Three-Dimensional Microscopy by Laser Scanning and Multi-Wavelength Digital Holography

by

Alexander Khmaladze

A dissertation submitted in partial fulfillment of the requirements for the degree of
Department of Physics
College of Arts and Sciences
University of South Florida

Major Professor: Myung K. Kim, Ph.D.
Srikanth Hariharan, Ph.D.
Dennis Killinger, Ph.D.
Garrett Matthews, Ph.D.

Date of Approval:
September 12, 2008

Keywords: laser scanning microscopy, computer holography, holographic interferometry, interference microscopy, phase-contrast microscopy

© Copyright 2008, Alexander Khmaladze
Table of Contents

List of figures........................................................................................................................................ iv
Abstract................................................................................................................................................... vii
1. Introduction to optical microscopy.................................................................................................1
  1.1. Brief history of microscopy...........................................................................................................1
  1.2. Scanning versus full-field microscopy........................................................................................3
  1.3. Confocal, scanning electron and atomic force microscopy.......................................................4
    1.3.1. Confocal microscope..............................................................................................................4
    1.3.2. Atomic force microscope......................................................................................................6
    1.3.3. Scanning electron microscope..............................................................................................8
  1.4. Phase imaging microscopy..........................................................................................................10
    1.4.1. Zernike phase contrast microscopy.........................................................................................10
    1.4.2. Differential interference contrast microscopy..........................................................................11
  1.5. Digital holographic microscopy..................................................................................................14
2. Design and imaging properties of laser scanning microscope.......................................................17
  2.1. Introduction...................................................................................................................................17
  2.2. Laser scanning microscope design.............................................................................................19
  2.3. Imaging properties......................................................................................................................24
  2.4. Laser scanning with the position-sensitive detector.................................................................33
2.5. Bessel beams...................................................................................................37

3. Holography and three-dimensional imaging..........................................................39
   3.1. Introduction to holography.............................................................................39
   3.2. The retention of phase information.................................................................41
   3.3. Phase and height maps....................................................................................42

4. The principles of digital holography........................................................................45
   4.1. Angular spectrum method...............................................................................45
   4.2. Curvature correction.......................................................................................47

5. Dual-wavelength phase imaging by digital holography............................................52
   5.1. Phase unwrapping...........................................................................................52
   5.2. Multi-wavelength phase imaging....................................................................52
   5.3. Application of the phase noise reduction algorithm (fine map).....................56

6. Study of the porous coal samples...............................................................................60
   6.1. Experiment......................................................................................................60
   6.2. Two-wavelength optical phase unwrapping...................................................64
   6.3. Comparison between dual-wavelength and software phase unwrapping......66
   6.4. The application of the fine map algorithm......................................................68

7. Cell imaging...................................................................................................................70
   7.1. Experiment......................................................................................................70
   7.2. SKOV-3 ovarian cancer cells..........................................................................73

8. Conclusions....................................................................................................................78
   8.1. Laser scanning microscope.............................................................................78
   8.2. Dual-wavelength digital holography...............................................................79
List of Figures

Figure 1.1: Scanning confocal microscope .................................................................5
Figure 1.2: Atomic force microscope .........................................................................7
Figure 1.3: Scanning electron microscope ...............................................................9
Figure 1.4: Zernike phase contrast microscope .....................................................12
Figure 1.5: Differential interference contrast microscope ..................................13
Figure 1.6: Digital holographic microscope .......................................................16
Figure 2.1: Laser scanning microscope setup ...................................................22
Figure 2.2: Scanning apparatus ..............................................................................23
Figure 2.3: Scanning controls ...............................................................................25
Figure 2.4: USAF Resolution target groups 4 and 5 and zoomed 6 and 7 ..........26
Figure 2.5: The images of one cent coin ..............................................................28
Figure 2.6: Seeds of Ipomoea Violacea (Morning Glory) are imaged by PMT ..........29
Figure 2.7: Lantern fly’s eye area ...........................................................................30
Figure 2.8: Dragonfly’s eye ..................................................................................31
Figure 2.9: Ten images of the ant ........................................................................32
Figure 2.10: One cent coin imaged with a position-sensitive detector ..........34
Figure 2.11: Lantern fly’s head imaged with a position-sensitive detector ..........35
Figure 2.12: Ant’s legs imaged with a position-sensitive detector .....................36
Figure C.1. The depth of focus of the Bessel beam ...........................................................98

Figure C.2. Mathematica simulation of axicon parameters .................................................99

Figure C.3. The axicon illuminated by a Gaussian beam ..................................................100

Figure C.4. The intensity distribution in the focal plane of axicon ....................................101

Figure D.1. Dual-wavelength phase imaging digital holography ......................................103

Figure D.2. Fine map generation .....................................................................................105

Figure E.1. Main screen of dual-wavelength Labview program ....................................108

Figure E.2. Dual-wavelength phase unwrapping Labview diagram ..................................109

Figure E.3. Curvature correction Labview diagram .........................................................110
Three-Dimensional Microscopy by Laser Scanning and Multi-Wavelength Digital Holography

Alexander Khmaladze

ABSTRACT

This dissertation presents techniques of three-dimensional microscopy. First, an economical method of microscopic image formation that employs a raster-scanning laser beam focused on a sample, while non-imaging detector receives the scattered light is presented. The images produced by this method are analogous to the scanning electron microscopy with visible effects of shadowing and reflection. Compared to a conventional wide-field imaging system, the system allows for a greater flexibility, as the variety of optical detectors, such as PMT and position-sensitive quadrant photodiode can be used to acquire images. The system demonstrates a simple, low-cost method of achieving the resolution on the order of a micron. A further gain in terms of resolution and the depth of focus by using Bessel rather than Gaussian beams is discussed.

Then, a phase-imaging technique to quantitatively study the three-dimensional structure of reflective and transmissive microscopic samples is presented. The method, based on the simultaneous dual-wavelength digital holography, allows for higher axial range at which the unambiguous phase imaging can be performed. The technique is capable of nanometer axial resolution. The noise level, which increases as a result of
using two wavelengths, is then reduced to the level of a single wavelength. The method compares favorably to software unwrapping, as the technique does not produce non-existent phase steps. Curvature mismatch between the reference and object beams is numerically compensated. The 3D images of porous coal samples and SKOV-3 ovarian cancer cells are presented.
CHAPTER 1
INTRODUCTION TO OPTICAL MICROSCOPY

1.1 Brief history of microscopy

Microscopy can be defined as a technique for producing images of structures or details too small to otherwise be seen by the human eye. The device that is employed in that process is called a microscope. The microscopy field has been around for many centuries and has evolved greatly as various technological advances were applied to the development of microscopes. Two main branches of microscopy can be identified: wide field and scanning microscopy.

Wide field microscopy uses diffraction, reflection, or refraction of radiation incident upon the subject of study with the subsequent collection of this scattered radiation in order to build up an image. Scanning microscopy involves the interaction of a scanning probe with the surface or object of interest. Developments in the field of microscopy continue to play a paramount role in cell biology, medical science, the study of materials in chemistry and physics and are also an essential tool in many other areas of science and technology.

Recently, there has been an increased demand for tools that can be used for analyzing volume structures as small as a few nanometers in size. Furthermore, when investigating the properties of, for example, living cells, the samples can also be almost transparent and
very fragile, which requires additional efforts in order to effectively visualize the structure. In order to obtain high resolution, one can use electrons (instead of photons) to image the sample. However, such electron microscopes have to operate in vacuum and therefore cannot be used for in-vivo imaging. There are different scanning microscopic techniques, such as atomic force microscopy (AFM) and scanning tunneling microscopy (STM) with spatial resolution in nanometer range, but they have limitations when dealing with certain materials. Contact mode AFM tip can damage soft samples by scraping and shear forces can distort the image features. This problem may be partially solved by employing the tapping mode, but that comes at the price of the reduced scan speed and resolution. STM requires the material to be conductive, thus a complex sample preparations (e.g. gold plating) are needed before imaging biological samples.

Due to the finite focal depth of the imaging lens, a typical optical microscope can only provide a two dimensional information of an object. Moreover, out of focus light creates background, which increases noise. Optical sectioning microscopy (OSM) and scanning confocal microscopy (CSM) have been used for quantitative analysis of 3-D objects, but the mechanical scanning operation is often time consuming, thus making in-vivo imaging difficult.

Finally, for living cells, the lack of sufficient contrast makes it difficult to employ the ordinary optical microscope, as the internal structure of cells is typically colorless and almost transparent. One way to increase contrast is to stain the different structures with selective dyes, but this often involves killing cells and fixing the sample. Staining can introduce artifacts caused by the processing of the specimen and are thus not a legitimate feature of the specimen. This can be, to some degree, overcome by specific microscopy
techniques, which can non-invasively increase the contrast of the image. In general, these techniques make use of differences in the refractive index of cell structures and convert the difference in phase that light acquires while passing through the sample into amplitude (intensity) variation. The latter can be then observed by the human eye.

1.2 Scanning versus full-field microscopy

The resolution of the standard optical wide-field microscopy is diffraction limited to several hundred nanometers. Also, the technique can only effectively image strongly reflecting (or refracting) objects. In order to image features on a sub-nanometer scale, one needs to employ a scanning probe microscope.

The idea of scanning microscopy has been around since mid 20th century; however the hardware needed to do the effective scanning has largely become available only in 1970s. Scanning Probe Microscopy was really started in 1981, when Gerd Binnig and Heinrich Rohrer introduced Scanning Tunneling microscope, and later in 1986, Binnig, together with C.F. Quate and C.H. Gerber developed atomic force microscopy. These techniques gave birth to a wide variety of scanning probe microscopy instrumentations, and their applications have been increasing exponentially in diverse fields of physical sciences, engineering and technology [1].

Scanning probe microscopes allow imaging of a wide variety of material structures, such as man-made and natural systems, including biological systems, at exceedingly small scales. The family of scanning probe microscopes uses no lenses, but rather a probe that interacts with the sample surface. While these techniques are generally slower than full-field microscopy, the spatial resolution is far greater.
1.3 Confocal, scanning electron and atomic force microscopy

The type of interaction between the probe and the sample surface defines the type of scanning probe microscope being used. The method of Laser Scanning Microscopy, presented in chapter 2 is, to some degree, related to several of those techniques, which are briefly reviewed below.

1.3.1 Confocal microscope

In a laser confocal microscope, the scanning laser beam is expanded to fill the objective lens and is then focused onto a fluorescent specimen (see Figure 1.1). The mixture of reflected light and emitted fluorescent light is captured by the same objective and is focused onto a non-imaging photodetector via a beamsplitter. The reflected light is deviated by the beamsplitter, while the emitted fluorescent light passes through in the direction of the photodetector. A confocal aperture (pinhole) is placed in front of the photodetector, such that the fluorescent light from points on the specimen that are not within the focal plane (out-of-focus light) is largely obstructed by the pinhole. Thus, the confocal microscope is very efficient in terms of observing the signal from a very thin slice of the sample [2].
Figure 1.1: Scanning confocal microscope. Confocal aperture (pinhole) is used to block out-of-focus light.
Note that at any given instant, only one point of the sample is observed. The relative intensity of the fluorescent light, emitted from each point corresponds to the intensity of the resulting pixel in the image. The photodetector is attached to a computer, which builds up the image, one pixel at a time. By scanning many thin sections through the Z-axis, a very clean 3D image can be build. In order to scan the image plane effectively, various scanning mirrors and often acousto-optical modulators are used. The Z-axis scanning is usually done by a computer-controlled fine-stepping motor which moves the microscope stage up and down.

1.3.2 Atomic force microscope

The Atomic Force Microscope (AFM) works by scanning an atomistically sharp tip over a surface of a specimen (or scanning the specimen under the stationary tip). The tip is positioned at the end of a cantilever beam (see Figure 1.2). As the tip is repelled by or attracted to the surface, the cantilever beam deflects. The magnitude of the deflection is captured by a laser that reflects at an oblique angle from the very end of the cantilever. A plot of the laser deflection versus tip position on the sample surface provides the resolution of the hills and valleys that constitute the topography of the surface. Height image data obtained by the AFM is three-dimensional. The usual method for displaying the data is to color map heights on the computer screen [3].
Figure 1.2: Atomic force microscope.
The resolution of AFM images is far superior to any optical imaging methods and can, in principle be a fraction of the nanometer. However, either for the duration of the entire scan, if imaging in contact mode, or for a short period of time, if the changes in resonant frequency are measured in tapping mode, the sample is needed to be in contact with the mechanical tip, which often makes imaging of soft samples difficult.

1.3.3 Scanning electron microscope

Electron microscopes function similarly to optical microscopes, but instead of light illuminating the sample, they use a sharply focused beam of electrons to scan across the surface of the sample (see Figure 1.3).

Scanning Electron Microscope (SEM) provides topographical and elemental information at magnifications of up to 100,000x, which translates to lateral resolution of several nanometers with virtually unlimited depth of field [4].

The electrons interact with the sample and generate secondary (Auger) electrons, backscattered electrons, and characteristic X-rays. All of them can be analyzed to gain information about the sample structure and composition.

A stream of electrons is formed by the electron source and accelerated toward the specimen. The stream is condensed by the first condenser magnetic lens, which is used in conjunction with the condenser aperture to eliminate the high-angle electrons from the beam. The second condenser lens forms the electrons into a thin, tight, coherent beam. The final lens, the objective, focuses the scanning beam onto the spot on the specimen. When the beam strikes the sample and dwells on a particular spot for a few microseconds, interactions occur inside the sample.
Figure 1.3: Scanning electron microscope.
These interactions are detected with various instruments, which count the number of interactions and record a pixel, whose intensity is determined by the number of interaction (i.e. more reactions result in brighter pixel). The entire pattern can be scanned 30 times per second.

The clear downside of SEM imaging is the fact that in order for it to work, the sample has to be conductive. As a result, various specimens, including many biological materials are unsuitable for SEM imaging.

1.4 Phase imaging microscopy

Phase imaging techniques are used to convert phase variations that light wave acquired while passing through objects into amplitude variations that can be observed. Over the years, there have been a number of techniques developed to qualitatively perform this conversion. The examples include phase contrast (ZPC) microscopy and differential interference contrast (DIC) microscopy.

1.4.1 Zernike phase contrast microscopy

Phase contrast microscopy, first proposed in 1934 by Dutch physicist Frits Zernike, is a phase contrast-enhancing optical technique that can be utilized to produce high-contrast images of transparent specimens, such as live cells, medical tissues slices, microorganisms, fibers, etc. [5, 6].
Presented in Figure 1.4 is a diagram of a modern upright phase contrast microscope together with the schematic illustration of the phase contrast optical path. Partially coherent illumination, produced by the tungsten-halogen lamp is directed through a collector lens and focused on a condenser annulus. Wave fronts passing through the annulus illuminate the specimen and either pass through unperturbed or are diffracted and retarded in phase by structures and phase gradients present in the specimen. Both non-diffracted and diffracted light waves are collected by the objective and segregated at the rear focal plane by a phase plate and focused at the intermediate image plane to form the final phase contrast image observed in the eyepieces.

1.4.2 Differential interference contrast microscopy

Differential interference contrast microscopy (DIC), also known as Nomarski Interference Contrast (NIC) uses interferometry of two polarized light beams, which take slightly different paths through the sample [7]. The length of each optical path differs, the beams interfere, when they are recombined, which gives the appearance of a three-dimensional image.

