Using holography to measure particle extinction and surface topography

by

Peter A. Anderson

B.S., Mississippi State University, 2013

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Physics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2021

Approved by:

Major Professor
Matthew J. Berg

Copyright

Abstract

Since its conception, holography has been applied to everything from microscopy and interferometry to art and security. Two separate applications of holography are discussed in detail. In the first project, holography is used to obtain a measurement of the extinction cross section of various micron sized particles, which are compared to the discrete dipole approximation (DDA) cross sectional values. We are also interested in the structure of “fairy castles”, a model developed to explain the photometric behavior of the surface of the moon. A holographic solution in conjunction with phase unwrapping is obtained, and the basics of phase unwrapping are discussed.

Table of Contents

List of Figures

Acknowledgments

I would like to thank the U.S. Army Research Office and the Directorate of Research Science for funding these projects.

Dedication

For my brothers, who are my candles in the dark within this demon haunted world.

Chapter 1

Introduction

Holography was invented by Dennis Gabor, an accomplishment for which he was awarded the Nobel Prize in 1971. In his article, titled “A new microscopic principle” [3], Gabor outlines a method to account for correcting spherical aberration of electron lenses. The setup in Gabor’s experiment is of simple design: it consists of a coherent light source illuminating a small object that is deposited on a window. The light scattered off the object interferes with the light from the source, and the interference pattern (hologram) is then recorded on a recording medium.

In this dissertation, two applications of holography for characterization are discussed. Chapter 2 will detail an experiment where holography is used to measure the extinction cross section of various particles directly from a hologram. Chapter 3 will cover another experiment where holography is used to measure rough surfaces.

1.1 Measuring extinction cross section of particles

The goal of this experiment was to make measurements of the extinction cross section (i.e., the shadow) of micron sized particles from a contrast hologram. The extinction cross section is a parameter useful for particle characterization.

When a particle is larger than the wavelength of light illuminating it, the extinction cross section is approximately double the size of the geometric cross section of the particle in question, a phenomenon known as the extinction paradox. Our measurements were compared to discrete dipole approximation (DDA), transition-matrices (TM), and Mie Theory (MT) simulations, and largely fell within ±10% of the predicted value. Our analysis also allows us to approximate the absorption cross section.

1.2 Phase unwrapping of surfaces

The motivation behind the second project begins with the surface of the moon. The lunar surface exhibits unusual photometric behavior. It is known that the moon is significantly brighter, both as a whole and looking at individual areas, when it is full. In his paper "A Theoretical Photometric Function for the Lunar Surface", Bruce Hapke describes how the fine powder on the lunar surface could pack in such a way that would account for this [1; 4]. The structure the powder forms has been referred to as "fairy castles", which is a fractal structure resembling tree branches or dentrites.

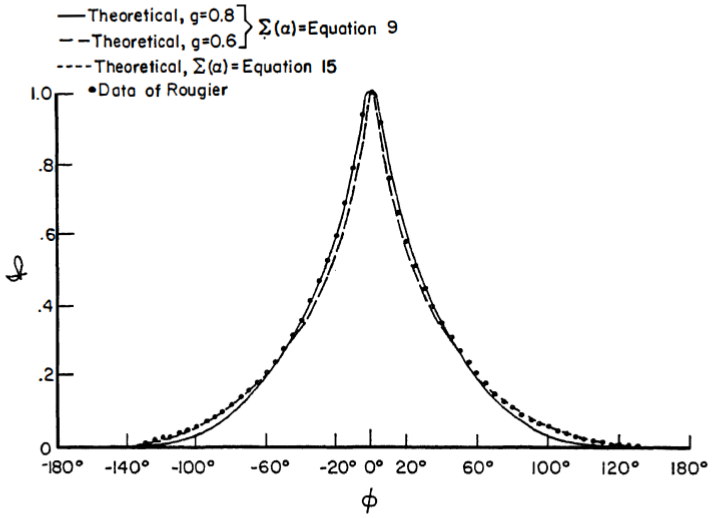

Figure 1.1: Total moon intensity. ϕ is the angle of the moon's orbit around Earth, where $\phi = 0$ is a full moon [1].

The fairy castles' size is on the order of tens of microns, making holography a viable candidate for measuring. Using a microscope is problematic because the dust on the moon's surface is charged by the solar wind and would deposit onto the apparatus. A microscope only works effectively if you can get close to the subject you want to observe. Holography allows these measurements to be made from a safe distance, where both the equipment and surface structures can be preserved, while maintaining a resolution that can match a microscope.

The goal here was to measure and reconstruct micro-structures in lunar regolith. A hologram of the surface would be recorded, and phase unwrapping would be used to create a 3D topographical map of the surface, which will be referred to as a “depth map”. As a proof of concept, this technique was tested on more readily available terrestrial objects. A degree of success was found with the surface reconstruction of a U.S. quarter coin. While the phase unwrapped map has not been calibrated to give us the heights of the features on the quarter's surface, the shapes of those features (specifically the date “1788”) are clearly visible.

The basics of phase unwrapping will be briefly covered before further discussion of aliasing and residues. These topics are relevant because the bulk of the experimental data was filled with residues, which rendered many common unwrapping methods unusable. These obstacles present challenges to overcome and implications for future work to improve the technique.

Chapter 2

Measuring extinction

2.1 Theory

Consider a particle in a non-absorbing medium that is illuminated by a time harmonic plane wave:

$$\mathbf{E}^{\mathrm{inc}} = E_0^{\mathrm{inc}} \exp(ikz - i\omega t)\ \hat{\mathbf{x}}$$

where k is the wavenumber, ω is the angular frequency, $\hat{\mathbf{x}}$ is the polarization angle, and z is the position parallel to the direction of propagation. Now, imagine the particle is surrounded by an arbitrary closed surface S, with a portion of S is making contact with a nearby sensor, as illustrated in Fig. 2.1. The total energy passing through the surface is given by [5]:

$$W = -\oint_S \langle \mathbf{S} \rangle_t \cdot \hat{\mathbf{r}}\ dA \tag{2.1}$$

where $\langle \mathbf{S} \rangle_t$ is the time-averaged Poynting vector:

$$\langle \mathbf{S} \rangle_t = \frac{1}{2\mu_0} \mathrm{Re}\{\mathbf{E} \times \mathbf{B}^*\} \tag{2.2}$$

and $\mathbf{E}$, $\mathbf{B}$, and μ_0 are the electric field, magnetic field, and the magnetic permeability in a vacuum, respectively.

