by comparing the frequencies on the spectrogram to the frequencies on the music scale (see Figure 3). We also observed that each note was being played for about .5 seconds and the last note for about 2 seconds. We were able to use the same process with the recorder version of the song and were able to see the notes BAGA BBB AAA BBBB AGA BBBB AABA GG (see Figure 4) being played for also about 0.5 seconds and last note for about 2 seconds.

Our spectrograms for the two signals revealed that for the song being played on the recorder, the highest frequency notes are the brightest (in color) and for the piano spectrogram (see Figure 3 & 4), the brightest notes ended up being the ones with the lowest frequency since the central frequency is stronger. This tells us that even if two instruments were played at the exact same frequency, the main difference would be that the piano would have stronger lower overtones and the recorder would have stronger high overtones.

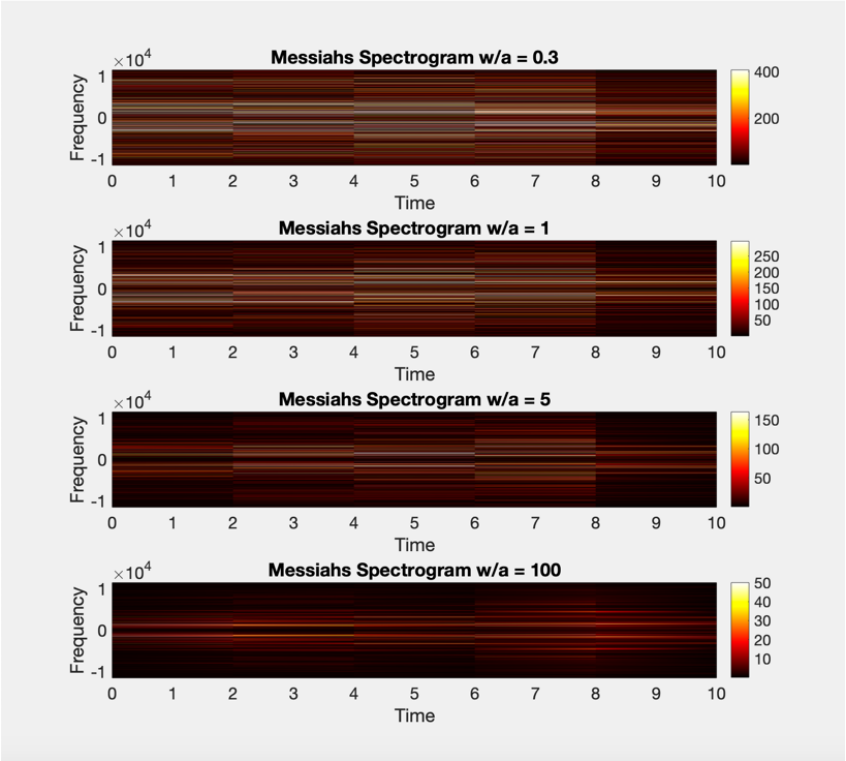

Figure 1: Shows spectrogram of Messiah filtered through use of Gabor Transform with various width a (of filter) values.