The light is polarized and splits into two beams by a Wollaston prism (see Figure 1.5). These two beams are focused by the condenser so they will pass through two adjacent points in the sample, fraction of a micron apart. This results in the phase difference, which is then converted into amplitude variation by recombining the two beams in the second Wollaston prism and another polarizer (analyzer) and thus making them interfere. The interference pattern then contains the information about the phase variation in the sample.
**Figure 1.4**: Zernike phase contrast microscope.
Figure 1.5 Differential interference contrast microscope.
1.5 Digital holographic microscopy.

In ZPC and DIC phase contrast microscopy, the phase to amplitude conversion is nonlinear. Therefore, these methods cannot be used to quantify the phase change. The importance of quantitative phase information is in the fact that the phase change indicates the change in the optical path length the light has travelled. The optical path length can be then converted to physical thickness, providing the sample’s height information. Thus, quantitative phase imaging is a 3D imaging technique.

There has been a number of quantitative phase imaging techniques proposed in the recent years. Barone-Nugent et al. [8] have demonstrated a quantitative phase imaging microscope that separates phase information from amplitude information and produces pure phase images. Several phase shift interferometry techniques have been used for quantitative phase imaging [9]. Another technique is diffraction phase and fluorescence (DPF) microscopy [10], which uses simultaneous quantitative phase imaging and epi-fluorescence imaging of living cells.

Recently, due to the advances in computer technologies as well as the availability of high-resolution CCD-cameras, digital holographic microscopy (DHM) has emerged as a powerful tool to obtain quantitative phase information and provide information about 3D structure of microscopic samples. In digital holography (as in the case of conventional holography) the process of recording is done optically, but the recording media is usually a CCD-array, which allows for easy digitization and storage of the holographic recording in the computer memory. This recording then contains the information of not just the intensity of light (amplitude signal), but also its phase.
After the phase and amplitude information is recorded, the process of extracting it can be accomplished by numerically diffracting and propagating the reference wave through the holographic recording. The resulting complex wave field produces the amplitude and phase maps simply as the amplitude and phase of a set of complex numbers. The process of the holographic reconstruction is, therefore, essentially reduces to a numerical diffraction problem, which is done entirely by a computer (see Figure 1.6).

One of the main advantages of such a technique is that image processing algorithms can be easily applied on various stages of the reconstruction. Here, we use the angular spectrum algorithm, which provides a number of significant advantages in filtering in Fourier domain, software curvature correction and numerical focusing.
Figure 1.6: Digital holographic microscope.
2.1 Introduction

For years, the conversion of images produced by a microscope into digital form posed a number of challenges. CCD arrays have more than a few disadvantages in terms of resolution and image quality. Even on the higher-end systems, the increase of the number of pixels on CCD matrix leads to the decrease in sensitivity, requiring longer exposure time and brighter illumination. In addition to that, the optical system of the microscope itself can give rise to aberrations, further reducing the image quality.

Various scanning microscopy systems have been developed to offer a number of significant advantages to the conventional wide-field microscope imaging systems. The resolution of the scanning microscopes is not limited by diffraction of imaging optics, but only by the size of the probe. These techniques gave origin to a wide variety of scanning probe microscopy instrumentations. In laser scanning confocal microscopy, a laser beam is projected onto a point on a sample, which is then imaged and apertured at the detector. The image is then formed by illumination of the sample point by point with the subsequent reconstruction by a computer. This principle is quite versatile and allows for a large array of variations of imaging techniques. For example, scanning tunneling microscopy (STM), which has spatial resolution in nanometer range but also have
limitations when dealing with certain materials. STM requires the material to be conductive, thus a complex sample preparations (e.g. gold plating) are needed before imaging biological samples. Scanning electron microscopy (SEM) is yet another well-known technique, which offers high resolution and striking contrast, but requires complex sample preparation.

Here we present a development in a laser scanning microscopy technique [11] - a method of imaging that is analogous to SEM, but performed entirely by means of optical scanning. Thus, the method neither requires an imaging array detector nor needs complicated sample preparation. It uses a laser beam, which is focused by a system of lenses into a single spot on a sample, while a large-numerical-aperture non-imaging detector receives the scattered light. The technique is similar to a scanned beam endoscope [12], where imaging is also conducted by scanning the laser spot over the area of the sample, while measuring the scattered energy for each point. Here, the laser scanning microscope was used together with the position-sensitive detector to study the orientation of the object surface.

Because all the illumination energy falls on the particular spot of interest, there is no need to form a conjugate image plane. Moreover, there is no need to exclude the light from elsewhere in the field of view with a lens or aperture. Thus, the system does not use a pinhole (typical for a laser scanning confocal microscopic system). Instead, the scanning laser beam parameters are manipulated freely by the system of lenses to produce a beam profile of a required cross-section and confocal beam parameter. It is worth noting that the lenses used in the system are simple glass lenses (not microscope objectives, as they are not used to form an image, but only to focus the beam) which
greatly reduce the system’s cost. Even with these simple lenses the system remains free from all non-longitudinal aberrations such as coma, astigmatism, field curvature, and distortion.

The lateral resolution, as well as the depth of focus of this system is effectively determined by the laser beam spot size. If the visible light is used, the lateral resolution can be on the order of a micron, while depth of focus (axial resolution) is determined by the laser beam profile. The non-invasive nature of this method, as well as the ease of operation and low cost far outweighs the shortcomings of lower than SEM resolution for a variety of scientific and industrial applications.

### 2.2. Laser scanning microscope design

In order to build the laser scanning microscope with the position-sensitive detector, one first needs to consider the parameters of the laser beam, such as its intensity, divergence and spot size required to generate the images of the desired axial and lateral resolution. In our system, laser (assumed to be emitting TEM$_{00}$ Gaussian mode) spot size was measured by gradually blocking the laser beam with knife edge, while recording the change in the overall beam intensity and numerically reconstructing the beam intensity profile (see Appendix A). The size of the laser spot on the sample can be determined theoretically, by using the complex beam parameter $q(z)$ [13]. The location of the beam's waist $W_0$ and its size were determined by measuring the beam's spot in several locations and effectively reconstructing the intensity profile of the Gaussian beam. The initial laser beam waist was found to be equal to 360 μm and was located near the output mirror of the laser. At the waist, the inverse to the complex beam parameter was found to be:
\[ \frac{1}{q(z)} = \frac{1}{R(z)} - i \frac{\lambda}{\pi W(z)^2} = -i \cdot 1.55 m^{-1}, \quad (2.1) \]

where \( \lambda = 0.63 \mu m \) is the laser wavelength, \( R(z) \) is the radius of curvature and \( W(z) \) is the beam width as functions of axial coordinate \( z \). At the waist \( R(z) \) is infinite and \( W(z) = W_0 \).

In the case of Gaussian beam, its depth of focus is determined as twice the Rayleigh range \( (2z_0 = \frac{2\pi W_0^2}{\lambda} = 1.29 \mu m) \), which shows that in order to achieve higher lateral resolution and longer focal depth, a compromise is necessary, as lowering the spot size on the sample will lead to the lowering of the focal depth.

In the paraxial approximation any optical system is completely characterized by its ray transfer ABCD matrix. We used this approach for propagation of the Gaussian beam and the determination of the lateral range of the system. For the system of three focusing lenses, presented here (focal lengths of the lenses L1, L2 and L3 are 10 cm, 100 cm and 5 cm respectively), we constructed its ray transfer matrix and then evaluated it by a computer. The output complex beam parameter \( q_{out} \) was found using the matrix elements and the parameter \( q_{in} \) of the input beam:

\[ q_{out} = \frac{A q_{in} + B}{C q_{in} + D} = -3.56 \cdot 10^{-6} + i \cdot 3.83 \cdot 10^{-5}. \quad (2.2) \]

The final beam spot size was then computed using Equation (2.1) and was equal to 2.7 \( \mu m \) with the depth of focus of 76 \( \mu m \) (see Appendix A).
Figure 2.1 shows the schematics of the apparatus. The 15mW He-Ne laser beam is steered by mirrors and expanded by lenses L1 and L2, which together created a telescopic system to produce a beam with larger cross-section in front of lens L3. Lens L3 can then focus the beam into a small spot on an object. The object is mounted on a translation stage driven by a motorized micrometer, and is scanned along the vertical axis (see Figure 2.2). The fast horizontal scanning is achieved by means of galvanometer based scanning mirror. With the typical frequency of fast scanning of 100 Hz, the entire image of a hundred lines can be acquired in a second. It is worth noting that the system can easily be used with two scanning mirrors, in which case the scan speed can be increased to many frames per second, but at a higher cost.

Equation

\[ \Delta x = \tan[\theta] \cdot f, \quad (2.3) \]

where \( f \) is the focal length of the focusing lens L3 and \( \theta \) is the mechanical angle of mirror rotation, is used to determine the actual horizontal scanning range \( \Delta x \) due to the mirror. Because both the sample and the scanning mirror are at the focal points of the lens L3 there is no need for geometric correction as the beam’s angle of incidence on the sample is always zero degrees.
Figure 2.1: Laser scanning microscope setup. Neutral density (ND) filters were used to avoid damage to fragile specimen. Focal length of lenses L1, L2, L3 and L4 are 10 cm, 100 cm, 5 cm and 10 cm respectively. The focusing lens L3 can be an ordinary spherical lens or an axicon.
Figure 2.2: Scanning apparatus
The light, scattered from the sample, is partially collected by lens L4 and the intensity information is obtained by the photodetector PD. The system was first used with a photodiode and a photomultiplier tube (PMT). Finally, a quadrant position-sensitive detector was employed. A 12-bit data acquisition board feeds the detected signal into a computer, where a set of LabVIEW programs (See Appendix B) control the scanning and generation of the two dimensional image of the object (see Figure 2.3).

2.3 Imaging properties

Some images produced by our laser scanning system are shown below. First, in order to assess the system’s resolution, a USAF resolution target is imaged (Figure 2.4) using a simple photodiode. The image is 1176 x 1296 pixels and the pixel size is 0.9 x 1 μm.

The measured beam spot size at the target was around 5 μm. Note that since the laser beam intensity profile is Gaussian, by making the pixel (step) size smaller than a spot size (over-sampling) the actual resolution is even slightly higher than spot size.

The lateral resolution can be further improved by using a system with the higher ratio of the foci of lenses L2 to L1. There is still a practical limit, however, after which the resolution does not improve, due to the diffraction limit of the finite apertures. Also, due to the quadratic dependence between the beam waist size and the Rayleigh range, the system with smaller spot size will have much lower depth of focus.
Figure 2.3: Scanning controls
Figure 2.4: USAF Resolution target groups 4 and 5 and zoomed image of groups 6 and 7. Group 7 element 2 (line thickness less than 3.5 μm) is resolved.
Figure 2.5 shows an area of a penny. The images are 500x500 pixels and the pixel size is 2.79x3 μm. The images exhibit the perception of the 3D relief structure of the letters, evident as highlighted and shadowed regions. Note that since the letters on the penny protrude from background, the image appears as if the object is illuminated by a light source on the left with a camera viewing it from the front. In reality, the laser beam illumination is from the front while a point detector is located on the left. Note also that the grainy patterns of the background are due to the actual texture of the surfaces. Although the light source is a coherent laser, any coherent speckle effect is averaged out by the large aperture of the detector, thus removing one of the common sources of noise in laser-based imaging systems. Figure 2.6 shows the image of a flower seed. The sample is of low reflectivity, which results in the signal level too low to be collected by an ordinary photodiode, so PMT was used instead.

Note the reversed shadow effect as described for Figure 2.5. To the first order, these images are equivalent to ones that can be obtained using conventional wide-field microscopy, but with the scanning system the optics is greatly simplified.

Some images of insects are shown in Figure 2.7, Figure 2.8 and Figure 2.9. A lantern fly’s head and eye area is visible in Figure 2.7, where the individual ommatidia are resolved. Similarly, Figure 2.8 shows the eye of a dragonfly. The images, shown in Figure 2.9, show the body of the ant for the various axial positions of the sample. For each consecutive image, the axial position of the sample was changed by 250 μm. When looking at the individual features, one can estimate the depth of field in this case to be about 1.5 mm.
Figure 2.5: The images of one cent coin (parts of the inscription "E-PLURIBUS UNUM" are shown). The image frames are 1.4 x 1.5 mm² (500 x 500 pixels).
Figure 2.6: Seeds of Ipomoea Violacea (Morning Glory) are imaged by PMT. The image frame is 1.8 x 1.2 mm$^2$ (434 x 289 pixels).
Figure 2.7: Lantern fly’s eye area (500 x 793 pixels)
Figure 2.8: Dragonfly’s eye (500 x 1000 pixels). The image frame is 2.1 x 3.3 mm²
Figure 2.9: Ten images of the ant spanning the axial range. The image frame is 2.1 x 4.2 mm$^2$ (500 x 1000 pixels). The individual features (see the grey square) on the first image of the series are still visible on the seventh image.
2.4 Laser scanning with the position-sensitive detector

The detector was then changed to the position-sensitive (quadrant) detector. The detector consists of 4 areas (A, B, C and D diodes) positioned in 4 quadrants each capable of detecting the light intensity separately from the other. The detector output contains 3 channels: the Sum intensity of all 4 quadrants, Top-minus-Bottom (T-B) and Left-minus-Right (L-R). If the quadrant detector is initially positioned in such a way that the scattered light collected by the lens L4 in Figure 2.1 is in the middle of the detector, T-B and L-R channels are approximately zero, while the Sum channel registers the same signal as in the case of an ordinary photodiode.

As the laser beam was scanned over the features of the sample, the position of the intensity maximum of the scattered light moved up/down and sideways to reflect the spatial orientation of the scanned surface. The T-B and/or L-R channels were monitored and the sign and the magnitude of the signal were then indicative of not only the reflective properties of the sample, but of the way the surface is oriented in a particular spot.

The image in Figure 2.10 is of a one-cent coin. The reflectivity of the surface was approximately the same everywhere on the coin, but the scattering in the direction of the detector is higher when the surface is turned towards the object and lower when the surface is turned away, thus showing the 3D structure of an individual column. Similarly, the images of a lantern fly’s head and ant’s legs, where the individual 3D features are highlighted, are shown in Figures 2.11 and 2.12.
Figure 2.10: One cent coin imaged with a position-sensitive detector: a column of Lincoln’s memorial. 0.56 x 1.5 mm$^2$ (200 x 500 pixels).
Figure 2.11: Lantern fly’s head imaged with a position-sensitive detector. The image frame is 1.4 x 3.0 mm$^2$ (500 x 1000 pixels)
Figure 2.12: Ant’s legs imaged with a position-sensitive detector. The image frame is 1.95 x 2.4 mm² (700 x 800 pixels).
2.5 Bessel beams

In conventional optics, the lateral resolution and the depth of focus are conflicting requirements and cannot be both maximized at the same time. A possible solution is the use of a Bessel beam generated by an axicon lens [14]. On-axis illumination of the axicon by the Gaussian beam produces a Bessel beam - an interference pattern, which results in energy redistribution between the central maximum and the side lobes (see Appendix C).

The use of Bessel beams has been already demonstrated in OCT applications [15], for example. In the case of the ideal Bessel beam, the size of a central maximum remains the same as the beam propagates through space [16, 17]. If this beam is being used in a laser scanning microscope, such system will have infinitely long depth of focus. Even for a real Bessel beam the depth of focus is still much longer than a Gaussian beam [18].