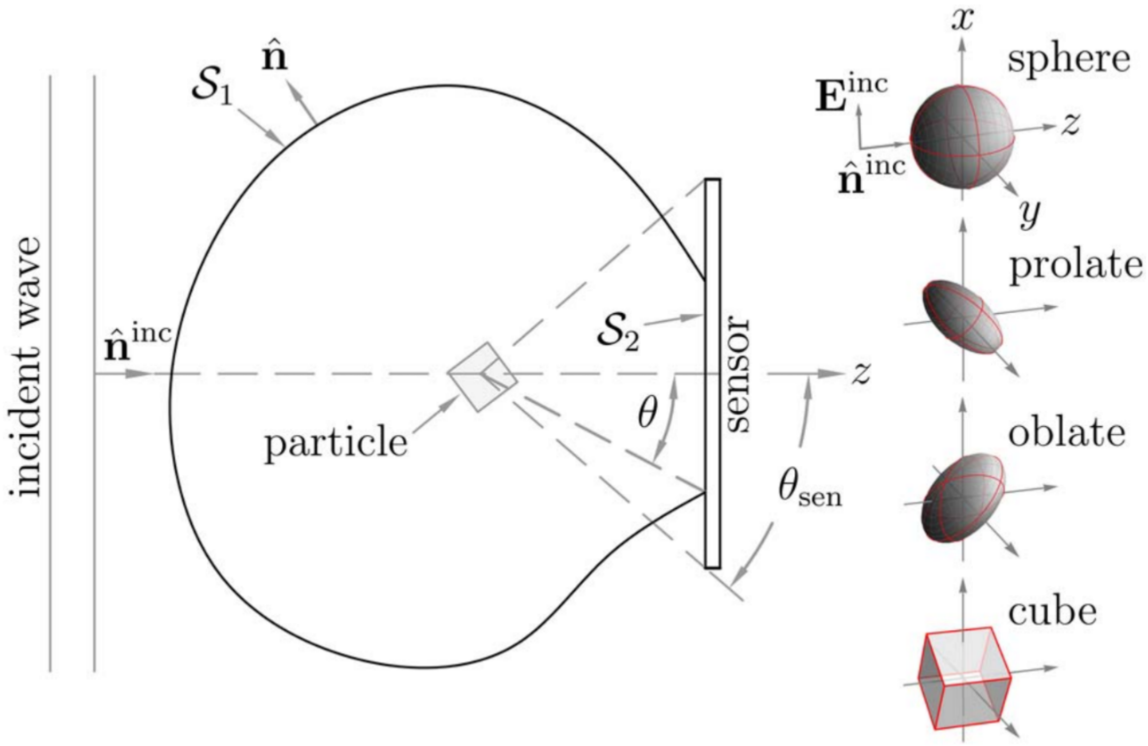

Figure 2.1: Illustration of the surface of integration $S = S_1 + S_2$. The right side displays some particle shapes that were tested. [2]

Both Eq. 2.1 and Eq. 2.2 can be broken into parts corresponding to components from the incident beam, particle absorption, particle scattering, and extinction, with these parts denoted by the superscripts *inc*, *abs*, *sca*, and *ext*, respectively. The relationship between these energy components is:

$$W^{\mathrm{abs}} = W^{\mathrm{inc}} - W^{\mathrm{sca}} + W^{\mathrm{ext}} \tag{2.3}$$

$$\begin{aligned} W^{\mathrm{abs}} &= -\oint_S \mathbf{S}^{\mathrm{abs}} \cdot \hat{\mathbf{r}}\, dA, \\ W^{\mathrm{inc}} &= -\oint_S \mathbf{S}^{\mathrm{inc}} \cdot \hat{\mathbf{r}}\, dA, \\ W^{\mathrm{sca}} &= \oint_S \mathbf{S}^{\mathrm{sca}} \cdot \hat{\mathbf{r}}\, dA, \\ W^{\mathrm{ext}} &= -\oint_S \mathbf{S}^{\mathrm{ext}} \cdot \hat{\mathbf{r}}\, dA \end{aligned} \tag{2.4}$$

The total energy flow through S can also be represented as time averaged Poynting vectors:

$$\langle \mathbf{S} \rangle_t = \langle \mathbf{S}^{\text{inc}} \rangle_t + \langle \mathbf{S}^{\text{sca}} \rangle_t + \langle \mathbf{S}^{\text{ext}} \rangle_t \tag{2.5}$$

This can be simplified further by recognizing that the incident beam contributes a net zero energy, and conservation of energy gives:

$$W^{\text{ext}} = W^{\text{abs}} + W^{\text{sca}} \tag{2.6}$$

$$W^{\text{abs}} = -\int_V \nabla \cdot \langle \mathbf{S} \rangle_t \, dV \tag{2.7}$$

Remember that S is a closed surface that completely surrounds the particle. S_1 is the arbitrary surface surrounding the particle except for the part touching the sensor and S_2 is the portion making contact with the sensor surface, so that $S = S_1 + S_2$. The energy terms can be expressed as:

$$W^{\text{abs}} = -\left[\int_{S_1} \langle \mathbf{S}^{\text{abs}} \rangle_t \cdot \hat{\mathbf{n}} \, da + \int_{S_2} \langle \mathbf{S}^{\text{abs}} \rangle_t \cdot \hat{\mathbf{z}} \, da \right] \tag{2.8}$$

$$W^{\text{ext}} = -\left[\int_{S_1} \langle \mathbf{S}^{\text{ext}} \rangle_t \cdot \hat{\mathbf{n}} \, da + \int_{S_2} \langle \mathbf{S}^{\text{ext}} \rangle_t \cdot \hat{\mathbf{z}} \, da \right] \tag{2.9}$$

$$W^{\text{sca}} = \int_{S_1} \langle \mathbf{S}^{\text{sca}} \rangle_t \cdot \hat{\mathbf{n}} \, da + \int_{S_2} \langle \mathbf{S}^{\text{sca}} \rangle_t \cdot \hat{\mathbf{z}} \, da \tag{2.10}$$

2.1.1 The f-curve

The f-curve is an integral over the contrast hologram, and it has units of distance squared. The f-curve is defined as:

$$f(\theta) = \mathcal{I}_0^{\text{sen}}(\theta) - \mathcal{I}^{\text{sen}}(\theta) \tag{2.11}$$

Eq. 2.12 and Eq. 2.13 are integrals over the reference beam and hologram, where I_0 is the intensity of the incident wave. The difference of these two terms is the contrast hologram.

$$\mathcal{I}_0^{\text{sen}}(\theta) = \frac{1}{I_0} \int_{S_2} \langle \mathbf{S}^{\text{inc}} \rangle_t \cdot \hat{\mathbf{z}} \; da \tag{2.12}$$

$$\mathcal{I}^{sen}(\theta) = \frac{1}{I_0} \int_{S_2} \langle \mathbf{S} \rangle_t \cdot \hat{\mathbf{z}} \; da \tag{2.13}$$

It's important to keep in mind that S_2 has the built in θ dependence and is identical for both $\mathcal{I}_0^{\text{sen}}(\theta)$ and $\mathcal{I}^{\text{sen}}(\theta)$.

How does the f-curve behave? To begin, Eq. 2.11 can be expressed as follows:

$$f(\theta) = \frac{1}{I_0} \left[\int_{S_1} \langle \mathbf{S}^{sca} \rangle_t \cdot \hat{\mathbf{n}} \; da - W^{sca} + \int_{S_1} \langle \mathbf{S}^{ext} \rangle_t \cdot \hat{\mathbf{n}} \; da + W^{ext} \right] \tag{2.14}$$

In the limit $\theta \to 0$, the first and third terms in 2.14 approach W^{sca} and $-W^{ext}$ respectively, and $f(\theta) \to 0$. This makes intuitive sense when referring back to Eq. 2.12 and Eq 2.13; the surface of integration S_2 is nonexistent in this limit.

Eq. 2.11 can also be expressed as:

$$f(\theta) = \frac{1}{I_0} \left[W^{abs} + \int_{S_2} \langle \mathbf{S}^{sca} \rangle_t \cdot \hat{\mathbf{z}} \; da - W^{sca} + \int_{S_1} \langle \mathbf{S}^{ext} \rangle_t \cdot \hat{\mathbf{n}} \; da \right] \tag{2.15}$$

As θ increases from 0, the f-curve will rise to a peak, then oscillate [6]. Particles that are larger than λ scatter more strongly in the forward direction, so the second term approaches W_{sca}. The fourth term starts approaching zero, meaning that $f(\theta)$ approaches W_{abs} asymptotically. If there is no absorption, then $f(\theta) \to 0$, as in Fig. 2.3.

