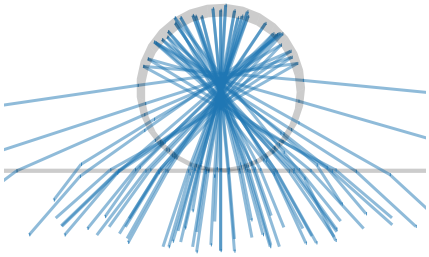
Ray Tracing Simulation of Optical Coherence Tomography to Study Subnanometre Expansion of Active Neurons

Bachelor Thesis
Physics and Astronomy

Sjoerd Seinhorst



Supervised by:
Pegah Asgari
Gerhard Blab
Allard Mosk

Utrecht University

Department of Physics

June 2021

Abstract

It has long been know that when neurons transmit a signal, they expand slightly.[7] The mechanisms behind this are however not yet understood. In an effort to learn more about the functioning of neurons, we want to observe the neurons them and record the expansion in real time. The expansion is around the size of a nanometre, and the signal only takes milliseconds to pass. In addition, the method of observation must not harm the neuron as they must be alive to transmit signals. Optical coherence tomography is has many properties that make it ideal for this task. But the downside of standard OCT it that it has a resolution of micrometres, not nanometres as required. In this Thesis I perform simulations of the OCT process based on ray tracing and show that it is possible to extract these subnanometre diameter changes in the ideal scenario.

Contents

1 Introduction

Optical Coherence Tomography (OCT) has long been used in the medical industry for non-invasive imaging of translucent media like skin and eyes. An OCT scan can penetrate several millimetres into these media and make 2D or 3D images to reveal the structures within. It is able to do so with a resolution down to the micrometre level. Since it relies on the scattering of harmless infrared radiation, this can be done *in vivo*: without needing to extract tissue from the body.[2]

Neurons are known to deform and expand when they transmit a signal.[7] The expansion is on the scale of a nanometre: a billionth of a metre, and a thousandth of the diameter of the neuron. The exact reason for this deformation is not known. To be able to learn more about this phenomenon we want to precisely measure the expansion. But the neuron must remain alive during the measurement process for it to be able to transmit the signal. The signal has a duration of around a millisecond so the measurement process must also be faster than this.

OCT fulfills many of these criteria. The infrared radiation does not harm it. The penetration depth would allow it to measure neurons inside small organisms. The use of visible or infrared light allows for simple high speed signal detection . But the major downside of traditional OCT for our use case is the resolution. The micrometre scale resolution of this process is thousandfold higher than the nanometre resolution needed to measure the expansion of the neuron. We must rely on different data processing methods to extract this data from the OCT scan.

Specifically, in section 2 will be explained how the light spectrum is sinusoidal, and how we can use the phase of this wave to find a change in diameter. When performing an OCT scan on a neuron this phase is nearly proportional to the diameter of the neuron and can be accurately measured. One of the main goals of this project is to find how much the phase changes when the diameter of the neuron changes. This number will be needed for calculating the diameter change from the measured phase in the real experiment. It is referred to as the phase slope or the sensitivity.

Performing experiments with such a precise OCT set-up and biological samples can be tedious and time consuming. Being able to simulate these experiments can be very advantageous. Setting up a simulation takes little preparation. A series of simulations can be run much faster and cheaper than a series of experiments. Simulations can also be perfectly repeatable, where this can be an issue for biological samples. We can simplify, break up and inspect many parts of the simulation and much more data can be acquired than by one or a few physical sensors.

There are two common types of light transport simulations. The first type aims to simulate the full electromagnetic field. This type is very accurate but requires detailed knowledge of the set-up and takes a long time to run.[4] We choose to do a ray tracing based simulation as it is much simpler and faster, yet it can produce very usable results. Section 4 elaborates on this choice.

2 What is OCT?

To give an intuition about OCT we can compare it to ultrasound imaging. In an ultrasound, a probe emits a pulse of ultrasonic sound waves. The sound travels through the medium at the speed of sound and after some time a portion of the sound reflects off something. This reflection takes some more time to return to the probe where it is measured again. The time delay between the emission and detection of the signal tells us how far from the probe the reflection occurred.[2, 1]

In OCT we use light. But we still measure the time light takes to echo back through reflections or scattering. One of the advantages of light is that its smaller wavelength allows us to measure

smaller features. Light is so much faster than sound that it would be infeasible to directly measure the travel time over the resolutions we hope to measure. In a vacuum, light would travel 1 µm in 3.336 fs. That is 3.336×10^{-15} s. There are as many of those durations in a second as there are seconds in 9.5 million years. This would be the diameter of a neuron, but we are trying to detect size changes more than a thousand times smaller than this. So instead we must rely on interference of light to give depth information.