Mathematically, the influence of a thin axicon on the transmitted Gaussian beam can be taken into account by the radial phase factor $\exp\left[-\frac{2\pi i (n_a - 1) \alpha}{\lambda}\right]$, where $\lambda$ is the wavelength of laser light, $n_a$ is the refraction index, $\alpha$ is the wedge angle of the axicon (the wedge angle is equal to (180 degrees - cone angle)/2) [19]. The Gaussian beam of amplitude $A(r, \varphi)$ transmitted through the axicon can be written as

$$A(r, \varphi)\exp\left[-\frac{2\pi i (n_a - 1) \alpha}{\lambda}\right]. \quad (2.4)$$

Physically, the focusing of the beam means the phasing in of its spatial components. To model the propagation of the beam in space, one can obtain its spatial spectrum and use the Fresnel integral. The resulting spatial spectrum can be converted back to the amplitude form via the inverse Fourier transform. It is possible to obtain the approximate solution (Equation 26 in [19]) using the asymptotic approximation of the Bessel function.
The beam intensity is then obtained as the squared amplitude, and can be used to estimate the size of the Bessel beam spot. The calculations, based on this approximation, for the incident beam spot diameter (Gaussian TEM\(_{00}\) mode) of \(W_0 = 8\text{mm}\), index of refraction \(n_a = 1.5\), wedge angle \(\alpha = 10^\circ\) would produce an axicon focal depth:

\[
L = \frac{W_0}{(n_a - 1)\alpha} = 9.2\text{cm}
\]  

with the beam spot size at the object of about 3 \(\mu\text{m}\). The significant down side of the use of axicon in this system is the presence of the side lobes, which draw a significant amount of energy, thus reducing the overall contrast of the system. However, using different wedge angles may result in an acceptable tradeoff between lower contrast and higher depth of focus.
3.1 Introduction to holography

Holography (from Greek “holos” meaning “the whole”), a method of recording both intensity and phase information of a complex wave-field, was first proposed by Dennis Gabor in 1948 [20]. However, it was only in the 1960s, when the construction of laser and the introduction of off-axis technique, pioneered by Leith and Uptaniesks [21], made it practical. When an object is illuminated with a light wave, the reflected (or transmitted) wave’s amplitude and phase are perturbed as a result of the wave interaction with the object. In case of ordinary photography, only the amplitude information is retained. If the object light wave is sufficiently coherent, it can be made to interfere with the reference wave.

The resulting interference pattern then contains the information about both amplitude and phase. This process is commonly referred to as holographic recording, which can be done on a photographic plate or an electronic device, such as CCD.

In order to reconstruct the hologram, one needs to illuminate it with the reference wave. When the reference wave illuminates the holographic recording, it diffracts from it. The diffracted wave still contains the information about both amplitude and phase of the original object wave, and the resulting diffraction pattern produces a 3D image of the
original object, the hologram. Physically, it is the presence of phase information that
gives the depth perception for a human observer. It is also then possible to use this
information to actually measure the depth (physical thickness) of the original object,
which makes holography an important tool in scientific and engineering research.

If the reference and object beams are at an angle to each other, the resulting
holographic recording still retains all the properties, described above, but the
reconstruction then will produce spatially separated zeroth and first orders of diffraction
(known as DC, real and mirror images). In this case, the real holographic image is not
obscured by the DC background.

Holographic interferometry, designed as a non-contact method for studies of
defection, strains, vibrations and heating was proposed by Stetson et al [22]. It made it
possible to non-invasively profile surfaces with an accuracy of less than a micron.

Also in the late 1960’s, the origin of first computer generated hologram can be traced
to Goodman and Lawrence [23] and Kronrod et al. [24]. These methods involve either
numerical generation of the holographic recording with the subsequent reconstruction, or
the optical recording of the hologram with the reconstruction done digitally by a
computer.

The conventional process of holographic recording on photographic plates is rather
complicated and time-consuming, which makes real-time imaging difficult. In the past
decade, the emphasis has been shifting to digital holography [25]. In this case, the
hologram is recorded by a high resolution CCD array [26-28]. As in the case of the
ordinary holography, the hologram contains the information of not just the amplitude
distribution of light, but also of its phase. After the hologram recording, the extraction of
the amplitude and phase can be accomplished by numerically propagating the reference wave. The complete and accurate description of the propagation of the optical field by the diffraction theory allows numerical reconstruction of an image as an array of complex numbers, which represent the amplitude and phase of the optical field [29]. In addition to the ability of fast image acquisition and the retrieval of both quantitative amplitude and phase information, digital holography offers the versatility of various image processing techniques that can be applied to the complex field, which may not be feasible in real space holography. A number of different methods have been considered for numerical reconstruction including Fresnel transform, Huygens convolution, and angular spectrum [30-32].

Application of digital holography in microscopy is especially important, because of the extremely narrow depth of focus of high-magnification systems [33, 34]. Microscopic imaging by digital holography has been applied to image microstructures and biological systems [35-37]. Numerical focusing of holographic images can be accomplished from a single exposed hologram. Direct accessibility to the phase information can be used for numerical correction of different aberrations of the optical system, such as field curvature and anamorphism [38].

3.2 The retention of phase information

Numerically, if a light wave passes through a specimen of thickness \( t \), the phase shift \( \Delta \phi \) changes due to variation of refractive index. This can be expressed as

\[
\Delta \phi = \left( \frac{2\pi}{\lambda} \right) t \Delta n
\]  

(3.1),
where $\Delta n$ is the change in refractive index of the specimen relative to the surrounding medium.

Phase-contrast techniques convert the phase changes suffered by the light wave, while passing through or reflecting from objects, into observable intensity variations. Over the years, a number of techniques have been developed to qualitatively perform this conversion. As it was mentioned before in chapter 1, ZPC and DIC phase contrast microscopy techniques, while giving an appearance of 3D imaging, cannot be easily used to quantify the phase change, since the conversion of phase to intensity modulation is nonlinear. Since the phase change indicates the change in the optical path length, it can be then converted to physical thickness, providing the sample’s height information. Thus, the direct access to the quantitative phase information makes digital holographic microscopy a true 3D imaging technique.

3.3 Phase and height maps

If the light wave reflects from an object, its surface is described by a height map $h(x,y)$, which is determined from the phase map $\phi(x,y)$ of the holographic reconstruction at a given wavelength by

$$h(x,y) = \frac{\lambda}{4\pi} \phi(x,y)$$  (3.2)

(there is a factor of $\frac{1}{2}$ due to the fact that light travels to the surface and then reflects back).

Figure 3.1a shows the phase map of the aluminum-covered USAF resolution target. The step size is approximately 2.2 radians. According to equation 3.2, it is proportional to the target height map, whose profile corresponding to the line over the phase map is
sketched. The step height is of about 100nm. This result is confirmed by the AFM scan shown in Figure 3.1b.

On the other hand, if the object is a mostly transparent cell on the reflective substrate, so that the light propagates through it, reflects from the substrate and propagates back, the physical thickness is

\[ h(x, y) = \frac{\lambda}{4\pi} \frac{\phi(x, y)}{(n-n_0)} \]  

(3.3),

where \((n-n_0)\) is the refractive index difference between the cell and air.
Figure 3.1: Phase map and height profile for $\lambda$=633 nm: (a) the profile is taken along the line over the phase map and (b) AFM image and height profile that confirm the results in (a).
4.1 Angular spectrum method

Once a hologram has been acquired, it is reconstructed by numerically propagating the optical field along the direction perpendicular to the hologram plane (z-direction) in accordance with the laws of diffraction. 

Fresnel-Kirchoff formula can be expressed as Fourier integral [39]:

\[
A_0(k_x, k_y; 0) = \iint E_0(x, y; 0) \exp[-i(k_x x + k_y y)] dxdy
\]

where \(k_x\) and \(k_y\) are spatial frequencies corresponding to \(x\) and \(y\) respectively. 

\(E_0(x, y; z = 0)\) is the intensity distribution recorded by the CCD camera. This is the expression for Fourier transform and \(A_0(k_x, k_y; 0)\) is the angular spectrum of the optical field \(E_0(x, y; z = 0)\) at the hologram plane \(z = 0\). The object’s angular spectrum consists of a zero-order and a pair of first-order terms. One of the first-order terms is the angular spectrum of the object field and the other is its phase inverted version. Figure 4.1(a) shows the hologram of a USAF resolution target recorded by our dual wavelength experimental setup. The two crossing interference fringe patterns, formed by two wavelengths, can be clearly seen. Figure 4.1(b) presents the Fourier spectrum with the two pairs of first-order components, corresponding to the two wavelengths, plainly visible.
Figure 4.1: Two-wavelength hologram of the USAF resolution target: (a) digital hologram (640x480 pixels) and (b) its Fourier spectrum of the hologram with the red and the green wavelengths first order components shown.
The field $E_0(x, y; z = 0)$ can be regarded as a projection of many plane waves propagating in different directions in space and with the complex amplitude of each component equal to $A_0(k_x, k_y, 0)$. The angular spectrum can then be propagated in space along the $z$-axis:

$$A(k_x, k_y; z) = A_0(k_x, k_y; 0) \exp[ik_z z],$$

(4.2)

where $\exp[ik_z z]$ is the complex transfer function and $k_z = \sqrt{k^2 - k_x^2 - k_y^2}$, where $k = 2\pi / \lambda$. Here, there is no requirement for $z$ to be larger than a certain minimum value, as in the case of Fresnel transform or Huygens convolution. The complex wave-field at an arbitrary $z$ can be obtained by performing the inverse Fourier transform:

$$E(x, y; z) = \iint A(k_x, k_y; z) \exp[i(k_x x + k_y y)] dk_x dk_y$$

(4.3).

As both integrals in Equations (4.1) and (4.3) are computed via FFT algorithm, the angular spectrum method is well suited for the real-time imaging.

### 4.2 Curvature correction

The angular spectrum method described above is based on the premise that the reference and object waves are both plane waves. However, in the real setup, each wave has its wavefront curvature, resulting in a curvature mismatch. Consider the complex field captured by a CCD camera (see Figure 4.2).

The phase mismatch can be compensated numerically, by multiplying the original “flat” field $E_0(x, y; z = 0)$ by the phase factor $\exp[i\phi]$, where $\phi = kd$ is the phase difference between $A$ and $O$. Here, $k = 2\pi / \lambda$, where $\lambda$ is the wavelength of light and $d$ is the optical path difference:
**Figure 4.2:** Curvature correction. R is the wave’s radius of curvature centered at C, which can be determined experimentally for a given setup, $\vec{r}$ is the vector from the center of the CCD matrix (point O) to an arbitrary point A, and $\vec{r}_0$ is the vector from the center of the CCD matrix to the projection of the center of curvature on the CCD matrix P. Here $|\vec{r}| = \sqrt{x^2 + y^2}$, x and y are the coordinates of A and $|\vec{r}_0| = \sqrt{x_0^2 + y_0^2}$, $x_0$ and $y_0$ are the coordinates of P.
\[ d = CA - CO = \sqrt{CP^2 + PA^2} - \sqrt{CP^2 + PO^2} \] (4.4).

From geometry:

\[ d = \sqrt{R^2 + (r - r_0)^2} - \sqrt{R^2 + r_0^2} = \sqrt{R^2 + (x - x_0)^2 + (y - y_0)^2} - \sqrt{R^2 + x_0^2 + y_0^2} \] (4.5).

The difference can be positive or negative, depending on the angle of the curvature we are compensating. Finally,

\[ E(x, y; 0) = E_0(x, y; 0) \exp[ik(\sqrt{R^2 + (x - x_0)^2 + (y - y_0)^2} - \sqrt{R^2 + x_0^2 + y_0^2})] \] (4.6),

which is the exact expression for the curvature-corrected field. This expression agrees with the approximation from reference [38], in the case \( R > r \) and \( r_0 \to 0 \):

\[ k(\sqrt{R^2 + r^2} - R) = kr \left[ \frac{1 + r^2 / R^2 - 1}{1 + r^2 / R^2} \right] = 2\pi R \frac{x^2 + y^2}{\lambda} \] (4.7).

It is worth noting that Equation (4.7) is a known expression for Newton’s rings, which means that if the object is a plane mirror, the resulting interference pattern would be a set of concentric rings with the dark fringes of radius of \( \sqrt{mR\lambda} \), where \( m = 0, 1, 2, \ldots \). Therefore, for a wavelength of 532 nm, \( R = 3 \) cm, the radius of a first fringe is 126 micron and there is a total of 3 fringes visible in 174 micron frame (see Figure 4.3). If the field of view is increased, there are going to be more fringes visible and at some point the aliasing may occur. One can use this formula as an analytical expression to avoid fringe aliasing. For example, for the parameters above, in order for the fringes to alias (less than 2 pixel per fringe), one would have to have a field of view large enough for over 100 fringes.
If the parameters are chosen correctly, even a substantial curvature mismatch can be compensated. Figure 4.3 shows the phase image of the USAF resolution target covered with a layer of aluminum to make it entirely reflective. The pattern on the resolution target is elevated approximately 100 nm above the flat background. Figure 4.3(a) shows the reconstructed image before the curvature correction. Figure 4.3(b) is the same image after the curvature correction was applied, and the curvature mismatch completely compensated.
Figure 4.3: The reconstructed phase image of the USAF resolution target: (a) without curvature correction and (b) with curvature correction applied. The images are 174x174 μm² (450x450 pixels).
CHAPTER 5

DUAL-WAVELENGTH PHASE IMAGING BY DIGITAL HOLOGRAPHY

5.1 Phase unwrapping.

Phase images of objects with variations in optical thickness greater than the wavelength are ambiguous and results in phase wrapping. Consequently, the phase map exhibits discontinuities at the positions where the total phase change exceeds $2\pi$.

The phase $\Delta \phi$ in equations 3.2 and 3.3 can only vary from 0 to $2\pi$, which corresponds to optical thickness variation of 0 to $\lambda/2$. The phase imaging of objects with the optical thickness variation higher than that is ambiguous and gives rise to $2\pi$-discontinuities in the phase image.

There are various phase unwrapping method available, which involve dividing the phase image into horizontal lines and unwrapping them by scanning pixels and adding a $2\pi$ offset to each pixel, when needed. After that, the unwrapping process is done along vertical lines. Software algorithms that exist for detecting and removing $2\pi$ discontinuities often require user intervention, computationally slow and habitually produce erroneous results when the phase profile is noisy.

5.2 Multi-wavelength phase imaging

We have introduced a multiple-wavelength phase-imaging technique that removes the $2\pi$-discontinuities [40]. Unlike software algorithmic approaches to phase unwrapping, it
does not require user intervention and has only minimal requirement on the level of phase noise and discontinuity. Furthermore, the method allows imaging to be performed faster – the only time constraint being the speed at which the Fourier transforms in angular spectrum method is performed.

Suppose we image an object such as a mirror, which is positioned at an angle to the optical axis of the setup. If the angle is different from 90 degrees, the surface of the mirror will have a slope. When it is imaged by using a wavelength smaller than its overall height, the phase image will contain $2\pi$ discontinuities, as shown in Figure 5.1a and Figure 5.1b.

However, if the simultaneous dual-wavelength phase imaging is performed, the discontinuities of the two maps will occur at different positions, since the two wavelengths are different. It allows unwrapping the phase by comparing the two maps, as stated below. In this way, the $2\pi$ jumps are removed and then the phase ambiguities are resolved on a bigger range, which corresponds to a new synthetic “beat” wavelength (see Figure 5.1c).