Priyanshi Rastogi
1/28/2020

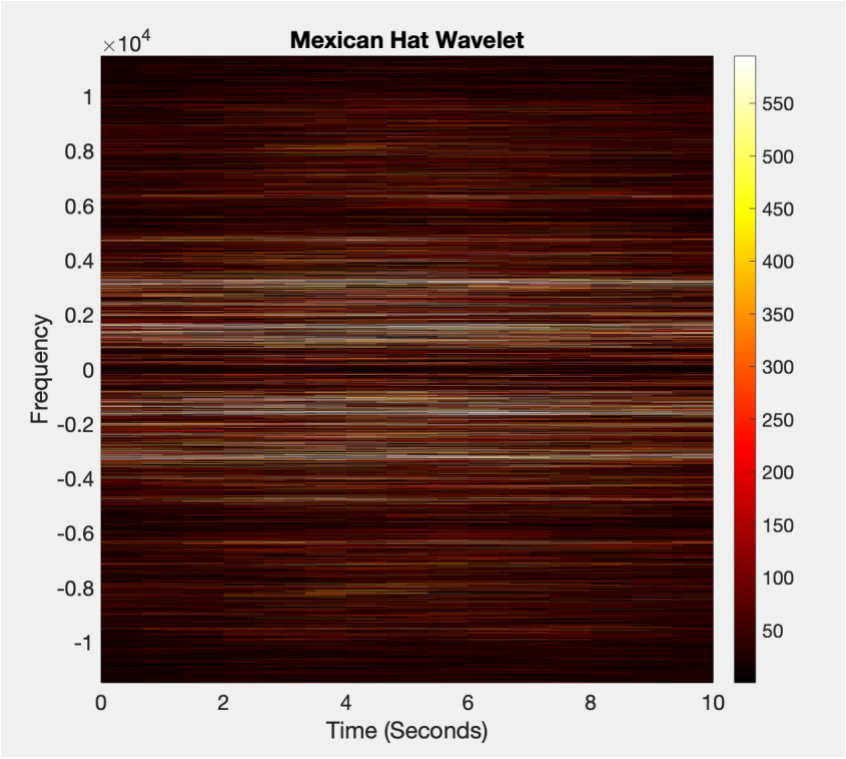

Figure 2: Shows filtered Messiah's spectrogram using the Mexican Hat Wavelet filter function.

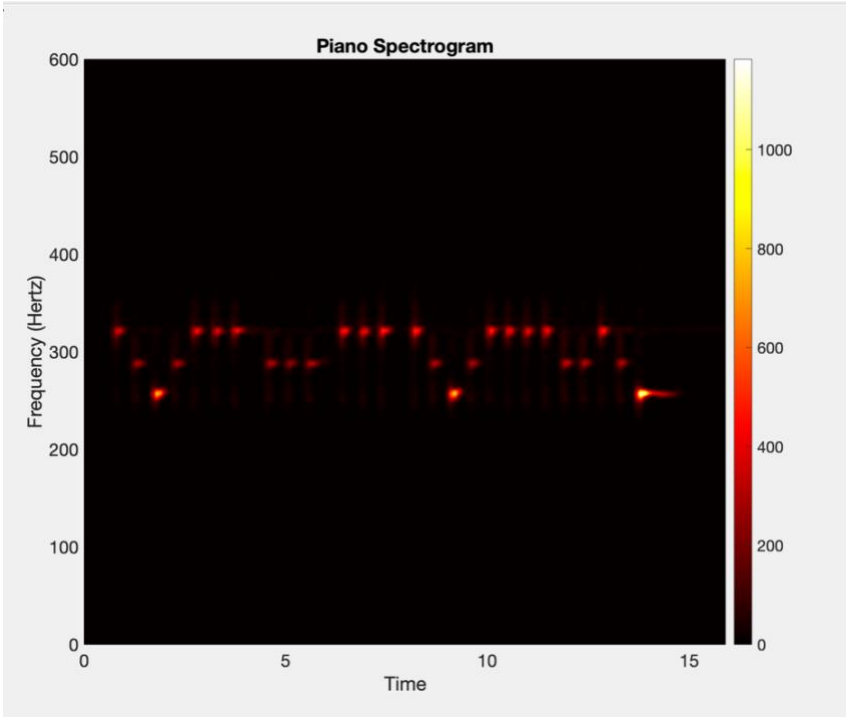

Figure 3: Shows how the spectrogram of the piano version of Mary Had a Little Lamb and the frequencies of the notes being played over time (seconds).