2.1 Interferometry

When two monochromatic rays of light combine they may do so with a different relative phase. If the two beams are in phase to each other, the beams combine constructively and the resulting beam is brighter than either incoming beam. If the two beams are out of phase to each other, the beams combine destructively and the resulting beam is dimmer than either incoming beam. This is called interference of light.[3]

We can proceed to build a Michelson interferometer. This interferometer is comprised of a monochromatic light source aimed at a beam splitter which equally splits the light into two arms. At the end of each arm is a mirror which reflects the light back into the beam splitter. The beam splitter recombines the light and sends it to a detector. We define the length of an arm L as the distance from the beam splitter to the mirror and back. If each arm is the exact same length, the beam would travel the exact same distance in each arm and when recombined, the light would be in phase. It results in constructive interference so we see a signal at the detector. If however we move one mirror to be half of a wavelength longer than the other, the light would be out of phase resulting in destructive interference and no output. But if this arm's length is increased by another half wavelength, the beams are in phase again and we get a signal.[2, 1, 3] The intensity of light is the square of the electric field amplitude. When the two beams from the two arms recombine, the resulting intensity is

$$I = |E_1|^2 + |E_2|^2 + 2|E_1||E_2| \cos(2k\,\Delta L) \tag{1}$$

where E_1 and E_2 are the electric field amplitudes from either arm of the interferometer, $k = 2\pi/\lambda$ is the wave number of the light and ΔL is the path length difference in the two arms [2].

The goal for this interferometer is to replace the stationary mirror by our sample, and use the moving mirror to scan where the light is reflecting in the sample. Through interference of monochromatic light we are unable to extract this depth information since moving the mirror just results in an oscillating signal. One solution is to use a low-coherent light source instead, which consists of not one but a range of wavelengths. When both arms are equal in length, all wavelengths are in phase and we get an output signal. But when the arms are not, the beams won't be in phase and we will not receive any signal. We can only receive a signal when the length that the light travels in each arm is almost exactly the same. So by scanning the moving mirror and keeping the sample stationary, we can measure at what depth we get reflections from within the sample. This is called an A-scan (axial scan).

This approach will however not work for us. Not only would the mirror and detector have to run at very high rates, the light itself is limiting the resolution. When the two arms of our interferometer are exactly the same length, we get constructive interference as explained above. But when the lengths mismatch only slightly, depending on the bandwidth of of the light source, there remains some level of constructive interference. The length scale over which significant constructive interference can occur is equal to the coherence length of the spectrum. This is on the order of 1 µm. So we cannot directly measure the nanometre scale diameter changes of the neuron in this way.[2]

We must extract more of the available information from the light and we do so by measuring the full spectrum of the return signal, as opposed to only its intensity. For nonmonochromatic light like low-coherent light, the intensity of eq. (1) actually becomes a spectrum of light intensity over wave number. The path length difference can be recovered by taking a Fourier transform of this spectrum. Each different value of ΔL produces a cosine with a different k-space frequency. The Fourier transform decomposes a function into component sinusoidal waves, each with a different k-space frequency, and gives their amplitude and phase offset. The k-space frequency, or Fourier dual of the wave number, is real space distance x. The Fourier transform of a function of k produces a complex function which is over x, where the magnitude of the complex number at some value for x is the amplitude of the wave with frequency x and the argument of the complex number is the phase of the wave. When we take the Fourier transform of $I(k)$ we get a function with two peaks: one peak at a zero distance, for the constant (zero frequency) offset caused by $|E_1|^2 + |E_2|^2$. We also get a peak at a distance of $2\,\Delta L$ from the $\cos(2k\,\Delta L)$ term. From this extracting ΔL is trivial. This pattern of peaks is the same A-scan as obtained with the previous method.

2.2 Parallel plates

Let's now change the setup a bit. Instead of two arms with each a mirror, we have just one arm. On the end of this arm are two thin plates, which reflect part of the light and transmit the rest. The distance between the plates is d and let's also say there is a medium in between the plates with a refractive index n. This is very similar to a Michaelson interferometer in that we receive one high intensity reflection from each plate. The light bounces between the plates which results in multiple reflections returning. But if the reflectivity of the plates is low, the returned spectrum can be well approximated as a constant offset plus a cosine term: $\cos(2ndk)$. We can repeat the Fourier transform process to find the optical path length (OPL) between the plates nd.

But we already determined that this process alone does not let us get any better than around 1 µm of resolution. For that we need to look at the phase. But to derive what this phase is exactly we must dissect the Fourier transform. The Fourier transform pairs up the signal with a reference wave. Specifically, the signal is multiplied by the complex function $\mathrm{e}^{-\mathrm{i}kx}$. The result is a complex function whose phase is the phase difference between the reference wave and the signal. The Fourier transform then integrates this product over k and effectively averages out the phase difference along the full spectrum to get the final phase. In this case the phase of the reference wave is always zero at $k = 0$, so the signal's phase returned is also at $k = 0$.