2.2 Experiment

Our setup, as shown in Figure 2.2, consists of (in series) a class III pulsed Nd:YLF 526.5nm laser, a beam expander, a spatial filter, a particle deposited on a window (from here, it will be referred to as the "stage"), and a charge-coupled device (CCD) detector. The lenses chosen for the beam expander widen the beam enough to cover the entire detector. The particles chosen for measurement are soda-lime glass spheres, which are approximately 50 μm in size, and ragweed pollen, about 15 to 25 μm in diameter.

A window placed immediately after the laser serves as a beam splitter, and the resulting second beam is directed toward a photodiode to record the pulse intensity. The measured intensities are used to proportionally adjust the reference pulse and the hologram pulse intensities before constructing the contrast hologram. This is done to compensate in variations of pulse intensity.

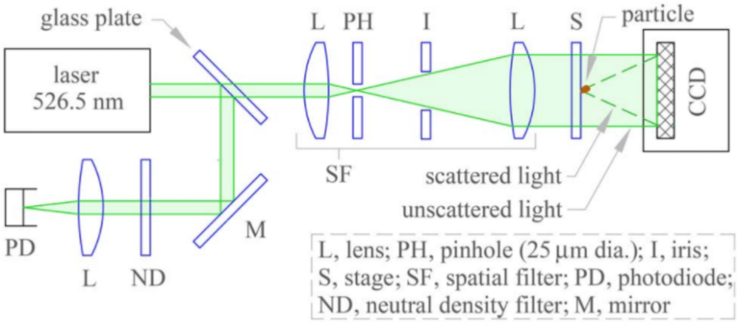
glass plate
laser
526.5 nm
L
PH
I
L
S
particle
CCD
SF
scattered light
unscattered light
PD
L
ND
M
L, lens; PH, pinhole (25 µm dia.); I, iris;
S, stage; SF, spatial filter; PD, photodiode;
ND, neutral density filter; M, mirror

Figure 2.2: Setup for measuring C_{ext} [2].

In this experiment, the hologram is recorded first, and a reference image is recorded afterwards. Between recordings, the particle is removed from the stage with an air canister, which ensures that the stage maintains its position. The images' pixel values are then scaled proportionally according to the voltage readouts from the photodiode, as mentioned previously. From here, the contrast hologram is generated by taking the difference between the two images' respective pixels.

To generate $f(\theta)$, the code adds up the pixel values within an area around the center of the contrast hologram, as shown in the inset of Fig. 2.3. The area is increased until it hits the edge of the hologram. Two functions are interpolated from the f-curve: f_{top}, which is interpolated from the maxima, and f_{bot}, which is interpolated from the minima. A trend curve, defined as the average or midpoint between f_{top} and f_{bot} is graphed, and because f_{top} ends at the first maxima, the trend terminates there as well. The trend curve is then extrapolated to $\theta \rightarrow 0$, giving us an estimate for C^{ext}. Fig. 2.3 also shows how this approach

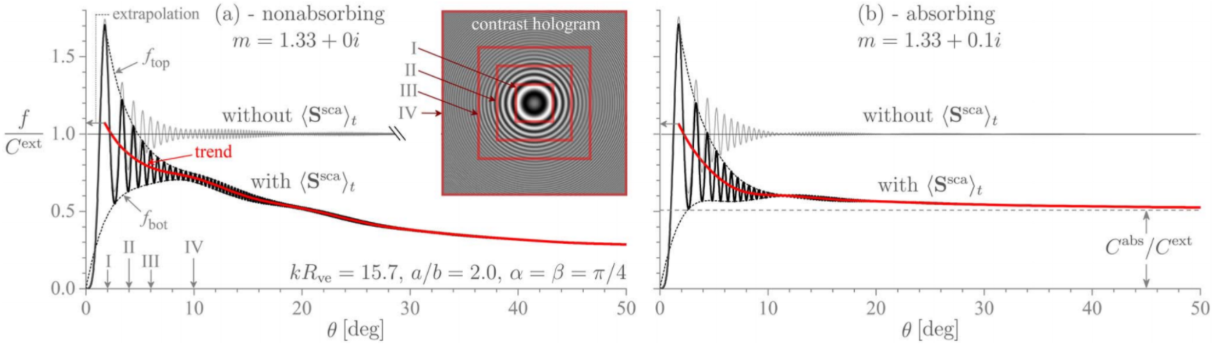

Figure 2.3: Plots of $f(\theta)$ curves for an oblate spheroid. The index of refraction is $m = 1.33{+}0i$ for (a), which is nonabsorbing, and $m = 1.33{+}0.1i$ for (b), which accounts for absorption [2].

is done.

2.3 Results

Fig. 2.3 shows simulated f-curves for an oblate spheroid with size parameter kR_{ve} (R_{ve} is the spherical volume equivalent radius), $a/b = 2.0$ (the ratio between the major and minor axes), and $\alpha = \beta = \frac{\pi}{4}$ (the Euler angles with respect to $\hat{\mathbf{n}}_{\text{inc}}$). The curve is also normalized with respect to the accepted values of the extinction cross section, which are calculated using TM.

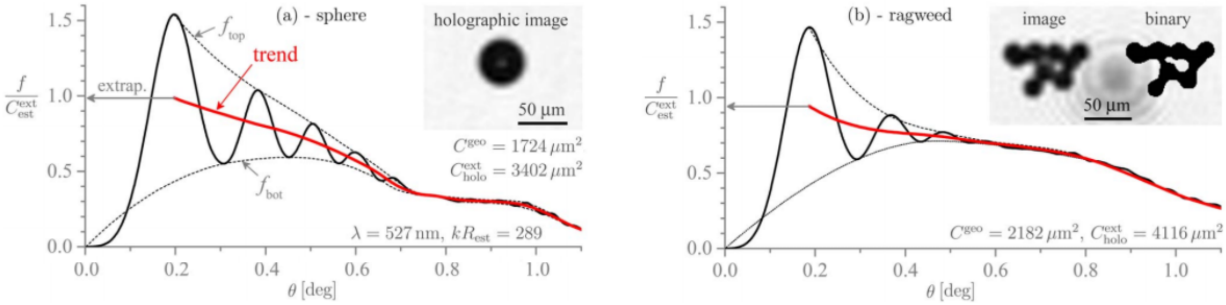

Figure 2.4: F-curves for a glass sphere and a cluster of ragweed pollen. The dimensions of the glass sphere are known from the manufacturer, so this was used for calibration [2].

Fig. 2.4 shows the normalized f-curves from laboratory measurements. Inset (a) is for a 50 μm glass sphere, and inset (b) is for a cluster of ragweed pollen. Reconstructed images are also included of the respective particles. As before, the red trend line is an average between the interpolated minima and maxima functions, with the cutoff at the first maxima. The cutoff is then extrapolated to $\theta = 0$ (as shown by the arrow), which indicates how close the measurement is to the accepted C_{ext} of the subjects in question.

Because these particles are larger than the wavelength of the beam, the extinction cross section can be estimated from the particle's geometrical cross section, where $C_{\mathrm{ext}}^{\mathrm{est}} \approx 2C_{\mathrm{geo}}$. The glass sphere's size is given to be $49 \pm 1.4\mu$m, and the pollen cluster's C_{geo} is calculated from a binarization of the reconstructed image. C_{geo} is included in Fig. 2.4.