Figure 5.2 shows the phase images of the USAF resolution target imaged at an angle. The images produced with single wavelengths exhibit multiple phase steps (Figure 5.2a and Figure 5.2b). The phase maps $\phi_1$ and $\phi_2$, derived from each wavelength are subtracted, so that $\phi_{21} = \phi_1 - \phi_2$ is obtained. Adding $2\pi$ wherever $\phi_{21} < 0$ yields a new phase map, practically free of discontinuities. It is equivalent to a phase map created by a single synthetic “beat” wavelength:

$$\Lambda_{12} = \frac{\lambda_1 \lambda_2}{|\lambda_1 - \lambda_2|}$$  

(5.1)
Figure 5.1: Phase maps resulting from imaging 5 micron sloped surface by $\lambda_2=532$ nm (a) and $\lambda_1=633$ nm (b), where multiple discontinuities are clearly visible and the resulting beat wavelength phase map (c) with the resulting extended range.
Figure 5.2: Phase maps of the resolution target for (a) $\lambda_1=532$ nm and (b) $\lambda_2=633$ nm. (c) Synthetic dual-phase map with beat wavelength $\Lambda_{12} = 3334$ nm and (d) its 3D rendering (the images are 174x174 μm² and the vertical scale for (a-c) is in radians).
For the wavelengths $\lambda_1=633$ nm and $\lambda_2=532$ nm, the beat wavelength is $\Lambda_{12} = 3334$ nm (Figures 5.2c and Figure 5.2d). Here, while the phase images produced by a single wavelength exhibit multiple discontinuities, in the final synthetic wavelength map, the discontinuities are removed. In fact, here the synthetic wavelength is such that the range of the dual-wavelength phase map is barely enough to resolve the discontinuities (some even remain on the left and right of Figure 5.2c).

5.3 Application of the phase noise reduction algorithm (fine map)

The drawback of the dual-wavelength method is the amplification of phase noise by the same factor as the range. Furthermore, the two phase maps differ in their noise distributions, so that the final dual-wavelength phase map can remain quite noisy even if the noise in the single-wavelength phase maps is low. However, one can use this dual-wavelength “coarse” map as a guide, together with one of the original phase maps ($\phi_1$ or $\phi_2$), to produce the low noise “fine” phase map. The method (detailed in the Appendix D and reference [40]) involves dividing the height of the coarse map into the integer number of one of the original wavelengths, say $\lambda_1$. Then, the wavelength high segments from the phase map $\phi_1$ are pasted into this coarse map, which achieves the desired effect of the reduced noise together with the extended range. In practice, the areas near the boundaries of the wavelength intervals are somewhat problematic. There, the noise present in a single wavelength map $\phi_1$ causes the height to change erratically by one wavelength. In order to partially solve this problem, one can compare this map to the coarse map and, if the difference is more then $\lambda_1/2$, add or subtract $\lambda_1$ depending on the size of the difference.
However, if the noise is too excessive the last step results in the shift of small portions of the final image by $\lambda_1$, from its true position. Since the height of such a shift is always $\lambda_1$, it can be fixed by software, by looking for the steps of this height and shifting them up (or down) by $\lambda_1$.

Indeed, the high noise level makes it very difficult to measure the height of the object’s individual features from the dual-wavelength phase map, as shown by the height profile in Figure 5.3 (a) obtained from the map in Figure 5.2 (c), despite the fact that the overall shape of the object is still preserved by the dual phase map. In contrast, the height profile of the “fine” map, shown in Figure 5.3 (c), again yields the phase image, where the step height of 100 nm is clearly observable (see chapter 3).

In order to numerically estimate the noise levels in the system, the height profiles of the flat area for a single wavelength, a coarse map, and a fine map (Figure 5.4) were taken and their rms noise were measured. While the rms noise in the coarse map is substantial (of order 54 nm), the noise for the fine map is almost equal to the single wavelength phase map (of order 8.5 nm and 6.5 nm respectively).
Figure 5.3: Height profiles of the resolution target of (a) “coarse” and (b) “fine” phase maps. (c) final “fine” map and (d) 3D rendering of (c). The image sizes are 174x174 μm².
**Figure 5.4:** Line intensity profiles of a flat area for the coarse, fine, and the single wavelength phase maps respectively.
6.1 Experiment

Activated coals, as well as coals treated with pyrolysis, are highly porous materials. This is the reason why these coals have a high capacity for absorption, which makes them very important in processes such as purification and filtering. These processes often depend on the size and morphology of the pores. Normally, porosity is evaluated using chemical methods, but these techniques can be rather complex and time consuming. Optical microscopy and digital imaging analysis have been previously used [41] to investigate coal samples, but these methods have their own limitations in terms of the minimum size of pores that can be observed. As it was mentioned earlier, ordinary microscopic techniques do not provide the 3D depth information about the sample. The porosity evaluation using AFM fails as the pores are too deep for the tip. Such limitations in analyzing the pores of coal samples can be overcome by reflection digital holographic microscopy as shown in this work.

The samples, imaged here, were acquired from the Coal Group at the National University of Colombia, Campus Medellin. These samples were treated by pyrolysis and prepared in a mixture with epoxy resin; then they were ground and polished with aluminum oxide abrasives (1, 0.5, 0.03 μm grain size). The process ensured that the samples are reflective and firm enough to be imaged.
Figures 6.1 and 6.2 shows the experimental setup, based on two overlapping Michelson interferometers that enable to fine-tune the location of the first-order components associated with each wavelength in the Fourier space. Its configuration is similar to the setups based on the modified Mach-Zehnder [42-46] interferometers, as typical for the reflection digital holographic microscopy. He-Ne ($\lambda_1=633$ nm) and diode-pumped solid-state ($\lambda_2=532$ nm) lasers were used as light sources. Neutral density filters (ND) control the intensity of the laser beams. 20x microscope objectives OBJ11 and OBJ12, together with the pinholes A, and the collimator lenses L11/L12 produce uniform plane waves, whose intensity is further controlled by the polarizing filters P1 and P2. Beam splitters BS1 and BS2 divide the beams into the reference and the object waves, which are reflected by the reference mirrors and the object. Thereafter, the beam splitters direct the waves toward the CCD camera.

Lenses L21 and L22 and 20x microscope objectives OBJ21 and OBJ22 again collimate the reference waves. Two separate reference arms are used to match the object path lengths for each object wave. An interference filter is placed into the reference arm of the diode-pumped solid-state ($\lambda=532$ nm) laser. It is designed to allow only this wavelength to pass and block the inverse reflection of the other laser. The 20x microscope objective OBJ1 focuses a magnified image of the sample onto the sensor of the CCD camera, where the interference pattern between the reflected reference and object waves is recorded. The images are acquired and processed using a set of Labview (see Appendix E) and C programs (see Appendix F), which were developed for this project.
Figure 6.1: Dual-wavelength digital holography setup for coal samples. The lateral magnification of all microscope objectives (OBJ) is 20x. The focal length of the lenses L21 and L22 are 17.5 cm and 10 cm respectively. The ND filters and polarizers P1 and P2 are used to control the intensity of the laser beams. Pinholes A are used to select only the central part of the Gaussian beam. Lenses L11, L12, L21 and L22 and objectives OBJ1, OBJ21 and OBJ22 assure an appropriate collimation of the waves (i.e. the beam waist is kept at “infinity”).
Figure 6.2: Digital holographic microscope system.
An angle between the object wave and each of the reference waves can be introduced by slightly tilting the reference mirrors. Furthermore, by tilting the two reference beams orthogonally to each other, we can precisely control the location of each spectral component in Fourier space. As a result, the two spectral components can be sufficiently separated to enable the effective filtering in the Fourier domain, which in turn allows for the real-time imaging.

6.2 Two-wavelength optical phase unwrapping

In Figure 6.3, a boundary between the porous coal and the resin is shown. The amplitude image Figure 6.3a is similar to what a regular microscope would display. The area corresponding to the resin is at the lower right corner. It is apparent that the boundary is barely visible in the amplitude image, as there is no significant difference between the reflection from the coal and resin.

Figure 6.3b and Figure 6.3c show the single wavelength phase images, which display multiple discontinuities that were removed in the dual wavelength coarse map (Figure 6.3d). The phase noise is significantly reduced in the fine map (Figure 6.3e), as appreciated in its 3D rendering (Figure 6.3f). The latter image shows the 3D surface profiles of coal and resin surfaces, which appear to “bend” towards each other at the boundary between the two surfaces, so the boundary itself is seen very clearly.
**Figure 6.3:** Boundary in a porous coal sample: (a) amplitude image; phase maps reconstructed at (b) $\lambda_1=0.63$ $\mu$m and (c) $\lambda_2=0.53$ $\mu$m; (d) the dual-wavelength coarse phase map, (e) fine map and (f) its 3D rendering. All images are 98x98 $\mu$m$^2$ and vertical scale (b-e) is in radians.
6.3 Comparison between dual-wavelength and software phase unwrapping

Figure 6.4 shows the images of porous coal samples treated with pyrolysis. Once again, the phase images at a single wavelength clearly exhibit $2\pi$ phase steps (Figures 6.4b and Figure 6.4c), while the dual wavelength unwrapped phase map in Figure 6.4d shows very few discontinuities. The parts of the images where discontinuity is still present correspond to low reflectivity areas on the sample hologram, where no interference pattern is visible. Consequently, the phase there is basically a random noise, which gives rise to multiple $2\pi$ phase jumps. These can generally be identified as deeper pores. With this method, pores with the lateral size on the order of a micron can be identified.

It is worth noting that unwrapping the single wavelength phase images using conventional algorithms is very problematic, as illustrated in Figure 6.4e and Figure 6.4f. A typical software unwrapping algorithm starts at a certain point of an image and moves along a 1D path (e.g. straight line, spiral). If it encounters what looks like a phase wrap, it shifts the map down/up. If the image has noisy areas (corners in Figure 6.4e and Figure 6.4f), where phase oscillates randomly, the software algorithm takes it as a real feature and creates nonexistent steps in phase/height profile, which clearly do not correspond to the real height profile of the sample.
Figure 6.4: Images of a porous coal sample: (a) amplitude image; phase maps reconstructed at (b) $\lambda_1=532$ nm and (c) $\lambda_2=633$ nm; (d) 3D rendering of the dual-wavelength phase map; software unwrapped phase maps reconstructed at (e) $\lambda_1=633$ nm and (f) $\lambda_2=532$ nm for comparison. All image sizes are 98x98 microns. The vertical scales of the phase maps are in radians.
6.4 The application of the fine map algorithm

Figure 6.5 shows the images of an activated porous coal in a resin sample. The areas on the left and right sides of the image are too dark, but the central region is well illuminated. Two vertical scratches due to the coal-cutting process are clearly visible. Figure 6.5a and Figure 6.5b represent the coarse and the fine phase maps respectively. A noise reduction by a factor of five is apparent by comparing the height profiles Figure 6.5c and Figure 6.5d from the coarse and the fine maps respectively. For further comparison, the single wavelength phase map and the same line profile are sketched in Figure 6.5e and Figure 6.5f respectively.

The “spots” in the images Figure 6.5c and Figure 6.5d, as well as the spike around 120 μm in Figure 6.5f, result from the high level noise in the single wavelength images. Consequently, some of the wavelength segments are erroneously shifted by a wavelength. Artifacts like these are rare and they do not prevent us from obtaining an accurate 3D picture of the sample. From the depth information available from the fine map (Figure 6.5d), one can see, for example that the depth of a small feature (scratch located around 75 μm from the left on all the line profiles) is about 100 nm, which is again consistent with the single wavelength phase map (Figure 6.5f). Obviously, the single wavelength possesses multiple artifacts due to phase wrapping.
Figure 6.5: Images of a porous coal sample (a) Coarse phase map, (b) fine phase map and line profiles for (c) coarse, (d) fine. For comparison, (e) single wavelength phase map at $\lambda=633$ nm and (f) line profile (the image sizes are 138x138 $\mu$m$^2$ and the vertical scales of the phase images are in radians).
CHAPTER 7
CELL IMAGING

7.1 Experiment

Dual-wavelength phase-imaging technique was also used to quantitatively study the three-dimensional structure of cells. We have obtained 3D images of SKOV-3 ovarian cancer cells with diffraction limited lateral resolution and axial resolution on the order of 5 nm. The cells display intracellular features with sufficient clarity to measure the thickness of the cell’s lamelipodium and observe the features of its nucleus. A similar study had been previously done using a single wavelength and software phase unwrapping [47].

Figure 7.1 shows the experimental apparatus. It is based similar to the design presented in Figure 6.1. The only major difference is that here the wave fronts in both reference arms remain spherical and the resulting curvature mismatch removal is entirely numerical. Once again, a relative angle can be introduced between the object and each of the two reference beams by slightly tilting the reference arms mirrors. By introducing different tilts in two orthogonal directions for two reference beams, we can separate each spectral component in Fourier space, which allows us to capture both wavelengths simultaneously. The actual interferometers are shown in Figure 7.2.
Figure 7.1: Dual-wavelength digital holography setup for imaging of cells. The focal length of the lenses L21 and L22 are 17.5 cm and 10 cm respectively. The beams are collimated between L11 and L21 and between L12 and L22 and again are collimated after 20x OBJ1 microscope objective.
Figure 7.2: Dual interferometers
7.2 SKOV-3 ovarian cancer cells

Here, we have applied the dual-wavelength phase imaging method to 3D imaging of SKOV-3 ovarian cancer cells. Figure 7.3 shows the confluent group of cells: Figure 7.3a shows the intensity image, which is similar to what one can see using the ordinary microscope, while Figure 7.3b displays a single wavelength wrapped phase image, and Figure 7.3c shows the coarse dual-wavelength unwrapped phase image. Finally, Figure 7.3d displays 3D rendering of the final fine map, where we see the cells connecting together with grooves between them. The area at the bottom of the images is the exposed part of the gold substrate, to which the cells are bound. The measurements of the optical thickness of cells can then be performed using equation 3.3. One also needs to make an assumption of the cells refractive index, which we took to be 1.375. While it may not be possible to precisely determine the refractive index of the cell at each individual point, this number is always close to the refractive index of water and unlikely to deviate by more than a few percent.

Figure 7.4 shows the image of SKOV-3 single cell, where the cell’s nucleus and pseudopodia are clearly seen. Once again, by using the phase to thickness conversion, we can easily determine the 3D features of the cell. In addition to phase images for a single wavelength (Figure 7.4a), coarse map (Figure 7.4b) and 3D rendering of the fine map (Figure 7.4c), Figure 7.4d displays the line intensity profile, which indicates, for example, that the overall cell height is about 1.47 μm. The separate measurement indicates that the thickness of the cells pseudopodia (lamelipodia) is around 270 nm.
Figure 7.3: Confluent SKOV-3 ovarian cancer cells: (a) amplitude image, (b) reconstructed phase for $\lambda=532$ nm, (c) dual-wavelength coarse phase image and (d) 3D rendering of fine map. All images are 92x92 $\mu$m$^2$ (240x240 pixels).
Figure 7.4: A single SKOV-3 cell: (a) reconstructed phase for $\lambda=633$ nm, (b) dual-wavelength coarse phase image, (c) 3D rendering of fine map and (d) line thickness profile. All images are 63.5x59 $\mu$m$^2$ (165x153 pixels).
Finally, the image in the Figure 7.5 shows a different confluent area of the same sample. Once again, the phase images generated using one wavelength clearly exhibit a number of $2\pi$ phase steps (see Figure 7.5b), while the dual wavelength unwrapped phase map Figure 7.5c shows a few spots where discontinuities are still present. These spots correspond to the lower intensity areas on the sample where no interference fringes were obtained. As a result, the phase is a random noise, which gives rise to multiple $2\pi$ phase steps. The images in Figure 7.5d and Figure 7.5e show the result of optical and software unwrapping respectively. Since this phase image has a noisy area, where phase oscillates randomly, the software algorithm took it as a real feature and created a step in phase profile (upper right corner of Figure 7.5e), which clearly does not correspond to the real thickness profile of the sample.
Figure 7.5: Comparison between optical and software unwrapping: (a) amplitude image; (b) single wavelength phase image, (c) coarse maps, (d) 3D rendering of the dual-wavelength fine phase map and (e) software unwrapped phase map. Images are 98x98 μm² (256x256 pixels).
CHAPTER 8
CONCLUSIONS

8.1 Laser scanning microscope

The concept of the laser scanning microscope with the position sensitive detector, demonstrated here, has a number of potential advantages. The resolution determining parameter is the actual size of the illuminating scanning beam and not the ability of the detector to receive and resolve the image through optics. The optical system is used only to focus the laser beam to an axial point. Therefore, the system is free from all non-longitudinal aberrations such as coma, astigmatism, field curvature, and distortion. If both the lateral resolution and the depth of focus of this system are improved (by means of a stronger telescopic system or by using Bessel beams), the system can be very effective in capturing images with submicron lateral resolution, while maintaining quite substantial depth of focus. The position sensitive detector is only one of possible detectors that can be used. In fact, there is much freedom in a type of a detector to choose. For example, weak or polarization-sensitive fluorescence can be detected by lock-in technique and the use of appropriate polarization elements in the system.