The discrete Fourier transform, which will be discussed in section 5 and actually be used for finding the phase, has the additional restriction that the reference wave must have the phase be zero at the start and end of the spectrum. The frequency of the reference wave can no longer exactly match the frequency of the spectrum, but we will end up choosing the closest available frequency. The difference in the number of oscillations between the start and the end of the spectrum is never more than half a wavelength. At the start of the spectrum, the phase difference between the signal and the reference is equal to the signal's phase here, because the reference wave has by definition zero phase here. For the same reason, the phase difference at the end of the spectrum is equal to the signal's phase here. The phase difference between these two points changes linearly, plus an oscillating term because the signal is only real valued while the reference wave is complex. And since the frequencies/wavelengths of the two waves match as closely as possible, the phase difference between the two waves will never be more than π. The Fourier transform again averages out the phase difference, where the oscillating term disappears, and linear phase averages out to the mean of the beginning and end phases.

For the two plate spectrum $\cos(2ndk)$ that runs from wave numbers $k_{\min}$ to $k_{\max}$, the phase at the start is $2ndk_{\min}$ and the phase at the end is $2ndk_{\max}$. The phase that we get from the discrete Fourier transform is then $2nd\ (k_{\min} + k_{\max})/2 = nd(k_{\min} + k_{\max})$. The phase slope, or sensitivity, or change in phase over diameter, is then

$$n(k_{\min} + k_{\max}). \tag{2}$$

3 Set-up

For simulating an OCT of a neuron we aim for a simulated set-up that is similar to the laboratory set-up. The set-up that this simulation is based on uses an inverted microscope where the objective is below the microscope slide. The space between the objective and the slide is filled with immersion oil which has the same refractive index as glass. This removes any reflection or refraction of light in this area. On top of the microscope slide is a neuron surrounded in water. The glass–water interface is represented as an infinite plane, and the neuron is represented as an infinite cylinder. For the glass we used a refractive index of 1.52, for water 1.33 and for the neuron we used 1.39. We assume there to be a 5 nm gap of water between the neuron and the glass slide. The microscope objective has 20× magnification, a focus distance of 300 µm and a numerical aperture of 1.2. The incoming light has a uniform (flat) spectrum with wavelengths from 650 nm to 900 nm. The beam initially consists of 3000 rays, but this may increase as the simulation progresses. After 11 intersections, at each of which the ray may reflect and be transmitted, the simulation is stopped. The amplitude of the remaining rays is then almost zero.

4 Ray Tracing Simulation

The first part of the procedure is the simulation of light in the set-up. Light consists of fundamental particles called photons. Quantum mechanics tells us that light can behave in two distinct ways: either like waves or like particles. The same distinction is present in light simulation software.

MIT's Meep [4], for example, simulates the full electromagnetic field, and with that the waves that enable propagation of light. It divides a finite area or volume of space into a grid where for each point the electric and magnetic field components are considered. Time advances in discrete steps, for each of which the electromagnetic field evolves according to Maxwell's equations. This approach gives a very complete picture of the field. It can simulate the behaviour of light in complex materials such as arbitrary anisotropic, non-linear, and dispersive electric and magnetic media. It also enables simulation of more involved phenomena like Cerenkov radiation. The main drawback of this approach over ray tracing is the simulation runtime. While considerable effort has been put into performance and parallelization of Meep, the runtime can be on the order of hours. This slows down the research process and can be expensive.

Ray tracing is more similar to the particle nature of light. One can imagine a ray as an arbitrarily narrow bundle of perfectly parallel light. It travels in a straight line until it intersects with an object in the scene. There it may get absorbed, transmitted and/or reflected, depending on the material. After several of such intersections, also called *bounces*, the ray hits a detector where the relevant information is collected. This technique is popular for the creation of computer generated imagery, especially when photorealism is desired. An advanced implementation of this style of light transport simulation can be found in Blender's Cycles render engine. The technique is used in marketing imagery, movies and in recent years increasingly as part of the rendering pipeline in games. The last case especially attests to the performance benefits this approach

offers, as render times generally need to be well under 20 ms. Our simulation won't be quite as performant but with the simple scene, render times stay well under a second. There is plenty of room for optimization with better memory management, algorithms or parallelization, by using techniques such as path tracing, metropolis light transport or a GPU, or switching to a faster programming language like C++. This is a major advantage over a field-based approach, and as such we implemented a ray tracing algorithm.

The purpose of our ray tracing simulation is not to produce an image though, but only an interference spectrum. For simplicity we disregard absorption by the materials as well as dispersion. This means that the spectrum of the ray is not altered by the objects in the scene, only the total intensity is. It also means that the path travelled does not depend on the wavelength of the light. We can conclude that the ray tracing is not affected by the spectrum of the light rays, nor is the detection. We do not need to keep track of the spectrum or wavelength of each ray, only the amplitude of the light. We also do not need to perform the simulation for each wavelength of light if we can find the phase of the light upon detection for all wavelengths.