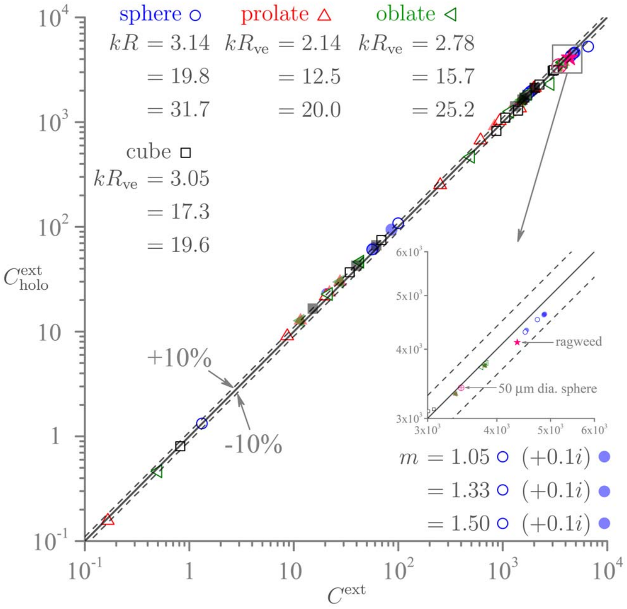

Figure 2.5: Comparison of our measurement of $C_{\mathrm{holo}}^{\mathrm{ext}}$ versus C^{ext} calculated using DDA/TM/MT. Hollow data points indicate no absorption, and solid data points include absorption. The inset highlights the 50μm glass spheres and ragweed pollen measurements mentioned previously [2].

A survey using particles of different shapes, sizes, and refraction indices was also conducted to determine how generally our method could be applied. Fig. 2.5 is a comparison of using our methods to calculate C^{ext} versus the true values found using Mie Theory (MT), T-matrices (TM), or the discrete dipole approximation (DDA). A majority of the f-curve measurements fall within ±10% of the true values, indicating that this method is reliable across a wide range of particle shapes, sizes, and refraction indices.

Chapter 3

Applying Phase Unwrapping to Rough Surfaces

3.1 Theory

A phase map is a 2D set of data of phase values. When the values at any position are confined between $\pm\pi$, the phase map is "wrapped". The wrapped phase map is denoted as $\psi(x, y) = \phi(x, y) + 2n\pi$, where $\phi(x, y)$ is the true phase, and n is an integer that forces $\phi(x, y)$ between $\pm\pi$. The idea of phase unwrapping is to "connect" the discontinuities in a phase map wherever there is a jump from π to $-\pi$ and vice versa in order to find the "true" phase ϕ. From there, it is a matter of scaling to obtain a depth map of the surface.

Consider Fig. 3.2, which is a one dimensional example of a true phase and wrapped phase. Starting from our first data point, one method of unwrapping the wrapped phase is to look at how far apart the next data point is from our current position. If the next data point is more than $\pm\pi$ away from the current point, the next point's phase is adjusted by $\pm 2n\pi$ such that it is within $\pm\pi$ of the current data point. The next two data points are then compared, and this continues until the end is reached. Ideally, when we apply this algorithm to the wrapped phase, we get a much smoother curve, like the one in Fig. 3.1. This is essentially the algorithm that MATLAB's built-in unwrapping algorithm uses.

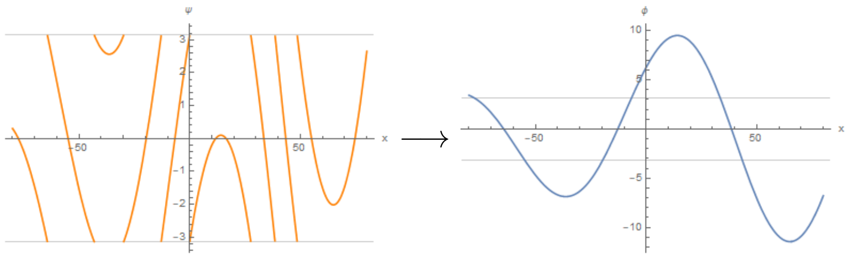

Figure 3.1: Wrapped phase ψ and unwrapped phase ϕ after using rudimentary unwrapping code. ψ is a 1D example of an input signal. Units on the x-axis are arbitrary.

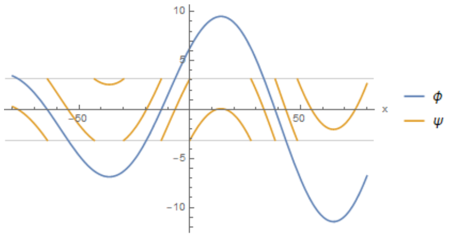

Figure 3.2: Wrapped phase ψ and unwrapped phase ϕ on the same graph for comparison. Units on the x-axis are arbitrary.

While phase unwrapping is trivial in the one dimensional case, it is considerably more difficult when working in higher dimensions. The topic is extensive, as there are many papers and even textbooks about the subject. [7–14] In the 2D case, noise and aliasing can lead to the presence of residues, which can make phase unwrapping difficult if not impossible if they are not properly dealt with.

3.1.1 Aliasing

Aliasing is a consequence of not being able to sample enough of a wrapped phase to acquire the true phase. This happens when phase shifts greater than π happen between two data points. The problem is further aggravated when noise is present in the phase map, and significant noise can cause the true phase to be unrecoverable.

Fig. 3.3 is an example that illustrates how aliasing can affect phase unwrapping. There are sections of the unwrapped phase map that are smooth, but the unwrapping fails at locations where the slope is high. Fig. 3.4 shows the same case, but with a slightly higher sampling rate. This small difference is sufficient to allow the phase unwrapping algorithm

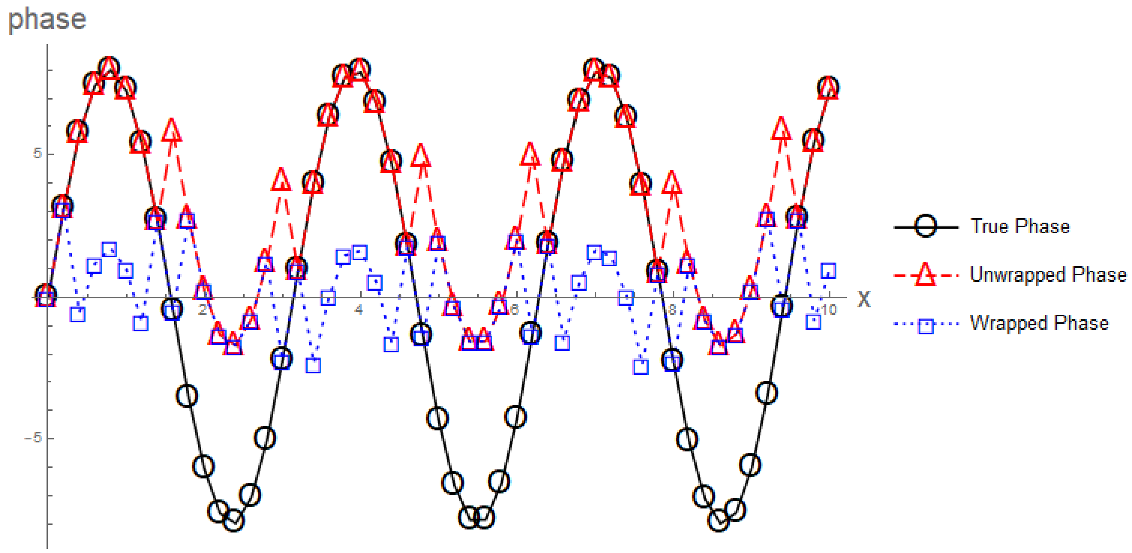

Figure 3.3: Sinusoidal wave with amplitude $A = 8$ and angular frequency $\omega = 2$. Failed phase unwrapping with a data point every $x = 0.20$. The black curve is the true phase, the blue curve is the wrapped phase, and the red curve is the unwrapped phase. The shapes are the data points considered when unwrapping is done. Note that the phase unwrapping process fails at locations with very steep slopes.