Faster scanning by using acousto-optic modulator or micro-electromechanical scanners can be considered to achieve video rate acquisition. In addition to improving the scanning beam characteristics, we can also look at the possibility of collecting light with
higher efficiency via optical fiber. Although the signal-to-noise ratio was acceptable for the samples that we have looked at, in order to make the system more versatile, adding the high NA fiber to both increase the amount of light that reaches detector, and to make it easier to collect light from different angles can be advantageous. Also, by inserting a beamsplitter between the laser and the scanning system, and capturing the reflecting image, the system can be configured as a confocal microscope. Finally, our system can also be used to scan in color. For that purpose, three lasers can be employed to simultaneously scan the sample, while three detectors collect light for RGB components.

In addition to all that, the system is very inexpensive even in comparison to a simple microscope model, can be made very compact and does not require complicated sample preparations. In theory, the method can be an optical analog of SEM capable of generating high quality images over a significant depth of focus. Although the lateral resolution is limited to a micron range, the samples can be imaged in its natural state and the imaging system is simple and cost-effective.

8.2 Dual-wavelength digital holography

The dual-wavelength phase imaging digital holography technique proved to be a powerful method of 3D imaging with the $2\pi$-ambiguity resolved. Its application to the detection and study of pores in coal samples has been demonstrated. In comparison to the software unwrapping, dual-wavelength optical unwrapping method is advantageous, as it requires no intensive computation procedures and can handle complex phase topologies. The method provides high-resolution, accurate quantitative profiles of surfaces and can be an effective tool in studying small and large scale 3D features of many natural and
manmade samples. The use of two wavelengths together with the fine map algorithm allows us to increase the maximum height of the features which can be imaged, while keeping the noise low (few nanometers).

The selection of two wavelengths which are closer to each other increases the axial range, but also increases the noise to the levels where fine map algorithm begins to fail. In order to further increase the axial range and still keep the final noise levels low, the same procedure can be applied at three- or more wavelengths. Furthermore, it is possible to use a tunable laser to iteratively increase the range while reducing the noise to the desired levels.

We have also demonstrated the application of digital holography for studying cells. As a result, the accuracy and the level of details of the dual-wavelength images of cells, presented here, are superior to what has been previously demonstrated.

The proposed method of curvature correction is simple and effective enough to easily implement the experiment without the microscope objectives in the reference arms of the Michelson interferometer. This greatly simplifies the optical setup and makes it much easier to do the initial adjustments of the apparatus. Simultaneous dual-wavelength setup utilized together with the angular spectrum algorithm provides an easy way to acquire single frame images in real time, which can be used to study cell migration.

The results of this work had been published in several journals and conference proceedings (see Appendix G).
REFERENCES


[38] P. Ferraro, S. De Nicola, A. Finizio, G. Coppola, S. Grilli, C. Magro and G. Pierattini, "Compensation of the inherent wave front curvature in digital holographic


APPENDIX A
LASER SPOT SIZE

In order to measure the laser spot size on the sample, the beam was gradually blocked by using a sharp blade attached to the computer controlled precision moving stage. The overall intensity was then measured by the power meter (Newport model 1815-c). If the beam profile is assumed to be Gaussian, the blocking of its part can be simulated (see Figure A.1) to obtain the percentage of the beam that passes through, when everything up to the beam waist radius is transmitted.

Thus, when the detector registers 97.72% of the total intensity, the beam is blocked at its waist. When the detector registers 50% of the total intensity, half the beam is blocked and therefore this represents the center of the beam. The relative distance, the blade needs to be moved to achieve these levels of intensity is then equal to the beam spot size.

Laser beam waist size and location was measured by simply measuring the size of the laser spot in 3 places, shown in Figure A.2. The experimental values of the beam spot size were then compared to theoretical values (Figure A.3). Thus, the beam waist is located on the output mirror and its size is 360 microns.
Finally, Figure A.4 shows the calculation of the final (as it is projected onto the sample) beam spot size and scanning parameters for a galvanometer based scanning mirror and a telescopic system.
\textbf{Figure A.1:} Mathematica simulation of Gaussian beam.
Figure A.2: Measuring laser spot in three places
The confocal input beam parameter (in m):

\[ w_0 = 360. \times 10^{-6}; \]
\[ n = 1; \]
\[ \text{wave} = 0.63 \times 10^{-6}; \]
\[ z_0 = \frac{\pi w_0 w_0 n}{\text{wave}} \]
\[ z = -0.0; (* \text{the location of the beam waist - if 0 - on laser} *) \]
\[ z_1 = z + 0.3; \]
\[ z_2 = z + 0.8; \]
\[ z_3 = z + 1.3; \]

0.64527

The size of the beam (in micron):

\[ w_1 = w_0 \sqrt{1 + \left(\frac{z_1}{z_0}\right)^2} \times 10^6 \]
\[ w_2 = w_0 \sqrt{1 + \left(\frac{z_2}{z_0}\right)^2} \times 10^6 \]
\[ w_3 = w_0 \sqrt{1 + \left(\frac{z_3}{z_0}\right)^2} \times 10^6 \]

396.396
572.378
808.703

The confocal output beam parameter (in mm):

\[ w = w_0 \times 10^{-5}; \]
\[ z_1 = \frac{\pi w w n}{\text{wave}} \times 10^3; \]

Figure A.3: Mathematica simulation of beam profile
First Lens parameters (in m):

\[ f = 0.1; \quad (\text{focal length of the 1st lens}) \]
\[ d = 0.15; \quad (\text{distance between laser and the first lens}) \]
\[ L1 = \{(1.0, 0.0), (-1.0/f, 1.0)\}; \]
\[ S1 = \{(1.0, d), (0.0, 1.0)\}; \]

Second Lens parameters (in m):

\[ f = 1; \quad (\text{focal length of the 2nd lens}) \]
\[ d = 1.1; \quad (\text{distance between previous lens and this lens}) \]
\[ L2 = \{(1.0, 0.0), (-1.0/f, 1.0)\}; \]
\[ S2 = \{(1.0, d), (0.0, 1.0)\}; \]

Focusing Lens parameters (in m):

\[ f = 0.1; \quad (\text{focal length of the focusing lens}) \]
\[ d = 0.05; \quad (\text{distance between the previous lens and scanning mirror}) \]
\[ dtarget = 0.1; \quad (\text{distance between the last lens and the image}) \]
\[ Sn = \{(1.0, d), (0.0, 1.0)\}; \]
\[ Sf = \{(1.0, f), (0.0, 1.0)\}; \]
\[ Lf = \{(1.0, 0.0), (-1.0/f, 1.0)\}; \]
\[ Target = \{(1.0, dtarget), (0.0, 1.0)\}; \]

Input beam parameters (in m), get:

\[ w1 = 0.360 \times 10^{-3}; \]
\[ R1 = 100000000.0; \quad (\text{Radius of curvature - almost <<}) \]
\[ n = 1; \]
\[ \text{wavelength} = 0.63 \times 10^{-2}; \]

Figure A.4: Mathematica simulation of final beam profile
APPENDIX B
LASER SCANNING MICROSCOPE LABVIEW PROGRAM

Figure B.1 below shows the main screen of the program used to acquire images for laser scanning microscope.
Figure B.1: Main screen of laser scanning microscope Labview program.
APPENDIX C

AXICON

The word “axicon” was first proposed by J. McLeod [48-49] to characterize any figure of revolution that by reflection, or refraction, or both will bend light from a point source on the axis of revolution so as to cross the axis not at the point, as would be the case of a lens, but along a continuous line of points of a substantial extent of the axis. Thus, the definition of axicon is not only limited to the conical lens, but may include many rotationally symmetric structures, such as a circular slit.

In the case of on-axis illumination by the Gaussian beam, an axicon produces the kind of interference pattern, which can be thought of as being non-diffractive. The idea is that the solution of the wave equation in this case is separable into two parts – one depends only on the transverse coordinate, and another - on axial. The intensity distribution can be expressed in term of the Bessel function.

While propagating along the axial direction, the energy flows in and out of the central maximum. In the case of the (ideal) Bessel beam, this does not result in the overall increase of the size of a central maximum. One should point out that therefore, when isolated, the central maximum will spread out as quickly as the Gaussian beam of the same waist.

Although for the ideal (infinite aperture) Bessel beam, the size of the central maximum will stay the same indefinitely, the non-ideal Bessel beam will not. In fact, it
has been shown both theoretically [16] and experimentally [17] that the maximum intensity of the Bessel beam will oscillate increasingly along the axial direction before rapidly decaying, when it reaches the geometrical shadow [18] (Figure C.1).

The conical surface of an axicon generates the Bessel beam with on-axis illumination. As it was shown before, the useful approximation exists to calculate the intensity distribution produced by an axicon (Equation 27 from [19]). Mathematica routine, shown in Figure C.2 was written to try out different wedge angles and see what would be the spot size and the depth of focus.

The results of this simulation were consistent with the theory outlined in [16] and [19]. The basic conclusion is that for the smaller beam spot size (the size of the central maximum) one needs smaller cone angle. Also, for a longer depth of focus, a bigger cone angle is needed. Running the simulation, and assuming the realistic parameters, it was concluded that for the sharp (approximately 1 micron) first peak and still relatively long (11 mm) depth of focus, we could use an axicon with the cone angle of 140 degrees (See Figure C.3).

Unfortunately, the images obtained with the axicon exhibited a rather poor contrast. The intensity distribution projected and observed on the object has a clearly defined zero-order diffraction peak. When this peak is used for scanning, it forms a bright line on the object, which is clearly visible with the naked eye (see Figure C.4). However, if one looks at the intensity profile in two dimensions, upon integration, the fraction of intensity in the central peak was calculated to be only 1.4% of the total energy. As a result, since our non-imaging detector looks at the total energy, there was very little light for it to work with.
Figure C.1: The depth of focus of the Bessel beam. $L$ is the “focal length” of the axicon.
Axicon intensity distribution and depth of focus

In[]: Clear[\[z, r]\];

- The distance along the axial direction (focus)

\[\text{In}[7]: \text{a} = 0.05;\]

- Order of the Gaussian beam

\[\text{In}[8]: \text{m} = 0;\]

- Spot size of the Gaussian beam

\[\text{In}[9]: \text{d} = 2 \text{10}^{-3};\]

- Wavelength of the Gaussian beam and index of refraction and wedge angle of Axicon

\[\text{In}[10]: \text{\[\lambda\]} = 0.63 \text{10}^{-6};\]

\[\text{In}[11]: \text{\[Axicon\]} = 1.5;\]

\[\text{In}[12]: \text{\[\alpha\]} = 20. \frac{\text{rad}}{\text{100}};\]

- Eq 2. 6, 13.19 and 24

\[\text{In}[13]: \text{\[\beta\]} = \frac{2 \pi (\text{\[Axicon\]} - 1) \text{\[\alpha\]}}{\text{\[\lambda\]}},\]

\[\text{In}[14]: \text{\[n\]} = \text{\[\beta\]} \text{\[d\]};
\]

\[\text{In}[15]: \text{\[Ld\]} = \frac{\text{\[n\]} \text{\[d\]}^2}{\text{\[\lambda\]}},\]

\[\text{In}[16]: \zeta = \frac{\text{\[n\]} \text{\[d\]}^2}{2 \text{\[Ld\]}},\]

\[\text{In}[17]: \xi = r / \text{\[d\]};
\]

\[\text{In}[18]: \text{\[F1\]} = (r_0 - \xi) \text{\[n\]} \text{\[d\]}^{1/2} \text{Exp}[-(r_0 - \xi)^2];\]

\[\text{In}[19]: \text{\[F2\]} = (r_0 - \xi) \text{\[n\]} \text{\[d\]}^{1/2} \text{Exp}[-(r_0 - \xi)^2] \text{UnitStep}[r_0 - \xi];\]

- Intensity distribution (Eq 27) by axicon

\[\text{In}[20]: \text{Intensity} = \frac{\pi \text{\[n\]} \text{\[d\]}^2}{2} \left[ (\text{\[F1\]} + \text{\[F2\]}) \text{BesselJ\[n, (\text{\[n\]} \xi)\]}^2 + (\text{\[F1\] - \text{\[F2\]}) \text{BesselJ\[n - 1, (\text{\[n\]} \xi)\]}^2 \right];\]

\[\text{Plot}[\text{Intensity}, \{r, 0, 0.00001\}]\]

\[\text{Out}[20]: \text{Graphics}\]

- Depth of focus (Eq 3 from imaging properties of axicon)

\[\text{In}[21]: \text{\[d\]} = \frac{\text{\[d\]} \text{\[Axicon\]} - 1) \text{\[\alpha\]}}{\text{\[Axicon\]}},\]

\[\text{In}[22]: \text{\[d\]} = 0.014592\]

**Figure C.2: Mathematica simulation of axicon parameters**
Figure C.3: The axicon illuminated by a Gaussian beam (left) and theoretical intensity distribution in the focal plane (right).
Figure C.4: The intensity distribution in the focal plane of axicon (left) and zoom of the central peak (right). The units of x- and y-axes are meters.
APPENDIX D

THEORY OF DUAL-WAVELENGTH PHASE IMAGING

The principle of multi-wavelength phase imaging, presented here is from the reference [40]. Suppose, we image a sloped mirror, which will appear as a tilted plane with multiple discontinuities (see figure 5.1). Let’s assume that the overall height of this plane is \( h = 5.0 \mu m \). Figures D.1 (a) and D.1 (b) display the phase maps \( \phi_1 \) and \( \phi_2 \) of the tilted object using wavelengths of \( \lambda_1 = 532nm \) and \( \lambda_2 = 633nm \) respectively. The phase maps contain multiple \( 2\pi \) discontinuities wherever the height is a multiple of the wavelength. Subtraction of the two phase maps \( \phi_1 \) and \( \phi_2 \) in Figures D.1 (a) and D.1 (b), results in a new phase map \( \phi_{12} = \phi_1 - \phi_2 \) as shown in Figure D.1 (c). By adding \( 2\pi \) to the phase map in Figure D.1 (c) wherever \( \phi_{12} < 0 \) produces a new phase map \( \phi_{12}(x) = \phi_{12}' + 2\pi \cdot (\phi_{12}' < 0) \) with a longer range free of discontinuities and extended axial range (Figure D.1 (d)). The new phase map is equivalent to that of a “beat wavelength”, which, in the case of using \( \lambda_1 = 532nm \) and \( \lambda_2 = 633nm \), is found as

\[
\Lambda_{12} = \frac{\lambda_1 \lambda_2}{|\lambda_1 - \lambda_2|} = 3.33 \mu m
\]

(D.1)
Figure D.1: Dual-wavelength phase imaging digital holography (a) phase map $\phi_1(x)$ of $\lambda_1 = 532 nm$; (b) phase map $\phi_2(x)$ of $\lambda_2 = 633 nm$; (c) difference phase map $\phi_{12}'(x) = \phi_1 - \phi_2$; (d) coarse map $Z_{12}'(x)$, with beat wavelength $\Lambda_{12} = 3.33 \mu m$. 

\[ \phi(x) = \frac{2\pi}{\lambda} \sin^2 \theta = \frac{2\pi}{\lambda} \left( \cos 2\theta - 1 \right) \]
However, by amplifying the range, any phase noise is amplified as well by the same factor. Suppose the single-wavelength phase maps $\phi_m(x)$ contain phase noise of $2\pi \varepsilon_m$, where $\varepsilon_m \sim 2\%$. Then, the corresponding surface profiles $Z_m(x)$ contain a noise level of

$$\varepsilon_m \lambda_m \sim 12 \text{nm} \tag{D.2}$$

The noise in the difference phase map $\phi_{12}(x)$ is then

$$2\pi \varepsilon_{12} = 2\pi (\varepsilon_1 + \varepsilon_2) \tag{D.3}$$

and that in the surface profile noise $Z'_{12}(x)$ is

$$\varepsilon_{12} \Lambda_{12} \sim 130 \text{nm} \tag{D.4}$$

Thus, the noise has in effect been amplified approximately by a factor of $2\Lambda_{12} / \lambda_m$ as one can see in the coarse map $Z'_{12}(x)$ in Figure D.1 (d) when compared to the single-wavelength phase maps in Figure D.1 (a) or Figure D.1 (b).