This can be achieved by tracking the optical path length of each ray's path. The optical path length Λ is the product of the distance a ray travels in a medium d and the refractive index of that medium n. $\Lambda = nd$. Two light rays with the same wavelength that traverse the same optical path length, will experience the same phase change. Since the refractive index of a vacuum is one, this is the same phase change as experienced by a ray that traverses the optical path length as a physical distance in a vacuum. The phase change $\Delta\phi$ over some OPL is $\Delta\phi = \Lambda k$. If we know the phase of the ray when it is generated and the OPL travelled to the detector, we can figure out the phase of the light at the detector. We can multiply the optical path length with many different wave numbers without redoing the ray tracing. Now we know we can perform the ray tracing simulation just once for a fixed scene geometry and find the interference spectrum of the returned light.[3]

But rays and waves behave inherently different from each other. There are phenomena that result from the wave like nature of light that a simple ray tracing simulation cannot accurately reproduce. A well-known example is the diffraction limit.[3] Even in a perfect setup, focused waves will always produce a diffraction pattern and can therefore never focus to a point. In our setup the smallest focus radius that can be reached is around 0.3 to 0.4 µm depending on the wavelength. This is the smallest distinguishable distance between two objects if we were to take a photograph. It is on the order of the radius of a neuron. Even though OCT is not directly limited by this phenomenon it is reasonable to assume that it will affect the results unfavourably. In a ray tracing simulation this could be approximated by purposefully defocussing the rays. Nevertheless we have chosen for simplicity in simulation and interpretation to not include this, and let the rays focus to a perfect point (if there were not obstacles).

4.1 Ray Generation

Light is always emitted from some kind of source. In the physical experiment the light is guided to the sample by a series of fibre optic wires, mirrors and lenses. Their arrangement can differ per set-up and it would be infeasible to trace the light rays through all these elements. Instead we make some basic assumptions about how an ideal set-up would deliver the light to the sample.

1. The beam has a circular cross section.
2. The distribution of light over the beam cross section is homogeneous and isotropic.
3. The light rays focus to a point.
4. The light is in phase at the focus.

Assumption 4 must follow from 3 in the wave picture of light since the propagation direction is always perpendicular to the wave front. The light rays approach the focus radially which means the wave fronts must be concentric spheres around the focus. When such a spherical wave front finally collapses in the focus, the wave front becomes a point which indicates that at that point all light is in phase. 3 is assumed because we want as much of the sampled light to have reflected off our sample neuron. We assume a focus to a point because we assume an ideal optical set-up without set-up–dependent aberrations. And for simplicity we also don't approximate the diffraction limit effects.

For the beam profile either a Gaussian or a uniform round distribution would be a natural choice. The Gaussian profile would be more accurate for a beam near the focus due to the wave behaviour of light. But we decided to use the uniform (homogeneous and isotropic) profile which includes the well-defined beam edge that the objective lens provides. Either way we suspect the choice would not significantly impact the final results.

With these assumptions we need only minimal information to produce the light rays. Specifically, we only need to know the aperture and the focal length of the objective lens from which the rays emerge. These two numbers may even be combined into a single number: the angle of the cone of rays.

We generate the rays on the planar surface of the objective lens. But we need to decide where to position the rays on this plane. A square or hexagonal grid pattern would be a good choice if the only goal was homogeneity. But direction along which such a grid is placed and alignment with the sample may affect the final outcome. We do not want this: we want an isotropic placement. Simply scattering the rays randomly over the surface of the lens would provide this and, with enough rays, would also be homogeneous. But this technique also has its downsides. Namely we lose local homogeneity: there may be clumps of rays in some places and large gaps between rays in others. The irregularity may also introduce unwanted noise. Although these downsides are minor, there is a better solution.

Helmut Vogel in 1978 found that the placement of seeds on the head of a sunflower is homogeneous and isotropic, in a regular pattern.[6] He derived the following spiral. The angle around the midpoint between one ray (or sunflower seed) and the next, in order of their radius, is equal to the golden ratio $\phi = (\sqrt{5}-1)/2 \approx 0.618$ revolutions. The radius is proportional to the square root of the number of revolutions. This is also known as a Fermat spiral or cyclotron spiral. So if we have N rays which are to be placed on a disk with radius R, the j^{th} ray, where $j = 0, 1, 2, \ldots, N-1$, is placed in polar coordinates at

$$\theta_j = 2\pi j\phi, \quad r_j = R\sqrt{\frac{j}{N-1}}. \tag{3}$$

This results in a disk-shaped regular pattern of rays that have, as Vogel calls it, the "most uniform angular distribution (…), and most uniform area distribution" of rays. The pattern is displayed in fig. 1

Now that we know where the rays' starting locations, their other properties are easy to derive. The direction of travel is to the focus. So for the direction vector we subtract the focus location from the starting location and normalise it. The rays must be in phase at the focus. This means that the optical path length travelled must be the same for all rays. We define the optical path length to be zero at the focus. So to get the optical path length of the rays at their generation locations, we calculate the optical path length from there to the focus (distance times refractive index) and subtract it from zero. So for a ray that is 300 µm away from the focus in a medium with a refractive index of 1.5, the optical path length is −450 µm. Each ray also gets a unit amplitude. A different amplitude such as $1/N$ would of course also work.