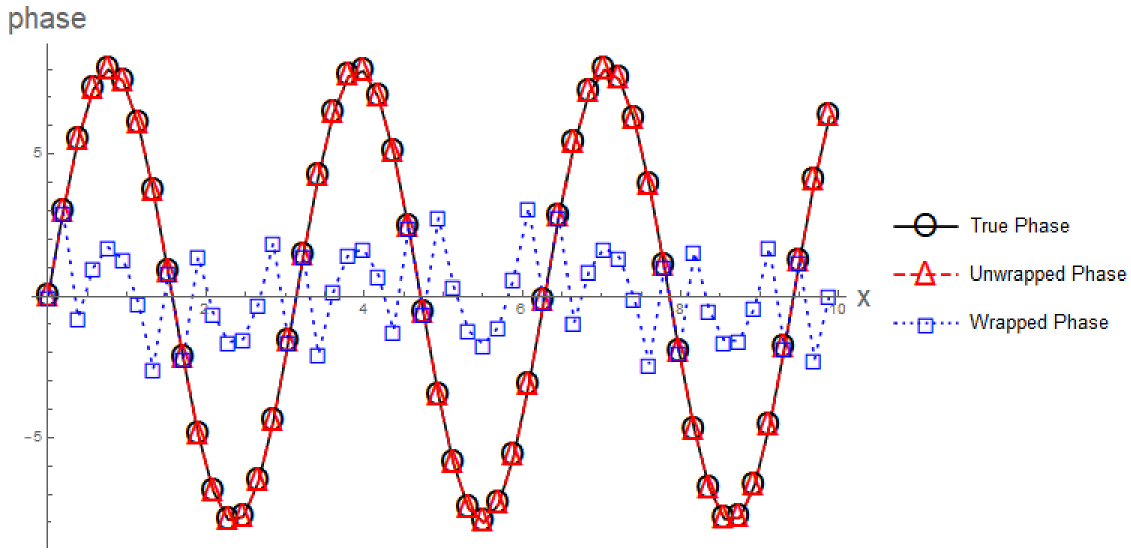

Figure 3.4: $A = 8$ and $\omega = 2$. Successful phase unwrapping with a sample every $x = 0.19$ units. A small change in the sampling rate can lead to more reliable results.

to succeed.

3.1.2 Residues

One other common issue that phase unwrapping algorithms have to address is that phase unwrapping is not always a path independent process. Consider a square of four pixels and define the change in their wrapped phase as follows [7]:

$$\begin{aligned}
\Delta_1 &= \mathcal{W}\{\psi(m,n+1)-\psi(m,n)\},\\
\Delta_2 &= \mathcal{W}\{\psi(m+1,n+1)-\psi(m,n+1)\},\\
\Delta_3 &= \mathcal{W}\{\psi(m+1,n)-\psi(m+1,n+1)\},\\
\Delta_4 &= \mathcal{W}\{\psi(m,n)-\psi(m+1,n)\}
\end{aligned} \tag{3.1}$$

The sum of those changes is defined as:

$$q = \sum_{i=1}^{4} \Delta_i \tag{3.2}$$

The sum q is sometimes referred to as the “charge” of a residue [7]. If $q = 0$, then no residue is present. Many algorithms incorporate “branch cutting”, which draws lines between a nearby residue of opposing charge so that they net zero. Residues near the edges of the map have a branch connecting to the edge if no opposing residues are nearby. The algorithms then begin unwrapping the phase and works around the branches rather than across them. So long as the unwrapping path does not cross those branches, phase unwrapping becomes a path independent process.

3.2 Experiment

Our setup, as illustrated in Fig. 3.5, is essentially a Michelson interferometer and consists of a 100 mW solid state 532 nm continuous wave (CW) laser, a neutral density filter (NDF),

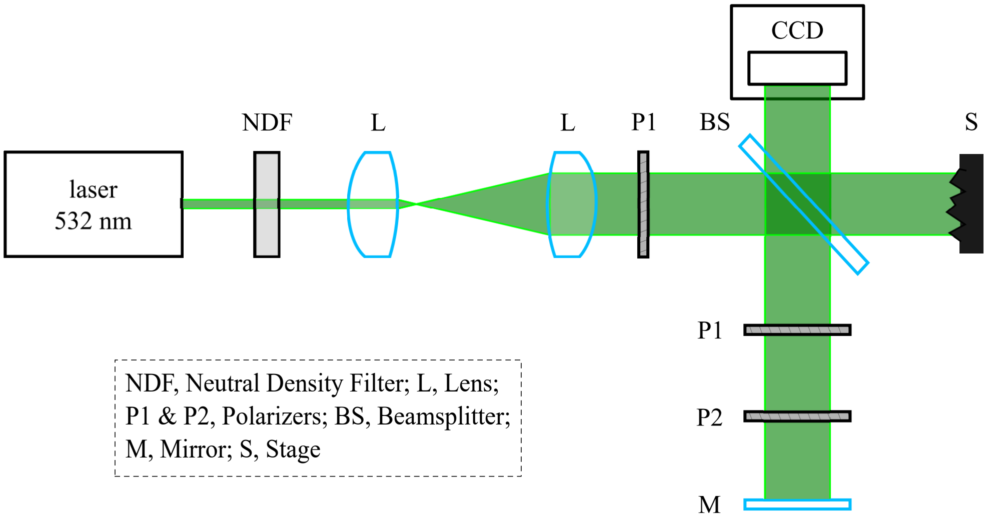
CCD
laser
532 nm
NDF
L
L
P1
BS
S
P1
P2
M
NDF, Neutral Density Filter; L, Lens;
P1 & P2, Polarizers; BS, Beamsplitter;
M, Mirror; S, Stage

Figure 3.5: Our setup for making surface measurements is a Michelson style interferometer. The lenses are used for expanding the beam to cover the entire sensor, and the polarizers P1 are set at the same angle, ensuring consistent polarization. Polarizer P2 allows us to adjust the brightness of the reference beam. Ideally, the reference beam should have the same intensity as the surface scattered light to ensure maximum interference.

a beam expander, a polarizer, and a beamsplitter, dividing the beam into two paths. The reference beam passes through two linear polarizers, with the first set at the same angle as the polarizer before the beamsplitter, and the second polarizer is used to control the brightness of the reference beam. A mirror reflects the reference beam back through the beamsplitter and on to the recording medium, a charge-coupled device (CCD) in this case.

The second beam simply illuminates the sample surface. Reflection of the beam off the target surface is then directed to the same CCD by the beamsplitter. The beams then converge on the CCD, creating the hologram, which is recorded. The hologram is then used to generate an in-focus reconstructed image of the surface, and the argument of the reconstructed image provides the phase map (see appendix A). A high power laser is necessary to adequately illuminate the target surface. As a consequence, a spatial filter could not be used in this setup, as the laser would damage the pinholes used for filtering.