Now, by using the fine map algorithm, it is possible to reduce the noise in $Z'_{12}(x)$ back to the level of the single-wavelength phase maps. The coarse profile, $Z'_{12}(x)$ shown in Figure D.1 (d) can be divided into integer multiples of either of the wavelengths to produce a new coarse profile $Z''_{12}(x)$ as illustrated in Figure D.2 (a).

Using $\lambda_1$, the new profile is defined as

$$Z''_{12}(x) = \text{int} \left( \frac{Z'_{12}(x)}{\lambda_1} \right) \cdot \lambda_1 \tag{D.5}$$

One can then paste segments of a single-wavelength onto this profile to obtain profile $Z''_{12}(x)$ as shown in Figure D.2 (b) such that
Figure D.2: Fine map generation (a) $Z''_{12}(x)$, obtained from dividing $Z'_{12}(x)$ into integer multiples of $\lambda_1$; (b) $Z'''_{12}(x)$, where $Z_1(x)$ is pasted on $Z''_{12}(x)$; (c) $Z_a$, derived from comparing $Z''_{12}(x)$ with the coarse map $Z'_{12}(x)$; (d) $Z_b$, resulting from addition or subtraction of $\lambda_1$ in $Z_a$ to remove spikes; (e) the fine map, $Z_{12}(x)$. 
At the boundaries of wavelength intervals, the noise in the single-wavelength phase map causes numerous jumps of size, $\pm \lambda_i$ as displayed in Figure D.2 (c). If the noise level is not excessive, most of the spikes in the can be removed by simply comparing of $Z_{12}^n(x)$ with $Z_{12}'(x)$ (see Figure D.2 (d)), and if the difference is more than $\lambda_i$, then $\lambda_i$ is either added or subtracted depending on the sign of the difference. Figure D.2 (e) shows the final result of the fine map procedure, where the noise level is approximately the same as that of $Z_i(x)$, the single wavelength profile, at around 12nm. The remaining spikes in the map are due to places where the coarse map is more than one half of $\lambda_i$. In this work, these remaining spikes are removed by a software routine, which scans over the image and removes these $\lambda_i$-high artifacts.

It is estimated [40] that, the maximum noise level for the method to work properly is given approximately by

$$\varepsilon_m < \frac{\lambda_m}{4\Lambda_{12}} \sim 4\%$$  \hspace{1cm} (D.7)

If the noise levels are not too high, the phase-unwrapping technique can be further extended to an iterative procedure of three or more wavelengths, which would yield an even bigger axial range.
APPENDIX E

DUAL-WAVELENGTH DIGITAL HOLOGRAPHY LABVIEW PROGRAMS

During the course of this work, a number of Labview programs were developed to both acquire and process the holographic images. Figure E.1 below shows the main screen of the program for dual-wavelength digital holography. It incorporates both wavelength images with curvature corrections, Fourier transforms with multiple filtering capabilities, phase images, coarse and fine maps.

Figure E.2 shows the diagram of dual-wavelength phase unwrapping. Figure E.3 shows the diagram of curvature correction Labview routine. This routine is called from the main program, with the set of parameters specified by user in the main screen (Figure E.1).
Figure E.1: Main screen of dual-wavelength Labview program.
Figure E.2: Dual-wavelength phase unwrapping Labview diagram.
Figure E.3: Curvature correction Labview diagram.
APPENDIX F

DUAL-WAVELENGTH DIGITAL HOLOGRAPHY C PROGRAMS

The printouts of all C routines developed to increase the speed of FFT in angular spectrum method are shown below.

```c
// spectrum.cpp : Defines the entry point for the DLL application.
/* Call Library source file */

#ifndef _extcode
#include "extcode.h"
#endif

#include "stdafx.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "fftn.h"

#undef REAL
#define REAL double

#define P2 6.283185307179586476925286766559

/* Typedefs */

typedef struct {
    double re, im;
} cmplx128;

typedef struct {
    long dimSizes[2];
    cmplx128 Numeric[1];
} TD2;

typedef TD2 **TD2Hdl;

typedef struct {
    double dxUm;
    double dyUm;
    TD2Hdl array2Dc;
} TD1;

typedef struct {
    double lambdaUm;
    double ZUm;
} TD3;

typedef struct {
    long x0;
    long y0;
} TD4;
```
__declspec(dllexport) void angspec(TD1 *HH2Dc, TD3 *diffractParam, TD1 *EE2DcDll, TD1 *filter2Dc, TD4 *shift, TD1 *FF2DcDll);

__declspec(dllexport) void angspec(TD1 *HH2Dc, TD3 *diffractParam, TD1 *EE2DcDll, TD1 *filter2Dc, TD4 *shift, TD1 *FF2DcDll)
{
    // TD2Hdl array;
    double *arre,*arim;
    int *map_fourier, *map_filter;
    int i, j;
    int ret;
    int dims[2];  /* pass fft dimensions */
    double k2,ky2;
    double *kx2,*ky2;
    double phase, mult;
    struct CENTER {
        int y, x;
    };
    struct CENTER center;
    /* dimSizes[0] is the number of rows */
    numrow = (*(HH2Dc->array2Dc))->dimSizes[0];
    numcol = (*(HH2Dc->array2Dc))->dimSizes[1];
    center.y = (int) (numrow/2);
    center.x = ( int) (numcol/2);
    FF2DcDll->dxUm = P2 / numrow / HH2Dc->dxUm;
    FF2DcDll->dyUm = P2 / numcol / HH2Dc->dyUm;
    k2 = P2*P2/(diffractParam->lambdaUm)/(diffractParam->lambdaUm);
    // numrow = HH2Dc->array2Dc.dimSizes[0]; // *array2Dc-> dimSizes[0];
    // numcol = TD1->array2Dc->dimSizes[1];
    /* x,y array dimensions to pass */
    dims [0] = numcol;
    /* scale one of these ways: */
    /* this is what MATLAB does, but it was then adjusted in Labview, so see below */
    #define FORWARD_SCALE 0.0
    #define INVERSE_SCALE -1.0
    */
    #define FORWARD_SCALE -2.0
    #define INVERSE_SCALE -2.0
    */
    /* Create separate real/imaginary arrays and shifting map*/
    arre = (double *) malloc (numrow * numcol * sizeof(double));
    arim = (double *) malloc (numrow * numcol * sizeof(double));
    kx2 = (double *) malloc (numcol * sizeof(double)); // check row and col
    ky2 = (double *) malloc (numrow * sizeof(double));
    map_fourier = (int *) malloc (numrow * numcol * sizeof(int));
    map_filter = (int *) malloc (numrow * numcol * sizeof(int));
    }
for (i = 0; i < numrow; i++) { /* copy data into the arrays */
    for (j = 0; j < numcol; j++) {
        arre[(i * numcol) + j] = (**(HH2Dc->array2Dc))->Numeric[(i * numcol) + j].re;
        arim[(i * numcol) + j] = (**(HH2Dc->array2Dc))->Numeric[(i * numcol) + j].im;
    }
}

/* Call to FFT - fwd transform, separate real/imaginary arrays */
ret = fftn (2, dims, arre, arim, 1, FORWARD_SCALE);
if (ret) return 1; // That is how it is done in MATLAB - for display purposes

/* DO THE SHIFTING MAP to Shift the arrays - shift is negative to match Labview */
shift_map(map_fourier, numrow, numcol, (center.y - shift->y0), (center.x - shift->x0));
shift_map(map_filter, numrow, numcol, -shift->y0, -shift->x0);

/* ----------now copy into FF ----------- */
/* REMEMBER - arim returned is (-arim) as it would have been in MATLAB */
for (i = 0; i < numrow; i++) {
    for (j = 0; j < numcol; j++) {
        (**(FF2DcDll->array2Dc))->Numeric[(i * numcol) + j].re =
            arre[map_fourier[(i * numcol) + j]] *
            (**(filter2Dc->array2Dc))->Numeric[map_filter[(i * numcol) + j]].re +
            arim[map_fourier[(i * numcol) + j]] *
            (**(filter2Dc->array2Dc))->Numeric[map_filter[(i * numcol) + j]].im;
        (**(FF2DcDll->array2Dc))->Numeric[(i * numcol) + j].im =
            -arim[map_fourier[(i * numcol) + j]] *
            (**(filter2Dc->array2Dc))->Numeric[map_filter[(i * numcol) + j]].re +
            arre[map_fourier[(i * numcol) + j]] *
            (**(filter2Dc->array2Dc))->Numeric[map_filter[(i * numcol) + j]].im;
    }
}

fflush (stdout);

/* ----------get kx and kz grid ---------- */
kx2=grid(center.x, numcol, FF2DcDll->dxUm);
ky2=grid(center.y, numrow, FF2DcDll->dyUm);
shift_map(map_fourier, numrow, numcol, -center.y, -center.x);

/* ----------multiply by exp ----------- */
for (i = 0; i < numrow; i++) {
    for (j = 0; j < numcol; j++) {
        kz2 = k2 - kx2[j] - ky2[i];
        if (kz2>=0) {
            phase=diffraction->ZUm * sqrt(kz2);
            arre[map_fourier[(i * numcol) + j]] =
/* Call to FFT - inverse transform, separate real/imaginary arrays */
ret = fftn (2, dims, arre, arim, 1, INVERSE_SCALE); //changed -1 to 1 !!!!
mult = 1./numcol/numrow; //sqrt((double) numcol*numrow);
for(i = 0; i < numrow; i++)
{
    for(j = 0; j < numcol; j++)
    {
        (*(EE2DcDll->array2Dc))->Numeric[(i * numcol) + j].re =
            arre[(i * numcol) + j]*mult;
        (*(EE2DcDll->array2Dc))->Numeric[(i * numcol) + j].im =
            arim[(i * numcol) + j]*mult;
    }
}

/* done, free up dynamically allocated memory */
fftn_free ();
free (arim);
free (arre);
free (kx2);
free (ky2);
free (map_fourier);
free (map_filter);

#include "stdafx.h"
#include <stdlib.h>
#include <stdio.h>

void shift_map (int *map, int ysize, int xsize, int yshift, int xshift)
{
    int i, j,
        index1, index2;

    while(yshift<0) yshift+=ysize;
    while(xshift<0) xshift+=xsize;

    while(yshift>ysize) yshift-=ysize;
    while(xshift>xsize) xshift-=xsize;

    for (i = 0; i < ysize; i++)
    { /* go over the array filling the shift map array */
        for (j = 0; j < xsize; j++)
        {
            (*(FF2DcDll->array2Dc))->Numeric[(i * numcol) + j].re *
                cos(phase) -
                (*(FF2DcDll->array2Dc))->Numeric[(i * numcol) + j].im *
                sin(phase);

            arim[map_fourier[(i * numcol) + j]] =
                (*(FF2DcDll->array2Dc))->Numeric[(i * numcol) + j].re *
                sin(phase) +
                (*(FF2DcDll->array2Dc))->Numeric[(i * numcol) + j].im *
                cos(phase);

            (*(FF2DcDll->array2Dc))->Numeric[(i * numcol) + j].re *= 2;
            (*(FF2DcDll->array2Dc))->Numeric[(i * numcol) + j].im *= 2;
        }
    }
    else {
        arre[(i * numcol) + j] = 0;
        arim[(i * numcol) + j] = 0;
    }
}
index1 = i-yshift; /* shift in y */
if(index1<0) index1+=ysize; /* if out of bounds by y coordinate, wrap */

index2 = j-xshift; /* shift in x */
if(index2<0) index2+=xsize; /* if out of bounds by x coordinate, wrap */

map[i * xsize + j] = index1 * xsize + index2;

#include "stdafx.h"
#include <stdlib.h>
#include <stdio.h>

double *grid(int center, int size, double delta)
{
    double value;
    double *array;
    value = delta * (size*0.5 - 1); /* this is how it is done in labview */
    array = (double *) malloc (size * sizeof(double));
    while (size--) {
        array[size]=value*value;
        value -=delta;
    }
    return array;
}

THE FILE BELOW HAD BEEN UPLOADED FROM THE INTERNET AND ADOPTED FOR THIS PROJECT

/*-------------------------------------*-C-*------------------------------------*
 * File: 
 *   fftn.c 
 *
 * Public: 
 *   fft_free(); 
 *   fftn / fftnf();
 *
 * Private: 
 *   fftradix / fftradixf();
 *
 * Descript: 
 *   multivariate complex Fourier transform, computed in place 
 *   using mixed-radix Fast Fourier Transform algorithm.
 *   
 *   Fortran code by: 
 *   RC Singleton, Stanford Research Institute, Sept. 1968 
 *
 *   translated by f2c (version 19950721).
 *
 * Revisions: 
 * 26 July 95   John Beale 
 *   - added maxf and maxp as parameters to fftradix() 
 *
 * 28 July 95   Mark Olesen <olesen@me.queensu.ca> 
 *   - cleaned-up the Fortran 66 goto spaghetti, only 3 labels remain. 
 *   - added fft_free() to provide some measure of control over 
 *     allocation/deallocation. 
 *   - added fftn() wrapper for multidimensional FFTs 
 *   - use -DFFT_NOFLOAT or -DFFT_NODouble to avoid compiling that 
 *     precision. Note suffix 'f' on the function names indicates 
 *     float precision. 
 */
* - revised documentation

* 31 July 95 Mark Olesen <olesen@me.queensu.ca>
* - added GNU Public License
* - more cleanup
* - define SUN_BROKEN_REALLOC to use malloc() instead of realloc()
  on the first pass through, apparently needed for old libc
* - removed #error directive in favour of some code that simply
  won't compile (generate an error that way)

* 1 Aug 95 Mark Olesen <olesen@me.queensu.ca>
* - define FFT_RADIX4 to only have radix 2, radix 4 transforms
* - made fftradix/fftradixf() static scope, just use fftn()
  instead. If you have good ideas about fixing the factors
  in fftn() please do so.