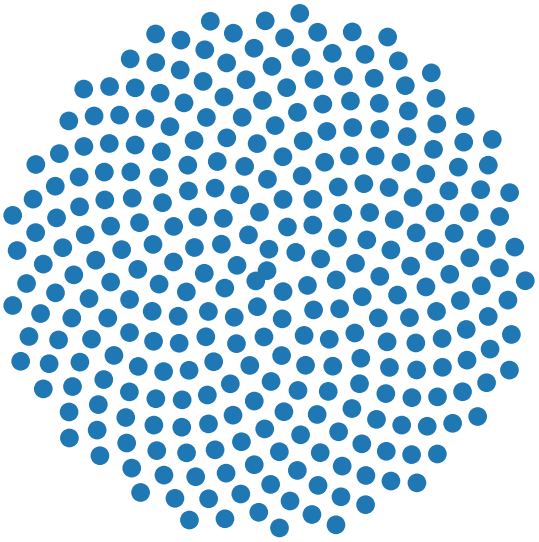


Figure 1: The sunflower pattern as derived by Vogel.[6] It is homogeneous and isotropic and a regular pattern. Here, 300 dots are shown.

4.2 Intersection

We now know how the rays will travel from the objective through the scene in a straight line. But without interacting with any objects, this would make for a very pointless simulation, as we would not even get any return signal. There are several objects in the physical set-up: an objective lens, immersion oil, a glass microscope slide, water, a neuron and a detector. For this simulation we do not worry too much about the volume inside these objects, but mostly about their surfaces. And specifically the surfaces where two volumes with different refractive indices meet. These are the places where the rays can reflect and refract. In the simulation there are two surface shapes: a cylinder for the neuron, and planes for everything else. That is the objective, detector and glass-water interface. The transition between immersion oil and glass has no interface, as it has the same refractive index as the glass.

Our goal is to find where each ray intersects with the next object along its path. So we need to know which object is closest along the direction of travel. And we need to know where the intersection takes place exactly, so we know where the reflection and transmission rays originate from. Finally we need the optical path length that was travelled. Since a ray can only travel forward along its direction vector, all three values can be derived from a single yet unknown number: the distance to each object along this direction vector. To find which object is closest, we find the minimum of all distances to each object. To find where the intersection is, we multiply the normalised direction vector by the distance and add this to the current position vector. And to find the optical path length, we multiply to distance by the refractive index of the medium.

The simplest of the two shapes is the plane, so let's start with that. Also see fig. 2. We assume the direction vector $\hat{\mathbf{v}}$ is normalised. It can be decomposed into three perpendicular basis vectors, whose sum results in the original: One basis vector $\mathbf{v}_\perp$ normal to the plane. This is the shortest path to the plane. And two basis vectors $\mathbf{v}_{\parallel 1,2}$ parallel to the surface of the plane. Following these vectors you will never get any closer or further away.

If you now follow the original direction vector by one unit, your shortest path distance to

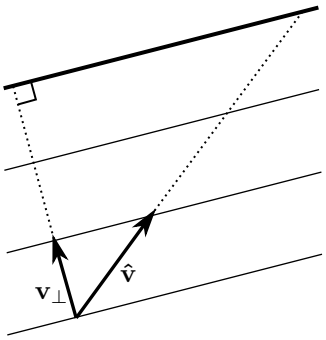


Figure 2: Finding the distance from the base of vector $\hat{\mathbf{v}}$ to the solid plane. If we first find $\mathbf{v}_\perp$, as finding the number of perpendicular vector lengths to the plane is easy, especially in our simulation where all planes are horizontal. In this diagram, the distance from the vector base to the plane is three vector lengths.

the plane will decrease by the length of $\mathbf{v}_\perp$. So the number of vector length you need to travel is the original shortest path distance divided by the length of $\mathbf{v}_\perp$. In our simulation all planes are along the x and y axes. This makes the calculation especially trivial. The normal vector is simply the z-component of the direction vector, and the shortest path distance is the distance along z. By dividing the latter by the former you know the distance that must be travelled along the direction vector.