Fig. 3.7 are the inputs used for testing our unwrapping code (located in appendix B), and each input has different signal to noise ratios (SNR). When the unwrapping code is applied, both the noiseless and the SNR = 2.5 phase maps are successfully unwrapped. However, the SNR = 1.67 phase map fails to unwrap, as indicated by the resultant bumps

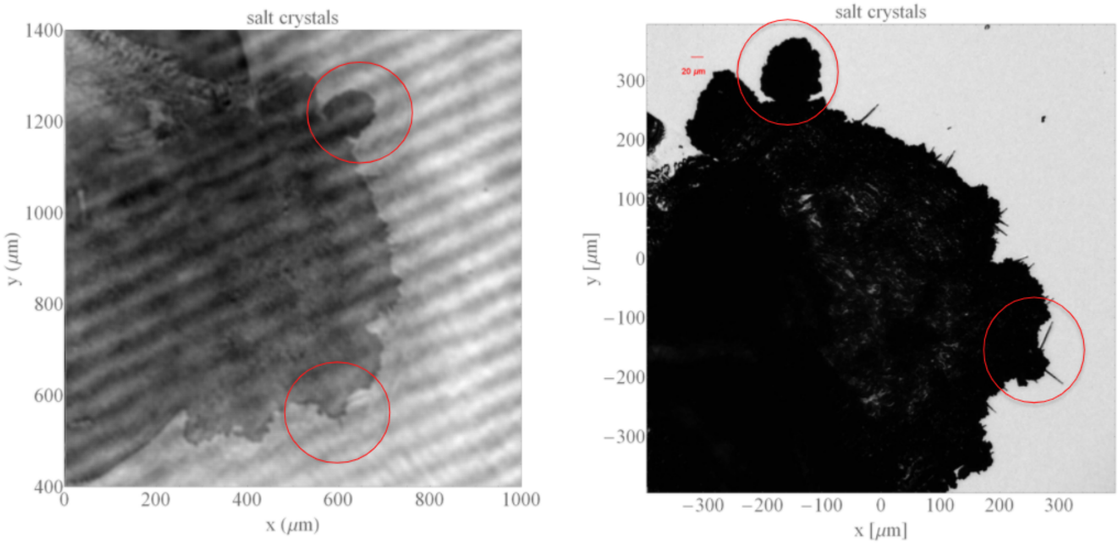

Figure 3.6: Reconstruction and microscope image of salt crystals, respectively. Note that, though the edge features are easy to see, the surface features are not very visible. Image resolution is on the order of a few microns.

and streaking pattern. From here, Goldstein's branch cut algorithm and Phase Unwrapping Max Flow/Min Cut (PUMA) were adopted [8].

Several subjects were measured, including a New York State Quarter. Fig. 3.9 is the reconstructed image of the quarter from the hologram, and Fig. 3.10 is the unwrapped phase map, now a depth map, using the PUMA algorithm and includes an overlay of a New York quarter for comparison. Goldstein's branch cut algorithm failed to unwrap the phase due to the substantial number of residues in the phase map.

3.3 Results

The streaking in Fig. 3.8 is a consequence of aliasing and residues in the phase map [7; 9] as well as how the unwrapping process is done, which unwraps one column after another. The streaks can be suppressed when neighboring data points in both dimensions are accounted for at the same time. [10–13].

Proper calibration was not performed to ensure the scale of the depth map is accurate. However, the PUMA code appears to have reasonably scaled the size of the quarter if you compare the dimensions given in the first image of Fig. 3.10 (9.5mm across). A quarter has a diameter of approximately 24.1mm and a thickness of 1.7mm. From the depth map in Fig. 3.10, the “1788” is clearly visible, as well as some other nearby impressions, which alone demonstrates that this method of imaging and characterizing surfaces does work.

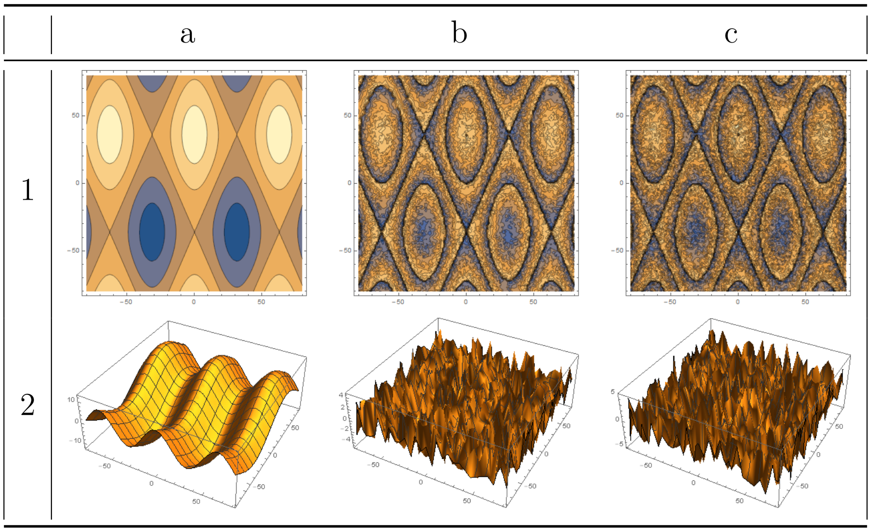

Figure 3.7: Phase map example, used for testing original phase unwrapping code. 1a is a contour plot of 1b, etc. Columns b and c are the wrapped phase maps with SNR = 2.5 and SNR = 1.67, respectively.

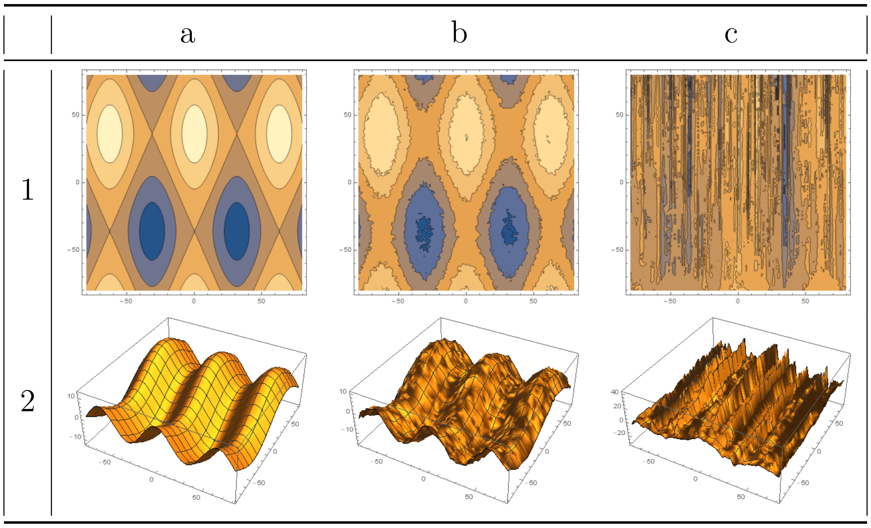

Figure 3.8: Phase map unwrapping results. 1a is a contour plot of 1b, etc. Columns a, b and c are the unwrapped phase maps with zero noise, SNR = 2.5, and SNR = 1.67, respectively. Note the presence of streaks in the unwrapped phase maps in column c.