* 8 Jan 95 mj olesen <olesen@me.queensu.ca>
* - fixed typo's, including one that broke scaling for scaling by
  total number of matrix elements or the square root of same
* - removed unnecessary casts from allocations

=====================================================================*
| NIST Guide to Available Math Software.                           *
| Source for module FFT from package GO.                           *
*=====================================================================

* int fftn (int ndim, const int dims[], REAL Re[], REAL Im[],
  int iSign, double scaling);

* NDIM = the total number dimensions
* DIMS = a vector of array sizes
  if NDIM is zero then DIMS must be zero-terminated

* RE and IM hold the real and imaginary components of the data, and return
  the resulting real and imaginary Fourier coefficients. Multidimensional
  data *must* be allocated contiguously. There is no limit on the number
  of dimensions.

* iSIGN = the sign of the complex exponential (ie, forward or inverse FFT)
  the magnitude of iSIGN (normally 1) is used to determine the
  correct indexing increment (see below).

* SCALING = normalizing constant by which the final result is *divided*
  if SCALING == -1, normalize by total dimension of the transform
  if SCALING < -1, normalize by the square-root of the total dimension

* example:
  tri-variate transform with Re[n1][n2][n3], Im[n1][n2][n3]
  * int dims[3] = (n1,n2,n3)
  * fftn (3, dims, Re, Im, 1, scaling);

* int fftradix (REAL Re[], REAL Im[], size_t nTotal, size_t nPass,
  * size_t nSpan, int iSign, size_t max_factors,
  * size_t max_perm);

* RE, IM - see above documentation

* Although there is no limit on the number of dimensions, fftradix() must
  be called once for each dimension, but the calls may be in any order.

* NTOTAL = the total number of complex data values
* NPASS  = the dimension of the current variable
* NSPAN/NPASS = the spacing of consecutive data values while indexing the
  current variable
* ISIGN - see above documentation
*
* example:
* tri-variate transform with Re[n1][n2][n3], Im[n1][n2][n3]
* ffradix (Re, lm, n1*n2*n3, n1, n1, 1, maxf, maxp);
* ffradix (Re, lm, n1*n2*n3, n2, n1*n2, 1, maxf, maxp);
* ffradix (Re, lm, n1*n2*n3, n3, n1*n2*n3, 1, maxf, maxp);
*
* single-variate transform,
* NTOTAL = N = NSPAN = (number of complex data values),
* ffradix (Re, lm, n, n, n, 1, maxf, maxp);
*
* The data can also be stored in a single array with alternating real and
* imaginary parts, the magnitude of ISIGN is changed to 2 to give correct
* indexing increment, and data [0] and data [1] used to pass the initial
* addresses for the sequences of real and imaginary values,
* example:
* REAL data [2*NTOTAL];
* ffradix ( &data[0], &data[1], NTOTAL, nPass, nSpan, 2, maxf, maxp);
*
* for temporary allocation:
* MAX_FACTORS >= the maximum prime factor of NPASS
* MAX_PERM >= the number of prime factors of NPASS. In addition,
* if the square-free portion K of NPASS has two or more prime
* factors, then MAX_PERM >= (K-1)
* storage in FACTOR for a maximum of 15 prime factors of NPASS, if NPASS
* has more than one square-free factor, the product of the square-free
* factors must be <= 210 array storage for maximum prime factor of 23 the
* following two constants should agree with the array dimensions.
* void fft_free (void);
* free-up allocated temporary storage after finished all the Fourier
* transforms.
*-----------------------------------------------------------------------*/

ifndef _FFTN_C
#define _FFTN_C
#include "stdafx.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "fftn.h"
/* double precision routine */
static int
fftradix (double Re[], double Im[],
        size_t nTotal, size_t nPass, size_t nSpan, int isign,
        int max_factors, int max_perm);

/* float precision routine */
static int
fftradixf (float Re[], float Im[],
        size_t nTotal, size_t nPass, size_t nSpan, int isign,
        int max_factors, int max_perm);

/* parameters for memory management */
static size_t SpaceAlloced = 0;
static size_t MaxPermAlloced = 0;
/* temp space, (void *) since both float and double routines use it */
static void *Tmp0 = NULL; /* temp space for real part */
static void *Tmp1 = NULL; /* temp space for imaginary part */
static void *Tmp2 = NULL; /* temp space for Cosine values */
static void *Tmp3 = NULL; /* temp space for Sine values */
static int  *Perm = NULL; /* Permutation vector */
#define NFACTOR 11
static int factor [NFACTOR];

#if defined (__FILE__) && !defined (lint)
Error: your compiler is sick! define __FILE__ yourself (a string)
eg, something like -D__FILE__="fftn.c"
#endif

#ifndef M_PI
#define M_PI 3.14159265358979323846264338327950288
#endif

#ifndef SIN60
#define SIN60 0.86602540378443865 /* sin(60 deg) */
#define COST72 0.30901699437494742 /* cos(72 deg) */
#define SIN72 0.95105651629515357 /* sin(72 deg) */
#endif

/* re-include this source file on the second pass through */
#undef REAL
#undef FFTN
#undef FFTNS
#undef FFTRADIX
#undef FFTRADIXS

#if define FFT_NOFLOAT
#define REAL  float
#define FFTN  fftnf  /* trailing 'f' for float */
#define FFTNS  "fftnf"  /* name for error message */
#define FFTRADIX fftradixf /* trailing 'f' for float */
#define FFTRADIXS "fftradixf" /* name for error message */
//# include __FILE__ /* include this file again */
#endif

#undef REAL
#undef FFTN
#undef FFTNS
#undef FFTRADIX
#undef FFTRADIXS

#if define FFT_NODOUBLE
#define REAL  double
#define FFTN  fftn
#define FFTNS  "fftn"  /* name for error message */
#define FFTRADIX fftradix /* trailing 'T' for float */
#define FFTRADIXS "fftradix" /* name for error message */
//# include __FILE__ /* include this file again */
#endif

#endif /* _FFTN_C */

*/

void
fft_free (void)
{
    SpaceAlloced = MaxPermAlloced = 0;
}
if (Tmp0 != NULL) { free (Tmp0); Tmp0 = NULL; }
if (Tmp1 != NULL) { free (Tmp1); Tmp1 = NULL; }
if (Tmp2 != NULL) { free (Tmp2); Tmp2 = NULL; }
if (Tmp3 != NULL) { free (Tmp3); Tmp3 = NULL; }
if (Perm != NULL) { free (Perm); Perm = NULL; }
}

int
FFTN (int ndim, const int dims[],
      REAL Re [],
      REAL Im [],
      int iSign,
      double scaling)
{
  size_t nSpan, nPass, nTotal;
  int ret, i, max_factors, max_perm, inSpan;
  /*
  * tally the number of elements in the data array
  * and determine the number of dimensions
  */
  nTotal = 1;
  if (ndim && dims[0])
    {
      for (i = 0; i < ndim; i++)
      {
        if (dims[i] <= 0)
          {
            fputs ("Error: " FFTNS "() - dimension error\n", stderr);
            fft_free (); /* free-up memory */
            return -1;
          }
        nTotal *= dims[i];
      }
    } else
    {
      ndim = 0;
      for (i = 0; dims[i]; i++)
      {
        if (dims[i] <= 0)
          {
            fputs ("Error: " FFTNS "() - dimension error\n", stderr);
            fft_free (); /* free-up memory */
            return -1;
          }
        nTotal *= dims[i];
        ndim++;
      }
    }
  /* determine maximum number of factors and permutations */
  #if 1
  /* follow John Beale's example, just use the largest dimension and don't
  * worry about excess allocation. May be someone else will do it?
  */
  max_factors = max_perm = 1;
  for (i = 0; i < ndim; i++)
    {
      nSpan = dims[i];
      inSpan = (int) nSpan;
      if (inSpan > max_factors) max_factors = inSpan;
      if (inSpan > max_perm) max_perm = inSpan;
    }
  #else
  /* use the constants used in the original Fortran code */
  max_factors = 23;
  max_perm = 209;
  #endif
/* loop over the dimensions: */
nSpan = 1;
for (i = 0; i < ndim; i++)
{
    nPass = dims[i];
nSpan *= nPass;
    ret = FFTRADIX (Re, Im, nTotal, nPass, nSpan, iSign,
               max_factors, max_perm);
    /* exit, clean-up already done */
    if (ret)
        return ret;
}

/* Divide through by the normalizing constant: */
if (scaling && scaling != 1.0)
{
    if (iSign < 0) iSign = -iSign;
    scaling = sqrt (scaling);
    if (scaling < 0.0)
        scaling = (scaling < -1.0) ? sqrt ((double) nTotal) : nTotal;
    scaling = 1.0 / scaling; /* multiply is often faster */
    for (i = 0; i < (int) nTotal; i += iSign)
    {
        Re[i] *= scaling;
        Im[i] *= scaling;
    }
}
return 0;

/*---------------------------------------------------------------*/

/*
 * singleton's mixed radix routine
 *
* could move allocation out to fftn(), but leave it here so that it's
* possible to make this a standalone function
*/
static int
FFTRADIX (REAL Re[],
      REAL Im[],
      size_t nTotal,
      size_t nPass,
      size_t nSpan,
      int iSign,
      int max_factors,
      int max_perm)
{
    int ii, mfactor, kspan, ispan, inc;
    int j, jc, jf, jj, k, k1, k2, k3, k4, kk, kt, nn, ns, nt;

    REAL radf;
    REAL c1, c2, c3, cd, aa, aj, ak, ajm, ajp, akm, akp;
    REAL s1, s2, s3, sd, bb, bj, bk, bjn, bjp, bkm, bkp;
    REAL *Rtmp = NULL; /* temp space for real part*/
    REAL *Itmp = NULL; /* temp space for imaginary part */
    REAL *Cos = NULL; /* Cosine values */
    REAL *Sin = NULL; /* Sine values */
    REAL s60 = SIN60;  /* sin(60 deg) */
    REAL c72 = COS72;  /* cos(72 deg) */
    REAL s72 = SIN72;  /* sin(72 deg) */
    REAL pi2 = M_PI;  /* use PI first, 2 PI later */

    /* gcc complains about k3 being uninitialized, but I can't find out where
     * or why ... it looks okay to me.
     * */
    /* initialize to make gcc happy
     */
k3 = 0;

/* gcc complains about c2, c3, s2, s3 being uninitialized, but they're
 * only used for the radix 4 case and only AFTER the (s1 == 0.0) pass
 * through the loop at which point they will have been calculated.
 * initialize to make gcc happy
 */
c2 = c3 = s2 = s3 = 0.0;

/* Parameter adjustments, was fortran so fix zero-offset */
Re--;
Im--;

if (nPass < 2)
    return 0;

/* allocate storage */
if (SpaceAlloced < max_factors * sizeof (REAL))
{
    #ifdef SUN_BROKEN_REALLOC
    if (!SpaceAlloced) /* first time */
        {
            SpaceAlloced = max_factors * sizeof (REAL);
            Tmp0 = malloc (SpaceAlloced);
            Tmp1 = malloc (SpaceAlloced);
            Tmp2 = malloc (SpaceAlloced);
            Tmp3 = malloc (SpaceAlloced);
        }
    else
        {
        }
    #endif

    SpaceAlloced = max_factors * sizeof (REAL);
    Tmp0 = realloc (Tmp0, SpaceAlloced);
    Tmp1 = realloc (Tmp1, SpaceAlloced);
    Tmp2 = realloc (Tmp2, SpaceAlloced);
    Tmp3 = realloc (Tmp3, SpaceAlloced);
    #ifdef SUN_BROKEN_REALLOC
    }
    #endif
    
    else
    {
        /* allow full use of alloc'd space */
        max_factors = (int) SpaceAlloced / sizeof (REAL);
    }
    if (((int) MaxPermAlloced < max_perm)
{
    #ifdef SUN_BROKEN_REALLOC
    if (!MaxPermAlloced) /* first time */
        Perm = malloc (max_perm * sizeof(int));
    else
    
    #endif
        Perm = (int *) realloc (Perm, max_perm * sizeof(int));
        MaxPermAlloced = max_perm;
    }
    else
    {
        /* allow full use of alloc'd space */
        max_perm = (int) MaxPermAlloced;
    }
    if (Tmp0 == NULL || Tmp1 == NULL || Tmp2 == NULL || Tmp3 == NULL
        || Perm == NULL)
        goto Memory_Error_Label;

    /* assign pointers */
    Rtmp = (REAL *) Tmp0;
    Itmp = (REAL *) Tmp1;
    Cos = (REAL *) Tmp2;
    Sin = (REAL *) Tmp3;
/*
 * Function Body
 */
inc = iSign;
if (iSign < 0) {
    s72 = -s72;
    s60 = -s60;
    pi2 = -pi2;
    inc = -inc; /* absolute value */
}

/* adjust for strange increments */
nt = inc * (int) nTotal;
ns = inc * (int) nSpan;
kspan = ns;
nn = nt - inc;
jc = ns / ((int) nPass);
radf = pi2 * (double) jc;
pi2 *= 2.0; /* use 2 PI from here on */
i = 0;
jf = 0;
/* determine the factors of n */
mfactor = 0;
k = (int) nPass;
while (k % 16 == 0) {
    mfactor++;
    factor[mfactor - 1] = 4;
    k /= 16;
}
j = 3;
jj = 9;
do {
    while (k % jj == 0) {
        mfactor++;
        factor[mfactor - 1] = j;
    }
j += 2;
jj = j * j;
} while (jj <= k);
if (k <= 4) {
    kt = mfactor;
j = 2;
do {
        if (k % j == 0) {
            mfactor++;
            factor[mfactor - 1] = j;
        }
j = ((j + 1) / 2 << 1) + 1;
} while (j <= k);
} else {
    if (k - (k / 4 << 2) == 0) {
        mfactor++;
        factor[mfactor - 1] = 2;
        k /= 4;
    }
    kt = mfactor;
}
if (kt) {
    j = kt;
do {
        mfactor++;
        factor[mfactor - 1] = factor[j - 1];
        j = ((j + 1) / 2 << 1) + 1;
    } while (j <= k);
}
j--;
} while (!j);
}

/* test that mfactors is in range */
if (mfactor > NFACTOR)
{
    fputs ("Error: " FFTRADIXS "() - exceeded number of factors\n", stderr);
    goto Memory_Error_Label;
}