The distance to the cylinder is a bit more involved. A cylinder is symmetric over its entire length. We can take a planar cross section of the cylinder and be left with a circle. This time we split the direction vector into two components: one in the cross section plane, and one perpendicular to it, along the cylinder. We only concern ourselves with the circle and vector in this cross section, see fig. 3. In the figure we want to find the distance from the base of vector $\mathbf{v}_\parallel$ to the dot. The first step is to find the distance to the half way point in the circle. This distance is given by $m = \mathbf{x} \cdot \hat{\mathbf{v}}_\parallel$. We then add or subtract h to go to either side of the circle. To find h we identify the two right angle triangles that share an edge. The Pythagorean theorem says that the length of this edge equals both

$$\|\mathbf{x}\|^2 + m^2 = r^2 + h^2$$

from which we can extract

$$h = \sqrt{\|\mathbf{x}\|^2 + m^2 - r^2}.$$

We find the distance to the circle in the 2D projection by subtracting h or, if the rays is inside the circle going out (when $\|\mathbf{x}\| < r$), by adding h

$$d' = m \mp h,$$

which, similar to the plane, we finally divide by the length of $\mathbf{v}_\parallel$ to find the travel distance in 3D.

$$d = \frac{d'}{\|\mathbf{v}_\parallel\|}$$

Now that we know the distance to the plane or cylinder, we can find the location of the intersection by first scaling the direction vector to this length, and adding this displacement to

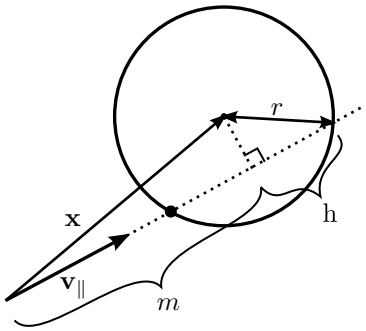


Figure 3: Finding the distance from the base of vector $\mathbf{v}_\parallel$ to the dot. The circle is the cross section of the cylinder and $\mathbf{v}_\parallel$ is the projection of the direction vector onto this cross section plane. $\mathbf{x}$ is the vector from the ray origin to the circle center, r is the circle (and neuron) radius, m is the distance to half way along the circle and h is the rest of the distance to the far side, but also the distance between the dot and the half way point.

the original location. The optical path length traversed is the product of the distance and the refractive index. This is added to the optical path length that the ray had travelled before.

Finally, we need to know the normal vector of the surface at the location of the intersection. This vector is of unit length and on the same side of the surface as the incoming beam. This means that the dot product of the ray direction and the normal vector must always be negative. The normal vector is used in the next step, for determining the direction and amplitude reduction factor of the reflecting and transmitted rays. For a plane this is trivial: the normal vector is independent of the intersection location and only needs to be flipped based on from which side the ray originates from.

For the cylinder, the normal vector always lies in the cross section plane that we used earlier, since this plane is perpendicular to the axial direction. We find it by subtracting the intersection location from the circle centre. Finally the vector is normalised and, if necessary, flipped to face towards the ray origin.

4.3 Transmission and Reflection

The next step after finding where a ray intersects with an object is to continue the path of the ray. But the rays do not continue on a straight path. Some part of the light rays is transmitted into and refracted by object it just intersected. The rest of the light is reflected. These two new directions of light are represented by two new rays which are clones of the original but with a different direction and amplitude. We must first find their directions before the amplitudes can be derived.

4.3.1 Law of Reflection and Snell's Law

Again we'll start with the simpeler case: the reflection. All rays, i.e. incoming, reflection and transmission, are coplanar. The normal vector also lies in this plane. The law of reflection states that the angle of incidence θ_i equals the angle of reflection θ_t.[3] The angle is measured between the direction and normal vectors. All vectors are unit vectors so the lengths of the projections of the direction vectors onto the normal vector and the surface remain constant. Only the projection onto the normal vector is inverted. The length of this component vector is

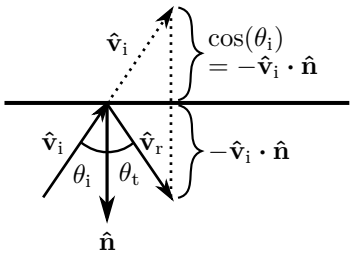


Figure 4: The incoming direction $\hat{\mathbf{v}}_\mathrm{i}$ must be turned into the reflected direction $\hat{\mathbf{v}}_\mathrm{r}$. This is done by subtracting $(\hat{\mathbf{v}}_\mathrm{i} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$ twice. The surface component of the vector remain constant.

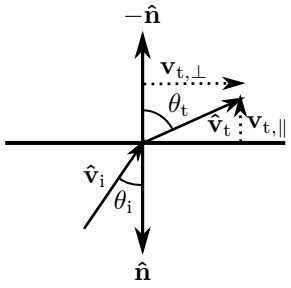


Figure 5: The incoming direction $\hat{\mathbf{v}}_\mathrm{i}$ must be turned into the transmitted and refracted direction $\hat{\mathbf{v}}_\mathrm{t}$. We calculate $\mathbf{v}_{\mathrm{t},\perp}$ and $\mathbf{v}_{\mathrm{t},\parallel}$ separately and add the components.

given by the dot product between the incident ray and the normal vector. We can subtract this component vector twice from the incoming direction to end up with the reflected direction

$$\hat{\mathbf{v}}_\mathrm{r} = \hat{\mathbf{v}}_\mathrm{i} - 2(\hat{\mathbf{v}}_\mathrm{i} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}.$$

This process is visualised in fig. 4.

