ECE278C Imaging Systems

Lab 2: Back Propagation

February 3, 2021

Student: Ivan Arevalo
Perm Number: 5613567
Email: ifa@ucsb.com

Department of Electrical and Computer Engineering, UCSB

0 Motivation

The main objective of this report is to explore image formation by backward propagation for a simplified 2D case instead of the full-scale 3D plan-to-plane model. We will be building off our work in Lab 1 where we visualized and analyzed the spectral distribution of 2-D coherent wave-field patterns under different parameter variations such as wavelength, sample spacing, and aperture size.

1 Image Reconstruction by Inverse Filtering

Image reconstruction involves estimating the source distribution from the wave-field pattern measured at the aperture. The wave-filed generated by a point source in 2D space is mathematically described by Green's function:

$$h(x, y) = \frac{1}{\sqrt{j\lambda_0 r}} exp\frac{j2\pi r}{\lambda_0})$$
$$r = \sqrt{x^2 + y^2}$$

We can therefore express the wave-field generated by any source distribution as a convolution of the source function with the impulse response of the system (Green's function):

$$s(x, y) * h(x, y) = g(x, y)$$

In Frequency domain:

$$S(f_x, f_y)H(f_x, f_y) = G(f_x, f_y)$$

In the context of a 2D wave-field pattern, we can consider a special case of line to line propagation. For this case, the source distribution is at a line at y_1 and the data acquisition is conducted at a line at y_2. The distance in between is therefore $y_0 = y_2 - y_1$.

$$S(f_x; y_1)H(f_x, y_0) = G(f_x, y_2)$$

$$H(f_x, y_0) = exp(j2\pi y_0\sqrt{\frac{1}{\lambda} - f_x^2})$$

We can now formulate a method to recover the source distribution by reversing the wave scattering process, often referred to as inverse filtering (Lee 2016). Inverting our line to line propagation filter found previously:

$$\begin{aligned} H^{-1}(f_x; y_0) &= exp(-j2\pi y_0\sqrt{\frac{1}{\lambda} - f_x^2}) \\ &= H^*(f_x; y_0) \end{aligned}$$

Given the transfer function is all-phase, its inverse is the same as its conjugate. Transforming back into the the space domain we get the equivalent expressions:

$$H(f_x; y_0) \rightarrow h(x,y) = \frac{1}{j\lambda r} exp(j2\pi r/\lambda)$$
$$H^*(f_x; y_0) \rightarrow h^*(-x,-y) = \frac{-1}{j\lambda r} exp(-j2\pi r/\lambda)$$

We conclude that given the transfer function is the conjugate of the forward wave-field function (Green's function), then the impulse response of the image reconstruction filter is the flipped and conjugated version of the forward function function. Given that the Green's function is even, flipping it is irrelevant and we just get the conjugated forward Green's function.

We can now state the forward and backward propagation equations as:

$$\text{Forward Propagation: } s(x,y) * h(x,y) = g(x,y)$$
$$\text{Backward Propagation: } g(x,y) * h(x,y)^* = \hat{s}(x,y)$$

where $s(x,y)$ is our source distribution, $h(x,y)$ is Green's function, $g(x,y)$ is our wave-field data, and $\hat{s}(s,y)$ is our reconstructed image.

We can now reconstruct the image by convolving the wave-field data collected at the aperture with the conjugate of Green's function.

2 Simulation

We begin our simulation by generating the wave field for a the following 6-point source distribution:

	scatters	*scatter locations*
1	(x_1, y_1)	$(0, +10\lambda)$
2	(x_2, y_2)	$(+10\lambda, 0)$
3	(x_3, y_3)	$(0, -10\lambda)$
4	(x_4, y_4)	$(-10\lambda, 0)$
5	(x_5, y_5)	$(-8\lambda, -6\lambda)$
6	(x_6, y_6)	$(+8\lambda, -6\lambda)$

Figure 1: Source Distribution

The receiver aperture is a centered linear receiver array with a span of $60\lambda_0$ (from x = $-30\lambda_0$ to x = $+30\lambda_0$). The linear receiver array is located at $y = yo = -60\lambda_0$. With quarter-wavelength spacing ($\lambda_0/4$) spacing, there are 241 wave-field data samples in total captured over the $60\lambda_0$-long linear aperture. Figure 2 shows the source wave-field pattern along with the receiver aperture highlighted in red.