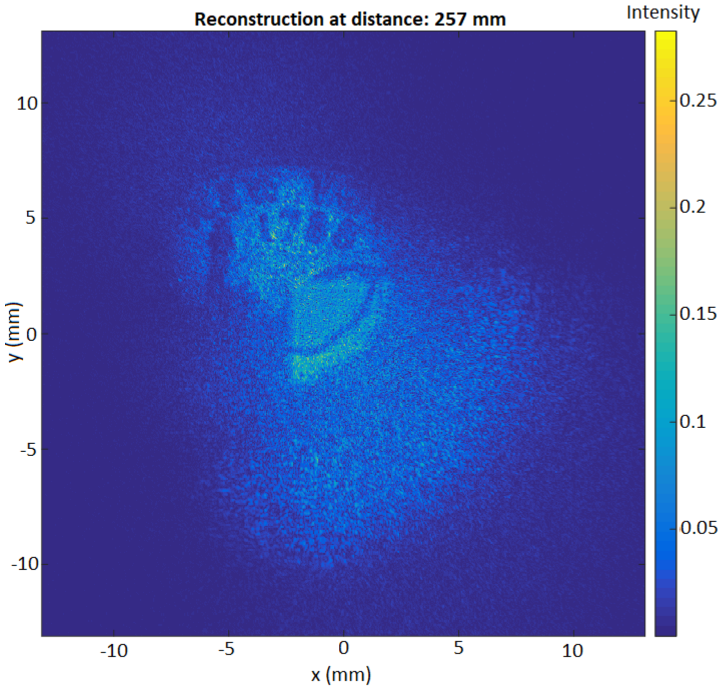

Figure 3.9: Reconstructed image of New York quarter from its hologram.

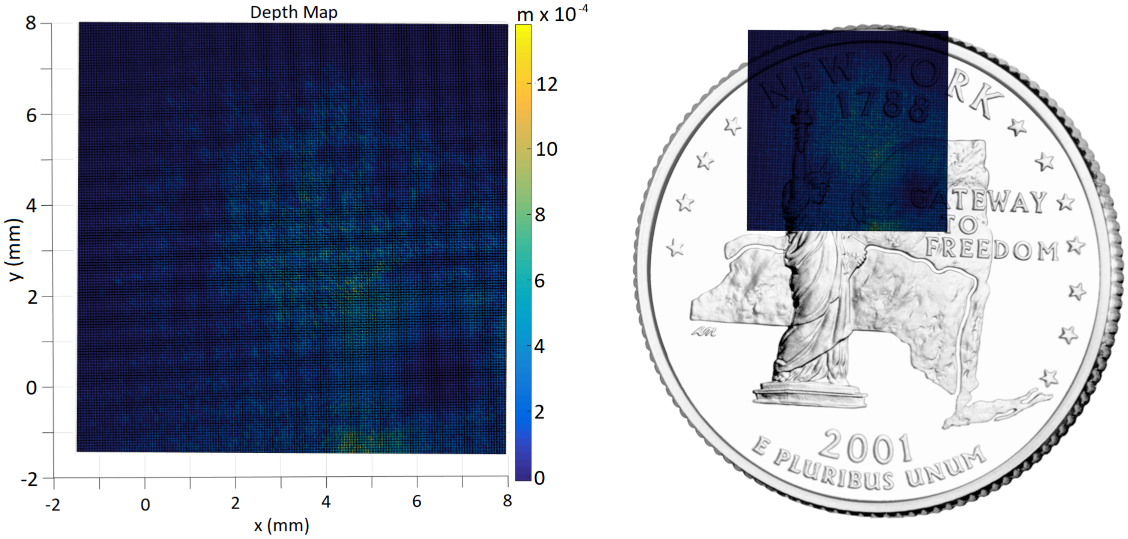

Figure 3.10: Depth map of a New York State quarter, and an overlay with an image of the quarter for comparison. Note that the dark blue areas are higher. Some features of the quarter, such as the “1788”, are patently visible.

Papers and Presentations

M. J. Berg, N. R. Subedi, P. A. Anderson, N. B. Fowler, "Using holography to measure extinction," Opt. Lett. 39 p. 3993-96. Also featured in OSA's Virtual Journal for Biomedical Optics, 9(9), p. 3993 (2014).

M. J. Berg, N. R. Subedi, P. A. Anderson, "Measuring extinction with digital holography: Nonspherical particles and experimental validation" Opt. Lett. 42 (5), p. 1011-1014 (2017).

"Digital Holographic Imaging of Rough Surfaces" ELS-XVII, March 7, 2018.

"Digital Holographic Imaging of Rough Surfaces" CMS Seminar, Kansas State University. April 6, 2018.

"An Introduction to Digital Holography" Journal Club, Kansas State University. November 6, 2019.

Bibliography

[1] B. Hapke. A theoretical photometric function for the lunar surface. *Journal of Geophysical Research*, 68(15):4571–4586, 1963.

[2] M. Berg, N. Subedi, and P. Anderson. Measuring extinction with digital holography: nonspherical particles and experimental validation. *Opt. Soc. Am.*, 42:1011–1014, 2017.

[3] D Gabor. A new microscopic principle. *Nature*, 161:777–778, 1948.

[4] B. Hapke. Packing properties of fine powders and the depth of the lunar dust layer. *Journal of Geophysical Research*, 69(6):1147–1151, 1964.

[5] C. F. Bohren and D. R. Huffman. *Absorption and Scattering of Light by Small Particles.* 1968.

[6] M. Berg, C. M. Sorensen, and A. Chakrabarti. *Opt. Soc. Am.*, 25:1504, 2008.

[7] D. C. Ghiglia and M. D. Pritt. *Two-Dimensional Phase Unwrapping.* New York: Wiley Interscience, 1998.

[8] G. Nehmetallah, R. Aylo, and L. Williams. *Analog and Digital Holography with MATLAB.* SPIE, 2015.

[9] S. Taylor. Application of phase unwrapping to digital holographic microscopy. June 2013.

[10] W. J. Hwang, S. C. Cheng, and C. J. Cheng. Efficient phase unwrapping architecture for digital holographic microscopy. *Sensors*, 11(10), Sept. 2000.

[11] J. Arines. Least-squares modal estimation of wrapped phases: application to phase unwrapping. *App. Opt.*, 42(17), June 2003.

[12] X. Zhang, X. Zhang, M. Xu, H. Zhang, and X. Jiang. Phase unwrapping in digital holography based on non-subsampled contourlet transform. *Opt. Comm.*, 407:367–374, 2018.

[13] C. Zuo, Q. Chen, W. Qu, and A. Asundi. Direct continuous phase demodulation in digital holography with use of the transport-of-intensity equation. *Opt. Comm.*, 309:221–226, 2013.

[14] N. K. Bambha, J. R. Bickford, and Jr. K. K. Klett. Two-dimensional phase unwrapping for digital holography. *Army Research Labs*, Sept. 2012.

[15] H. Bjelkhagen. Color holography: its history, state-of-the-art, and future. *Proc. SPIE*, 6252, June 2006.

Appendix A

Holography and Reconstruction

Put simply, a hologram is an interference pattern that forms when coherent light from a known reference beam interferes with the light scattered off an object. In digital holography, this pattern is recorded using a CCD or CMOS. The data is saved as an image, which is typically grayscale, though utilizing red, green, and blue (RGB) values are useful in cases such as color holography [15].

Because our recording medium is a two dimensional array of pixels and is discrete, we can apply the Fresnel transform to reconstruct an image (sometimes referred to as a "reconstruction") of the subject. This requires selecting the correct value for z, the image distance, in order to bring the reconstruction into focus. The Fresnel transform is:

$$K(x, y, z) = \sum_{m=1}^{N_x} \sum_{n=1}^{N_y} I_{\text{holo}}(x_m, y_n) \exp\left\{ \frac{ik}{2z} \left[(x - x_n)^2 + (y - y_m)^2 \right] \right\} \tag{A.1}$$

where $I_{\text{holo}}(x_m, y_n)$ is the hologram, i is the imaginary constant, k is the wavenumber, and $K(x, y, z)$ is the reconstruction. $\text{Arg}[K(x, y, z_0)]$ provides the phase maps used in Chapter 3, where z_0 is the distance at which the reconstruction is in focus.