The refraction direction is more involved since it requires Snell's law[3]

$$n_\mathrm{i} \sin(\theta_\mathrm{i}) = n_\mathrm{t} \sin(\theta_\mathrm{t}).$$

We can again split the transmission vector into two components: one along the normal vector $\hat{\mathbf{v}}_{\mathrm{t},\parallel}$ in its negative direction and one along the surface $\hat{\mathbf{v}}_{\mathrm{t},\perp}$ coplanar with the normal vector and incoming direction. See fig. 5. We don't yet know the direction of the surface component in 3D but we do know the normal vector. The direction vector is normalised so the length of the component along the normal is $\cos(\theta_\mathrm{t})$ and the component along the surface has length $\sin(\theta_\mathrm{t})$. With Snell's law and the Pythagorean trigonometric identity we can derive

$$\sin(\theta_\mathrm{t}) = \frac{n_\mathrm{i}}{n_\mathrm{t}} \sin(\theta_\mathrm{i}) \quad \rightarrow \quad \cos(\theta_\mathrm{t}) = \sqrt{1 - \left(\frac{n_\mathrm{i}}{n_\mathrm{t}}\right)^2 \sin^2(\theta_\mathrm{i})}$$

which only yields positive results, but that's no issue since $-\pi/2 < \theta_\mathrm{t} < \pi/2$. So we need to find $\sin^2(\theta_\mathrm{i})$. From the definition of the dot product we can derive

$$\hat{\mathbf{v}}_\mathrm{i} \cdot \hat{\mathbf{n}} = \|\hat{\mathbf{v}}_\mathrm{i}\| \, \|\hat{\mathbf{n}}\| \cos(\theta_\mathrm{i}) = \cos(\theta_\mathrm{i})$$

which leads to

$$\sin^2(\theta_\mathrm{i}) = 1 - \cos^2(\theta_\mathrm{i}) = 1 - (\hat{\mathbf{v}}_\mathrm{i} \cdot \hat{\mathbf{n}})^2.$$

Combining all this, the transmission component along the normal vector is given by the formula[5]

$$\mathbf{v}_{\mathrm{t},\parallel} = -\sqrt{1 - \left(\frac{n_\mathrm{i}}{n_\mathrm{t}}\right)^2 (1 - (\hat{\mathbf{v}}_\mathrm{i} \cdot \hat{\mathbf{n}})^2)}\hat{\mathbf{n}}.$$

The component along the surface should have a length of $\sin(\theta_\mathrm{t}) = (n_\mathrm{i}/n_\mathrm{t})\sin(\theta_\mathrm{i})$ But we don't yet have an expression for this direction yet in three dimensions. It should be perpendicular to the normal vector and coplanar with it and the incoming direction vector. We can produce this vector as follows. First we take the dot product of the incoming direction and the normal vector $\hat{\mathbf{v}}_\mathrm{i} \times \hat{\mathbf{n}}$ which produces vector that is perpendicular to $\hat{\mathbf{n}}$, so along the surface, but also perpendicular to $\hat{\mathbf{v}}_\mathrm{i}$, according to the right hand rule. Its length is $\sin(\theta_\mathrm{i})$. So if we now take another cross product with the normal vector from the other side $\hat{\mathbf{n}} \times (\hat{\mathbf{v}}_\mathrm{i} \times \hat{\mathbf{n}})$ we get the direction we want: along the surface and coplanar with the normal and incoming direction. The length is still $\sin(\theta_\mathrm{i})$. To get the required length of $\sin(\theta_\mathrm{t})$ we only need to multiply this by $n_\mathrm{i}/n_\mathrm{t}$ which gives[5]

$$\mathbf{v}_{\mathrm{t},\perp} = \frac{n_\mathrm{i}}{n_\mathrm{t}} \hat{\mathbf{n}} \times (\hat{\mathbf{v}}_\mathrm{i} \times \hat{\mathbf{n}}).$$

Finally just sum the two components to obtain the final direction of the transmitted light.

$$\hat{\mathbf{v}}_\mathrm{t} = \mathbf{v}_{\mathrm{t},\parallel} + \mathbf{v}_{\mathrm{t},\perp}$$

4.3.2 Fresnel Equations

When a light ray splits up like this the total power must be conserved. That means that the power going in must be equal to the total outgoing power. The power of a ray is related to its amplitude. So the amplitude of the two outgoing rays must be different from that of the incoming ray. How the amplitude is affected is given by the Fresnel equations. Real light has some polarisation, which can be broken down into two components. Each component has a separate amplitude and phase, which depend on the directions into which the polarisation is decomposed. These polarisation directions are always perpendicular to the direction of travel. In the Fresnel equations one component is coplanar to the normal vector and denoted by a $\parallel$ sign. The other component is perpendicular to this plane, denoted by $\perp$. The Fresnel equations are given by[3]