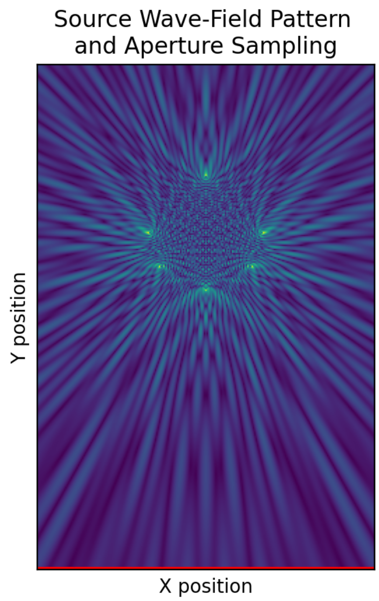

Figure 2: Source Wave-Field with Aperture Receiver shown in red

We can now reconstruct the $60\lambda_0 \times 60\lambda_0$ source region by convolving our receiver data with the conjugate Green's function. This is analogous to treating each of the 258 data samples at the receiver, as a point source and observing the resulting superimposed wave-field.

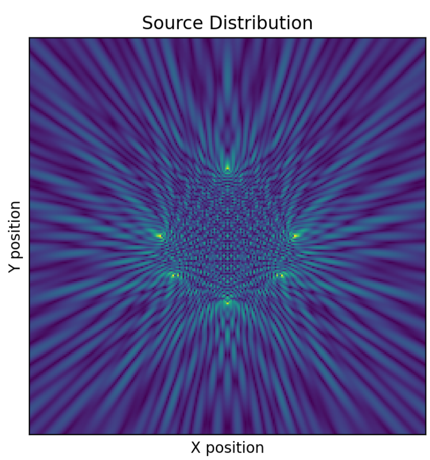

Figure 3: Source Distribution

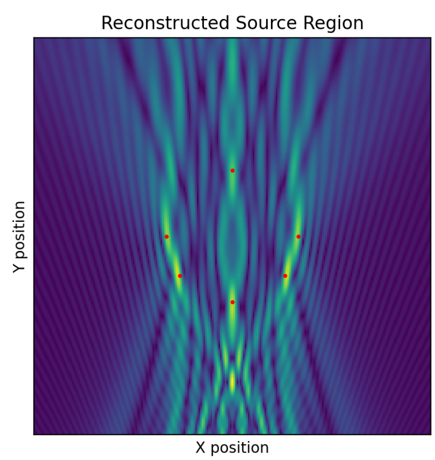

Figure 4: Reconstructed Source Region; Source Points Highlighted in Red

Figure 3 shows the original source distribution and generated wave-field pattern. Observing the reconstructed source region in figure 4, we notice the true point source locations highlighted in red, are well estimated in the x-axis but display significant distortion in the y-axis. As will be shown, this results from the receiver aperture only capturing wave-field data in the direction facing the aperture.

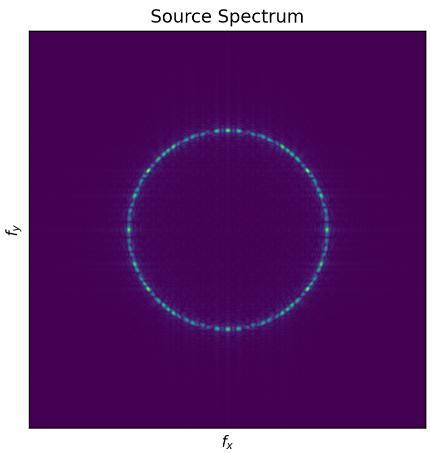

Figure 5: Source Spectrum

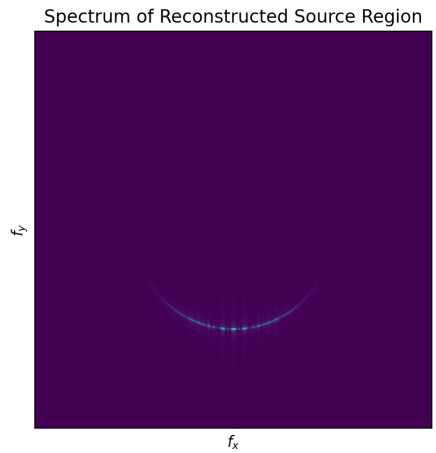

Figure 6: Spectrum of Reconstructed Source Region

Figure 5 and 6 show the source and reconstructed image spectrum respectively. We can observe that the reconstructed image spectrum has only a fraction of the original source spectrum in the direction facing the receiver aperture. As mentioned before, this causes distortion in the y-axis of the reconstructed source region.

3 Conclusions

We derived a method to reconstruct an image by identifying that the inverse transfer function is equal to We stated that Green's equation describes the wave-field pattern of a single source point and derived a method to reconstruct a source distribution by inverse filtering. We showed that the inverse of the forward propagation transfer function (Green's equation) is equal to it conjugate. We therefore just need to convolve the receiver data with the conjugate Green's equation to reconstruct our original source distribution. This is analagous to treating our receiver data as the source and generating the resulting wave-field which superimposes in a way that reconstructs our image. We observed that our reconstructed image had a good approximation of the source region but had significant distortion in the y-aixs. We hypothesized that this was a consequence of only sampling part of the wavefield in the direction of the receiver aperture. Finally, we analyzed the spectrum of the reconstructed source region and confirmed our hypothesis given it only contained spectrum information in the direction of the receiver.