Priyanshi Rastogi
1/28/2020

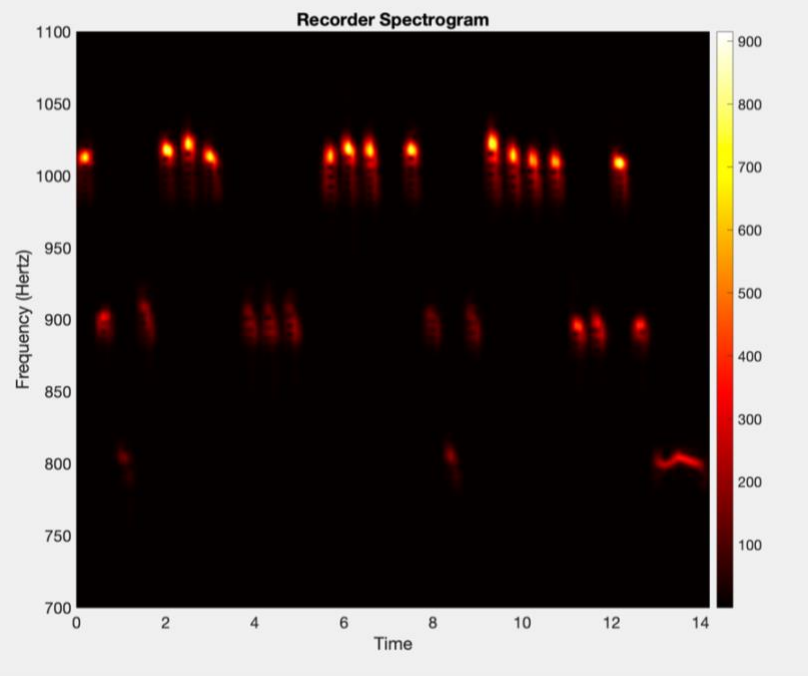

Figure 4: Spectrogram of the recorder version of Mary Had a Little Lamb using the Gabor transform and gaussian filter (to filter out overtones) containing the frequencies of the notes that were played over time (seconds).

7. Appendix A-Function Descriptions:

- fft: Fourier transform (in this case in 2D)
- fftshift: switching the first and second half of the domain so that it aligns when plotted
- abs: absolute value
- linspace: create an array of n points, from for ex. -L to L
- exp: e raised to any power passed in
- plot(x,y): plots passed in parameters x over y
- length(v): returns the length of the passed in parameter
- zeros(n,m): creates a matrix of size nxm with all zeros as its values.
- Subplot(rows, columns, index): puts the plot in the specifced index of the subplot (that is of dimension rows x columns)
- pcolor: creates a pseudocolor plot using the passed in parameters.
- colormap: map sets the colormap for the current figure to one of the predefined colormaps.
- shading interp: varies the color in each line segment and face by interpolating the colormap index or true color value across the line or face.

References: Kutz, Nathan J. (2013). *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data.*