/* compute fourier transform */
for (;;) {
    sd = radf / (double) kspan;
    cd = sin(sd);
    cd = 2.0 * cd * cd;
    sd = sin(sd + sd);
    kk = 1;
    ii++;
    switch (factor [ii - 1]) {
        case 2:
            /* transform for factor of 2 (including rotation factor) */
            kspan /= 2;
            k1 = kspan + 2;
            do {
                do {
                    k2 = kk + kspan;
                    ak = Re [k2];
                    bk = Im [k2];
                    Re [k2] = Re [kk] - ak;
                    Im [k2] = Im [kk] - bk;
                    Re [kk] += ak;
                    Im [kk] += bk;
                    kk = k2 + kspan;
                } while (kk <= nn);
                kk -= nn;
            } while (kk <= jc);
            if (kk > kspan)
                goto Permute_Results_Label; /* exit infinite loop */
            do {
                c1 = 1.0 - cd;
                s1 = sd;
                do {
                    do {
                        k2 = kk + kspan;
                        ak = Re [kk] - Re [k2];
                        bk = Im [kk] - Im [k2];
                        Re [kk] += Re [k2];
                        Im [kk] += Im [k2];
                        Re [k2] = c1 * ak - s1 * bk;
                        Im [k2] = s1 * ak + c1 * bk;
                        kk = k2 + kspan;
                    } while (kk < nt);
                    k2 = kk - nt;
                    c1 = -c1;
                    kk = k1 - k2;
                } while (kk > k2);
                ak = c1 - (cd * c1 + sd * s1);
                s1 = sd * c1 - cd * s1 + s1;
                c1 = 2.0 - (ak * ak + s1 * s1);
                s1 *= c1;
                c1 *= ak;
                kk += jc;
            } while (kk < k2);
            k1 += inc + inc;
            kk = (k1 - kspan) / 2 + jc;
        } while (kk <= jc + jc);
        break;
case 4: /* transform for factor of 4 */

    isize = kspan;
    kspan /= 4;

    do {
        c1 = 1.0;
        s1 = 0.0;
        do {
            do {
                k1 = kk + kspan;
                k2 = k1 + kspan;
                k3 = k2 + kspan;
                akp = Re[kk] + Re[k2];
                akm = Re[kk] - Re[k2];
                ajp = Re[k1] + Re[k3];
                ajm = Re[k1] - Re[k3];
                bkp = Im[kk] + Im[k2];
                bkm = Im[kk] - Im[k2];
                bjp = Im[k1] + Im[k3];
                bjm = Im[k1] - Im[k3];
                Re[kk] = akp + ajp;
                Im[kk] = bkp + bjp;
                ajp = akp - ajp;
                bjp = bkp - bjp;
                if (iSign < 0) {
                    akp = akm + bjm;
                    bkp = bkm - ajm;
                    akm -= bjm;
                    bkm += ajm;
                } else {
                    akp = akm - bjm;
                    bkp = bkm + ajm;
                    akm += bjm;
                    bkm -= ajm;
                }
                /* avoid useless multiplies */
                if (s1 == 0.0) {
                    Re[k1] = akp;
                    Re[k2] = ajp;
                    Re[k3] = akm;
                    Im[k1] = bkp;
                    Im[k2] = bjp;
                    Im[k3] = bkm;
                } else {
                    Re[k1] = akp * c1 - bkp * s1;
                    Re[k2] = ajp * c2 - bjp * s2;
                    Re[k3] = akm * c3 - bkm * s3;
                    Im[k1] = akp * s1 + bkp * c1;
                    Im[k2] = ajp * s2 + bjp * c2;
                    Im[k3] = akm * s3 + bkm * c3;
                }
                kk = k3 + kspan;
            } while (kk <= nt);
            c2 = c1 - (cd * c1 + sd * s1);
            s1 = sd * c1 - cd * s1 + s1;
            c1 = 2.0 - (c2 * c2 + s1 * s1);
            s1 *= c1;
            c1 *= c2;
        } /* values of c2, c3, s2, s3 that will get used next time */
        c2 = c1 * c1 - s1 * s1;
        s2 = 2.0 * c1 * s1;
        c3 = c2 * c1 - s2 * s1;
        s3 = c2 * s1 + s2 * c1;
        kk = kk - nt + jc;
    } while (kk <= jc);
    if (kspan == jc)
goto Permute_Results_Label;  /* exit infinite loop */
break;

default:
  /* transform for odd factors */
#endif FFT_RADIX4
fputs ("Error: " FFTRADIXS "(): compiled for radix 2/4 only/in", stderr);
fft_free ();  /* free-up memory */
return -1;
break;
#else /* FFT_RADIX4 */
k = factor [ii - 1];
ispan = kspan;
kspan /= k;

switch (k) {
case 3:  /* transform for factor of 3 (optional code) */
do {
  do {
    k1 = kk + kspan;
k2 = k1 + kspan;
    ak = Re [kk];
bk = Im [kk];
    aj = Re [k1] + Re [k2];
bj = Im [k1] + Im [k2];
    Re [kk] = ak + aj;
    Im [kk] = bk + bj;
    ak = 0.5 * aj;
bk = 0.5 * bj;
    aj = (Re [k1] - Re [k2]) * s60;
bj = (Im [k1] - Im [k2]) * s60;
    Re [k1] = ak - bj;
    Re [k2] = ak + bj;
    Im [k1] = bk + aj;
    Im [k2] = bk - aj;
  } while (kk < nn);
  kk -= nn;
} while (kk <= kspan);
break;
case 5:  /* transform for factor of 5 (optional code) */
c2 = c72 * c72 - s72 * s72;
s2 = 2.0 * c72 * s72;
do {
  do {
    k1 = kk + kspan;
k2 = k1 + kspan;
k3 = k2 + kspan;
k4 = k3 + kspan;
akp = Re [k1] + Re [k4];
  akm = Re [k1] - Re [k4];
bkp = Im [k1] + Im [k4];
bkm = Im [k1] - Im [k4];
    aip = Re [k2] + Re [k3];
aim = Re [k2] - Re [k3];
bjp = Im [k2] + Im [k3];
bjm = Im [k2] - Im [k3];
aa = Re [kk];
bb = Im [kk];
    Re [kk] = aa + akp + aip;
    Im [kk] = bb + bkp + bjp;
    ak = akp * c72 + aip * c2 + aa;
bk = bkp * c72 + bjp * c2 + bb;
    aj = akm * s72 + aim * s2;
bj = bkm * s72 + bjm * s2;
    Re [k1] = ak - bj;
    Re [k4] = ak + bj;
    Im [k1] = bk + aj;
    Im [k4] = bk - aj;
  } while (kk < nn);
  kk -= nn;
} while (kk <= kspan);
break;
}
ak = akp \times c2 + ajp \times c72 + aa;
bk = bkp \times c2 + bjp \times c72 + bb;
aj = akm \times s2 - ajm \times s72;
bj = bkm \times s2 - bjm \times s72;
Re [k2] = ak - bj;
Re [k3] = ak + bj;
Im [k2] = bk + aj;
Im [k3] = bk - aj;
kk = k4 + kspan;
\}
while (kk < nn);
kk = nn;
\}
while (kk <= kspan);
break;
default:
if (k != jf) {
   jf = k;
s1 = pi2 / (double) k;
c1 = cos(s1);
s1 = sin(s1);
if (jf > max_factors)
go to Memory_Error_Label;
Cos [jf - 1] = 1.0;
Sin [jf - 1] = 0.0;
j = 1;
do { Cos [j - 1] = Cos [k - 1] \times c1 + Sin [k - 1] \times s1;
   Sin [j - 1] = Cos [k - 1] \times s1 - Sin [k - 1] \times c1;
k--;
   Cos [k - 1] = Cos [j - 1];
   Sin [k - 1] = -Sin [j - 1];
   j++;
} while (j < k);
do {
do {
k1 = kk;
k2 = kk + ispan;
ak = aa = Re [kk];
bk = bb = Im [kk];
j = 1;
k1 += kspan;
do {
k2 = kspan;
j++;
   Rtmp [j - 1] = Re [k1] + Re [k2];
   ak += Rtmp [j - 1];
   Itmp [j - 1] = Im [k1] + Im [k2];
   bk += Itmp [j - 1];
   j++;
   Rtmp [j - 1] = Re [k1] - Re [k2];
   Itmp [j - 1] = Im [k1] - Im [k2];
   k1 += kspan;
} while (k1 < k2);
Re [kk] = ak;
Im [kk] = bk;
k1 = kk;
k2 = kk + ispan;
j = 1;
do {
k1 += kspan;
k2 = kspan;
j = j;
ak = aa;
bk = bb;
aj = 0.0;
bj = 0.0;
k = 1;
do {
k++;
}}
ak += Rtmp [k - 1] * Cos [jj - 1];
bk += Itmp [k - 1] * Cos [jj - 1];
k++;
aj += Rtmp [k - 1] * Sin [jj - 1];
bj += Itmp [k - 1] * Sin [jj - 1];
jj += j;
if (jj > jf) {
    jj -= jf;
}
}

while (k < jf);
k = jf - 1;
Re [k1] = ak - bj;
Im [k1] = bk + aj;
Re [k2] = ak + bj;
Im [k2] = bk - aj;
}
++;
while (j < k);
kk += ispan;
}
while (kk <= nn);
kk = nn;
}
while (kk <= kspan);
break;
}

/* multiply by rotation factor (except for factors of 2 and 4) */
if (ii == mfactor)
goto Permute_Results_Label;  /* exit infinite loop */
kk = jc + 1;
do {
    c2 = 1.0 - cd;
s1 = sd;
do {
        c1 = c2;
s2 = s1;
kk += kspan;
do {
            ak = Re [kk];
            Re [kk] = c2 * ak - s2 * Im [kk];
            Im [kk] = s2 * ak + c2 * Im [kk];
            kk += ispan;
        }
while (kk <= nt);
    ak = s1 * s2;
s2 = s1 * c2 + c1 * s2;
c2 = c1 * c2 - ak;
kk = kk - nt + kspan;
}
while (kk <= ispan);
c2 = c1 - (cd * c1 + sd * s1);
s1 += sd * c1 - cd * s1;
c1 = 2.0 - (c2 * c2 + s1 * s1);
s1 *= c1;
c2 *= c1;
kk = kk - ispan + jc;
}
while (kk <= kspan);
kk = kk - kspan + jc + inc;
}
while (kk <= jc + jc);
break;
#endif /* FFT_RADIX4 */
}

/* permute the results to normal order---done in two stages */
/* permutation for square factors of n */
Permute_Results_Label:
Perm [0] = ns;
if (kt) {
    k = kt + kt + 1;
    if (mfactor < k)
        k--;
    j = 1;
    Perm [k] = jc;
}
\begin{verbatim}
do {
    Perm [j] = Perm [j - 1] / factor [j - 1];
    Perm [k - 1] = Perm [k] * factor [j - 1];
    j++;
    k--;
} while (j < k);

k3 = Perm [k];
kspan = Perm [1];
kk = j + 1;
k2 = kspan + 1;
j = 1;
if (nPass != nTotal) {
    /* permutation for multivariate transform */
    Permute_Multi_Label:
    do {
        do {
            k = kk + jc;
            do {
                /* swap Re[kk] <> Re[k2], Im[kk] <> Im[k2] */
                ak = Re[kk]; Re[kk] = Re[k2]; Re[k2] = ak;
                bk = Im[kk]; Im[kk] = Im[k2]; Im[k2] = bk;
                kk += inc;
                k2 += inc;
            } while (kk < k);
            kk += ns - jc;
            k2 += ns - jc;
        } while (kk < nt);
        k2 = k2 - nt + kspan;
        kk = kk - nt + jc;
    } while (k2 < ns);
    do {
        do {
            k2 -= Perm[j - 1];
            j++;
            k2 = Perm[j] + k2;
        } while (k2 > Perm[j - 1]);
        j = 1;
        do {
            if (kk < k2)
goto Permute_Multi_Label;
            kk += jc;
            k2 += kspan;
        } while (k2 < ns);
    } while (kk < ns);
} else {
    /* permutation for single-variate transform (optional code) */
    Permute_Single_Label:
    do {
        /* swap Re[kk] <> Re[k2], Im[kk] <> Im[k2] */
        ak = Re[kk]; Re[kk] = Re[k2]; Re[k2] = ak;
        bk = Im[kk]; Im[kk] = Im[k2]; Im[k2] = bk;
        kk += inc;
        k2 += kspan;
    } while (k2 < ns);
    do {
        do {
            k2 -= Perm[j - 1];
            j++;
            k2 = Perm[j] + k2;
        } while (k2 > Perm[j - 1]);
        j = 1;
        do {
            if (kk < k2)
goto Permute_Single_Label;
            kk += inc;
            k2 += kspan;
        } while (k2 < ns);
    } while (kk < ns);
}

jc = k3;
\end{verbatim}
if ((kt << 1) + 1 >= mfactor)
  return 0;
ispam = Perm [kt];
/* permutation for square-free factors of n */
j = mfactor - kt;
factor [j] = 1;
do {
factor [j - 1] *= factor [j];
j--;
} while (j != kt);
kt++;
nn = factor [kt - 1] - 1;
if (nn > max_perm)
goto Memory_Error_Label;
j = jj = 0;
for (;;) {
k = kt + 1;
k2 = factor [kt - 1];
kk = factor [k - 1];
++;
if (j > nn)
  break;    /* exit infinite loop */
jj += kk;
while (jj >= k2) {
  jj -= k2;
k2 = kk;
k++;
  kk = factor [k - 1];
  jj += kk;
}
Perm [j - 1] = jj;
}
/* determine the permutation cycles of length greater than 1 */
j = 0;
for (;;) {
do {
  j++;
  kk = Perm [j - 1];
} while (kk < 0);
if (kk != j) {
do {
    k = kk;
k2 = Perm [k - 1];
    Perm [k - 1] = -kk;
  } while (kk != j);
k3 = kk;
} else {
  Perm [j - 1] = -j;
  if (j == nn)
    break;    /* exit infinite loop */
}
}
max_factors *= inc;
/* reorder a and b, following the permutation cycles */
for (;;) {
j = k3 + 1;
nt = ispan;
i = nt - inc + 1;
if (nt < 0)
  break;    /* exit infinite loop */
do {
    do {
        j--;
    } while (Perm [j - 1] < 0);
    jj = jc;
do {
        kspan = jj;
        if (jj > max_factors) {
kspan = max_factors;
}
jj -= kspan;
k = Perm [j - 1];
kk = jc * k + ii + jj;
k1 = kk + kspan;
k2 = 0;
do {
    k2++;  
    Rtmp [k2 - 1] = Re [k1];
    Itmp [k2 - 1] = Im [k1];
    k1 -= inc;
} while (k1 != kk);
do {
    k1 = kk + kspan;
    k2 = k1 + jc * (k + Perm [k - 1]);
    k = -Perm [k - 1];
do {
        Re [k1] = Re [k2];
        Im [k1] = Im [k2];
        k1 -= inc;
        k2 -= inc;
    } while (k1 != kk);
    kk = k2;
} while (k != j);
}

k1 = kk + kspan;
k2 = 0;
do {
    k2++;  
    Re [k1] = Rtmp [k2 - 1];
    Im [k1] = Itmp [k2 - 1];
    k1 -= inc;
} while (k1 != kk);
}

return 0;  /* exit point here */

/* alloc or other problem, do some clean-up */

Memory_Error_Label:
    puts("Error: " FFTRADIXS () - insufficient memory, in", stderr);
    fft_free ();  /* free-up memory */
    return -1;
}
#endif /* _FFTN_C */

/*--------------------------------*-C-*---------------------------------*
 * File:  
 * fftn.h  
 * ---------------------------------------------------------------------*
 * Re[]: real value array 
 * Im[]: imaginary value array 
 * nTotal: total number of complex values 
 * nPass: number of elements involved in this pass of transform 
 * nSpan: nspan/nPass = number of bytes to increment pointer 
 *        in Re[] and Im[] 
 * isign: exponent: +1 = forward -1 = reverse 
 * scaling: normalizing constant by which the final result is *divided* 
 *        scaling = -1, normalize by total dimension of the transform 
 *        scaling < -1, normalize by the square-root of the total dimension 
 *     */

#ifndef _FFTN_H
#define _FFTN_H

extern void fft_free (void);

#endif _FFTN_H

/*---------------------------------------------.*.C.*---------------------------------------------*/

/* See the comments in the code for correct usage! */

#define _FFTN_H

extern void fft_free (void);
extern int fftn (int ndim, const int dims[], double Re[], double Im[],
        int isign, double scaling);

extern int fftnf (int ndim, const int dims[], float Re[], float Im[],
        int isign, double scaling);

#endif /* _FFTN_H */
APPENDIX G

LIST OF PUBLICATIONS

The results of this dissertation have been published in the following journals and conference proceedings:


Dimensional Imaging (DH), Technical Digest (CD), (Optical Society of America, 2008), paper DMB5.


About the Author

Alexander Khmaladze received a Master of Science degree in Physics from the University of South Carolina in Columbia, SC in 2000. In 2003 he entered the PhD program in Applied Physics at the University of South Florida, where he worked at Digital Holography & Microscopy Laboratory with Professor M. K. Kim.