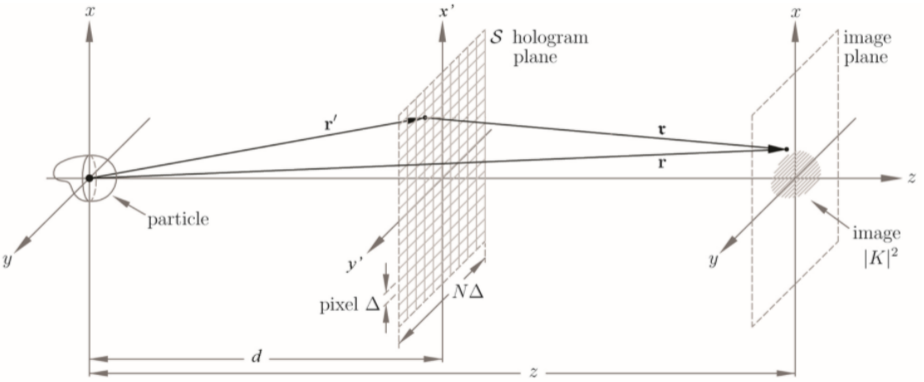

Figure A.1: Holographic reconstruction is analogous to projecting an image on a screen with a lens. In this case, the “lens” is the hologram, and the image can be generated using the Fresnel Transform, where z is the image plane distance.

Appendix B

Mathematica Phase Unwrapping code

The following pages contain the code created early on in an attempt to do 2D phase unwrapping using a known phase map. It wraps the phase of the known 2D phase map, adds noise (which is controlled by the variable “snr”), and attempts to unwrap the phase map. Given its limited ability to do phase unwrapping, this code was used to generate the figures in table 3.7 and table 3.8 as well as perform simple unwrapping on the example figures in Chapter 3.1.

```
ClearAll["Global`*"]
```

The idea is to take an arbitrary function and clip it within a range, using the SawtoothWave function

Generate phase maps

```
SetDirectory["\\\\TRAJAN\\home\\paanders\\profile\\desktop\\research"];

dim = 80;
snr = 0.2;
g[x_, y_] := 2 π (Cos[x/10] + Sin[y/23]);
f[x_, y_] := 2 π SawtoothWave[g[x, y]/(2 π) + 0.0001] - π + snr 2 π RandomReal[{-1, 1}];

array = Table[f[x, y], {x, -dim, dim, 1}, {y, -dim, dim, 1}];
arrayG = Table[g[x, y], {x, -dim, dim, 1}, {y, -dim, dim, 1}];
(*Discretize phase maps*)
plot = ListInterpolation[array, {{-dim, dim}, {-dim, dim}}];
plotG = ListInterpolation[arrayG, {{-dim, dim}, {-dim, dim}}];

ArrayPlot[Transpose[N[arrayG]], ColorFunction → "GrayTones"]
ArrayPlot[Transpose[N[array]], ColorFunction → "GrayTones"]

Plot3D[plotG[x, y], {x, -dim, dim}, {y, -dim, dim}]
Plot3D[plot[x, y], {x, -dim, dim}, {y, -dim, dim}]

i = 2;
neighbor = Table[0.0, {8}];
array2 = Table[0, {x, -dim + 1, dim - 1}, {y, -dim + 1, dim - 1}];
```

```
While[i ≤ 2 dim,
  j = 2;
  While[j ≤ 2 dim, (*Cycle through all pixels in array one by one.*)
   k = 1;
   neighbor[[1]] = array[[i - 1, j - 1]]; (*Assign values to neighboring pixels*)
   neighbor[[2]] = array[[i - 1, j]];
   neighbor[[3]] = array[[i - 1, j + 1]];
   neighbor[[4]] = array[[i, j - 1]];
   neighbor[[5]] = array[[i, j + 1]];
   neighbor[[6]] = array[[i + 1, j - 1]];
   neighbor[[7]] = array[[i + 1, j]];
   neighbor[[8]] = array[[i + 1, j + 1]];
   While[k ≤ 8, (*For neighbors more than π away from central pixel,
    adjust value by 2π. Repeat until within ±π.*)
    While[neighbor[[k]] < array[[i, j]] - π,
     neighbor[[k]] = neighbor[[k]] + 2 π;
    ];
    While[neighbor[[k]] > array[[i, j]] + π,
     neighbor[[k]] = neighbor[[k]] - 2 π;
    ];
    k++;];
   array[[i - 1, j - 1]] = neighbor[[1]];
   (*Assign new values to neighboring pixels*)
   array[[i - 1, j]] = neighbor[[2]];
   array[[i - 1, j + 1]] = neighbor[[3]];
   array[[i, j - 1]] = neighbor[[4]];
   array[[i, j + 1]] = neighbor[[5]];
   array[[i + 1, j - 1]] = neighbor[[6]];
   array[[i + 1, j]] = neighbor[[7]];
   array[[i + 1, j + 1]] = neighbor[[8]];
   array2[[i - 1, j - 1]] = Mean[neighbor];
   (*Take average value of cluster and assign to new array to compare*)
   j++;];
  i++;];

plot2 = ListInterpolation[array, {{-dim, dim}, {-dim, dim}}];
plotAvg = ListInterpolation[array2, {{-dim, dim}, {-dim, dim}}];
ArrayPlot[Transpose[N[array]], ColorFunction → "GrayTones"] (*Corrected*)
ArrayPlot[Transpose[N[array2]], ColorFunction → "GrayTones"] (*Averaged*)
ArrayPlot[Transpose[N[arrayG]], ColorFunction → "GrayTones"]
(*Starting function we expect*)

Plot3D[plot2[x, y], {x, -dim, dim}, {y, -dim, dim}]
Plot3D[plotAvg[x, y], {x, -dim, dim}, {y, -dim, dim}]
Plot3D[plotG[x, y], {x, -dim, dim}, {y, -dim, dim}]

(*ContourPlot[plot2[x,y],{x,-dim,dim},{y,-dim,dim}]
 ContourPlot[plotAvg[x,y],{x,-dim,dim},{y,-dim,dim}]
 ContourPlot[plotG[x,y],{x,-dim,dim},{y,-dim,dim}]*)
```

```
Export["correction snr=0.2.png",
 Plot3D[plot2[x, y], {x, -dim, dim}, {y, -dim, dim}]];
Export["avgcorrection snr=0.2.png",
 Plot3D[plotAvg[x, y], {x, -dim, dim}, {y, -dim, dim}]];
Export["wrappedphase snr=0.2.png",
  Plot3D[plot[x, y], {x, -dim, dim}, {y, -dim, dim}]];

Export["CPcorrection snr=0.2.png",
 ContourPlot[plot2[x, y], {x, -dim, dim}, {y, -dim, dim}]];
Export["CPavgcorrection snr=0.2.png",
 ContourPlot[plotAvg[x, y], {x, -dim, dim}, {y, -dim, dim}]];
Export["CPwrappedphase snr=0.2.png",
  ContourPlot[plot[x, y], {x, -dim, dim}, {y, -dim, dim}]];

Export["truesig.png", Plot3D[plotG[x, y], {x, -dim, dim}, {y, -dim, dim}]];
Export["CPtruesig.png", ContourPlot[plotG[x, y], {x, -dim, dim}, {y, -dim, dim}]];
```