4 References

Lee, Hua. *Acoustical Sensing and Imaging.* CRC Press, Taylor amp; Francis Group, 2016.

5 Code

```
import numpy as np
import matplotlib.pyplot as plt
from Lab1 import point_source
from scipy.signal import convolve2d

# Generate 2D Wave Field pattern extending it to y = -60 labmda data hyper-plane.
def visualize_wavefield(greens_amp=None, wavelength=1, relative_radius=30,
    sample_spacing= 1/4, pat_title=None, spec_title=None, data_plane=None):

    # Plot wave field pattern
    first_pt_flag = True
    for x,y in np.array([[0,10], [10,0], [0,-10], [-10,0], [-8,-6], [8,-6]]):
        wave_field_pattern = point_source.generate_wave_field_pattern(relative_radius,
            sample_spacing, wavelength,
                                                                  xshift=x, yshift=y,
                                                                      greens_amp=greens_amp,
                                                                      data_plane=data_plane)
        if first_pt_flag:
            wave_field_6pt_pattern = wave_field_pattern
            first_pt_flag = False
        else:
            wave_field_6pt_pattern += wave_field_pattern

    pattern_title = pat_title if pat_title else 'Source Wave-Field Pattern \nand
        Aperture Sampling'
    ax = point_source.plt_plot(np.abs(wave_field_6pt_pattern), title=pattern_title,
            xlabel='X position', ylabel='Y position')

    # Plot data plane at -60
    i, j = wave_field_6pt_pattern.shape
    y_v = (i-1)*np.ones(j)
    x_v = np.arange(j)
    plt.scatter(x_v, y_v, color='r', s=2)
    plt.show()

    return wave_field_6pt_pattern

def image_reconstruction(wave_field_at_receiver, radius, sample_spacing, wavelength):
    # Plot wave field pattern

    wave_field_at_receiver = np.conjugate(wave_field_at_receiver)
    first_pt_flag = True
    for x, amplitude in enumerate(wave_field_at_receiver):
        wave_field_pattern = point_source.generate_wave_field_pattern(radius,
            sample_spacing, wavelength,
```

```
                                                                xshift=x*sample_spacing
                                                                    -radius, yshift=-60,
                                                                greens_amp=amplitude,
                                                                    data_plane=None,
                                                                    inverse=True)
        if first_pt_flag:
            reconstructed_image = wave_field_pattern
            first_pt_flag = False
        else:
            reconstructed_image += wave_field_pattern

    pattern_title = 'Reconstructed Source Region'
    reconstructed_image = np.conjugate(reconstructed_image)
    point_source.plt_plot(np.abs(reconstructed_image), title=pattern_title,
            xlabel='X position', ylabel='Y position')

    # Overlay pointsource locations
    for x, y in np.array([[0, 10], [10, 0], [0, -10], [-10, 0], [-8, -6], [8, -6]]):
        x_idx = x/sample_spacing + (2*radius/sample_spacing)//2
        y_idx = -(y/sample_spacing - (2*radius/sample_spacing)//2)
        plt.scatter(x_idx,y_idx,color='r',s=3)

    # Plot wave field spectrum
    wave_field_spectrum = np.fft.fftshift(np.fft.fft2(reconstructed_image, s=(512, 512)))
    spectrum_title = 'Spectrum of Reconstructed Source Region'
    point_source.plt_plot(np.abs(wave_field_spectrum), title=spectrum_title,
            xlabel='$f_x$', ylabel='$f_y$')
    plt.show()

    # # Alternative: create wavefield by convolving aperture data with conjugate impulse
        response.
    # source = np.zeros((int(2*radius/sample_spacing) + 1,int(2*radius/sample_spacing) +
        1))
    # source[-1, :] = wave_field_at_receiver
    # wave_field_pattern = point_source.generate_wave_field_pattern(3*radius,
        sample_spacing, wavelength=wavelength,
    #                                              greens_amp=None)
    # wave_field = convolve2d(source, wave_field_pattern, mode='valid')
    # wave_field = np.conj(wave_field)
    # point_source.plt_plot(np.abs(wave_field), title="Magnitude Distribution of
        \nReconstructed Source Region",
    #         xlabel='X position', ylabel='Y position')
    #
    # plt.show()


if __name__ == '__main__':
    wave_field_2d = visualize_wavefield(data_plane=-60)
    wave_field_at_receiver = wave_field_2d[-1, :]
    print(wave_field_at_receiver.shape)
    image_reconstruction(wave_field_at_receiver, 30, 1 / 4, 1)
```