8. Appendix B:

```
%% Part 1:
load handel
v = y';
L=10; n=length(v);
```

Priyanshi Rastogi
1/28/2020

```
t2 = linspace(0,L,n+1);
t = t2(1:n);
k=(2*pi/(2*L))*[0:((n-1)/2) (-(n-1))/2:-1];%for odd number of
frequencies
ks=fftshift(k);
%amplitude over time
      %plot((1:length(v))/Fs,v);
      %xlabel('Time [sec]');
      %ylabel('Amplitude');
      %title('Signal of Interest, v(n)');
      %% Look at different window sizes: a = 5, 2, 10
% Gabor Transform
      windowSizes = [0.3 1 5 100]; %so a = large so it is narrow-
>meaning good frequency, low time if small = wide less frequency and
more time
      tslide = 0:2:L; %experimented and found 2 to be the best value
between width and frequency
      Vgt_spec = zeros(length(tslide), n);
      for i = 1:length(windowSizes)
            a = windowSizes(i); %width changes
            Sgt_spec = zeros(length(tslide),n);
            for j = 1:length(tslide)
                  g = exp(-a.*((t-tslide(j)).^2)); %the gabor transform
                  Vg = g.*v;
                  Vgt = fft(Vg);
                  Vgt_spec(j,:) = fftshift(abs(Vgt));
            end
            subplot(4,1,i)
            pcolor(tslide,ks,Vgt_spec.'),
            shading interp
            title(['Messiahs Spectrogram w/a =
',num2str(a)],'Fontsize',16)
            set(gca,'Fontsize',16)
            colorbar
            colormap(hot)
            ylabel('Frequency')
            xlabel('Time')
      end
      %% Mexican Hat Wavelet:
      m_hatVt_spec = zeros(length(tslide), n);
      a = .5;
      for j = 1:length(tslide)
            m_hat = (1-a*(t-tslide(j)).^2).*exp(-(a*(t-
tslide(j))).^2/2);
            m_hatV = m_hat.*v;
            m_hatVt = fft(m_hatV);
            m_hatVt_spec(j,:) = fftshift(abs(m_hatVt));
            pcolor(tslide,ks,m_hatVt_spec.'),
            shading interp
            title('Mexican Hat Wavelet','Fontsize',16)
```

```
            xlabel('Time (Seconds)');
            ylabel('Frequency');
            set(gca,'Fontsize',16)
            colorbar
            colormap(hot)
      end
      %p8 = audioplayer(v,Fs);
      %playblocking(p8);
      %% Part 2 - Mary Had a Little Lamb (piano/recorder)
      [y,Fs] = audioread('music1.wav');
      tr_piano=length(y)/Fs;
      v = y.';
      L = tr_piano; n=length(v);
      t2 = linspace(0,L,n+1);
      t = t2(1:n);
      k=(1/L)*[0:n/2-1 -n/2:-1];%for odd number of frequencies/convert
   to hertz
      ks=fftshift(k);
      %record time in seconds1
      plot((1:length(y))/Fs,y);
      a = 100;
      tslide = 0:0.1:L;
      width = 0.00009;
      centerFreq = 300;
      for j = 1:length(tslide)
            g = exp(-a.*((t-tslide(j)).^2)); %the gabor transform
            filtered = g.*y.';
            filtered_fft = abs(fft(filtered));
            gaussian = exp(-width*((k-centerFreq).^2)); %using gaussian
            filter to filter out overtones
            filtered_fft = filtered_fft.*gaussian;
            piano_spec(j,:) = fftshift(abs(filtered_fft));
      end
      pcolor(tslide,ks,piano_spec.'),
      shading interp
      title('Piano Spectrogram','Fontsize',16)
      set(gca,'Fontsize',16)
      colorbar
      colormap(hot)
      xlabel('Time')
      ylabel('Frequency (Hertz)')
      ylim([0, 600])
      % xlabel('Time [sec]'); ylabel('Amplitude');
      % title('Mary had a little lamb (piano)');
      p8 = audioplayer(y,Fs); playblocking(p8);
      %% Recorder version of MHLL pt.2
      [y,Fs] = audioread('music2.wav');
      tr_rec=length(y)/Fs;  % record time in seconds
      v = y.';
      L = tr_rec; n=length(v);
```

Priyanshi Rastogi
1/28/2020

```
    t2 = linspace(0,L,n+1);
    t = t2(1:n);
    k=(1/L)*[0:n/2-1 -n/2:-1];%for odd number of frequencies/convert
to hertz
    ks=fftshift(k);
    a = 100;
    tslide = 0:0.1:L;
    vector_Cf = zeros();
    width = .00009;
    centerFreq = 900;%based on initial spectrogram
    recorder_spec = zeros(length(tslide),n);
    for j = 1:length(tslide)
        g = exp(-a.*((t-tslide(j)).^2)); %the gabor transform
        filtered = g.*y.';
        filtered_fft = abs(fft(filtered));
        gaussian = exp(-width*((k-centerFreq).^2));
        filtered_fft = filtered_fft.*gaussian;
        recorder_spec(j,:) = fftshift(abs(filtered_fft));
    end
    pcolor(tslide,ks,recorder_spec.'),
    shading interp
    title('Recorder Spectrogram','Fontsize',16)
    set(gca,'Fontsize',16)
    colorbar
    colormap(hot)
    xlabel('Time')
    ylabel('Frequency (Hertz)')
    ylim([700, 1100])
    %plot((1:length(y))/Fs,y);
    %xlabel('Time [sec]'); ylabel('Amplitude');
    %title('Mary had a little lamb (recorder)');
    %p8 = audioplayer(y,Fs); playblocking(p8);
```