$$r_\parallel = \frac{n_\mathrm{t}\cos\theta_\mathrm{i} - n_\mathrm{i}\cos\theta_\mathrm{t}}{n_\mathrm{t}\cos\theta_\mathrm{i} + n_\mathrm{i}\cos\theta_\mathrm{t}} \qquad t_\parallel = \frac{2n_\mathrm{i}\cos\theta_\mathrm{i}}{n_\mathrm{t}\cos\theta_\mathrm{i} + n_\mathrm{i}\cos\theta_\mathrm{t}}$$
$$r_\perp = \frac{n_\mathrm{i}\cos\theta_\mathrm{i} - n_\mathrm{t}\cos\theta_\mathrm{t}}{n_\mathrm{i}\cos\theta_\mathrm{i} + n_\mathrm{t}\cos\theta_\mathrm{t}} \qquad t_\perp = \frac{2n_\mathrm{i}\cos\theta_\mathrm{i}}{n_\mathrm{i}\cos\theta_\mathrm{i} + n_\mathrm{t}\cos\theta_\mathrm{t}}$$

where the amplitude components of the incoming ray are multiplied by the respective $r_{\parallel,\perp}$ to yield the reflected ray's amplitudes, and multiplied by $t_{\parallel,\perp}$ for the transmitted ray's amplitudes. The reflected components' power each gets scaled by a factor of $R_{\parallel,\perp} = r^2_{\parallel,\perp}$ respectively while the transmitted power is scaled by $T_{\parallel,\perp} = \frac{n_\mathrm{t}\cos\theta_\mathrm{t}}{n_\mathrm{i}\cos\theta_\mathrm{i}} t^2_{\parallel,\perp}$.[3] The extra factor is included in the transmitted power since the refraction affects the cross section of the beam area, which changes the amplitude beyond the fraction that is lost to the reflection. Notice that for an external reflection, meaning $n_\mathrm{t} > n_\mathrm{i}$, $r_\perp$ will always be negatively valued and $r_\parallel$ will be negative when $\theta_\mathrm{i} > \arctan(n_\mathrm{t}/n_\mathrm{i})$. Similarly for an internal reflection, when $n_\mathrm{i} > n_\mathrm{t}$, $r_\perp$ is positive and $r_\parallel$ only under the same condition of $\theta_\mathrm{i} > \arctan(n_\mathrm{t}/n_\mathrm{i})$.

But in our simulation, keeping track of the polarisation, with the different directions and relative phases can take a lot of complicated computation. To simplify this we choose to not simulate the polarisation of light. We assume that every ray has equal power in each component of the polarisation. The reflected and transmitted power now each take half of their power from each component, i.e.

$$R = \frac{1}{2}(R_\parallel + R_\perp) \quad \text{and} \quad T = \frac{1}{2}(T_\parallel + T_\perp).$$

For polarised light one can show that $R_\parallel + T_\parallel = 1$ and $R_\perp + T_\perp = 1$, from which it follows that $R + T = 1$ thus energy is conserved.

In the simulation we keep track of a ray's amplitude, not its power. So we want to know r and t instead of R and T. These are given by the following formulas, where we assume r is

negative for all external reflections ($n_\mathrm{t} > n_\mathrm{i}$):

$$
\begin{aligned}
r &= \pm\sqrt{R} = \pm\sqrt{\frac{1}{2}(R_\parallel + R_\perp)} \\
&= \pm\sqrt{\frac{1}{2}(r_\parallel^2 + r_\perp^2)} = \pm\sqrt{\frac{1}{2}\left(\left(\frac{n_\mathrm{t}\cos\theta_\mathrm{i} - n_\mathrm{i}\cos\theta_\mathrm{t}}{n_\mathrm{t}\cos\theta_\mathrm{i} + n_\mathrm{i}\cos\theta_\mathrm{t}}\right)^2 + \left(\frac{n_\mathrm{i}\cos\theta_\mathrm{i} - n_\mathrm{t}\cos\theta_\mathrm{t}}{n_\mathrm{i}\cos\theta_\mathrm{i} + n_\mathrm{t}\cos\theta_\mathrm{t}}\right)^2\right)}, \\
t &= \sqrt{\frac{n_\mathrm{i}\cos\theta_\mathrm{i}}{n_\mathrm{t}\cos\theta_\mathrm{t}}T} = \sqrt{\frac{n_\mathrm{i}\cos\theta_\mathrm{i}}{n_\mathrm{t}\cos\theta_\mathrm{t}}\frac{1}{2}(T_\parallel + T_\perp)} = \sqrt{\frac{n_\mathrm{i}\cos\theta_\mathrm{i}}{n_\mathrm{t}\cos\theta_\mathrm{t}}\frac{1}{2}\left(\frac{n_\mathrm{t}\cos\theta_\mathrm{t}}{n_\mathrm{i}\cos\theta_\mathrm{i}}t_\parallel^2 + \frac{n_\mathrm{t}\cos\theta_\mathrm{t}}{n_\mathrm{i}\cos\theta_\mathrm{i}}t_\perp^2\right)} \\
&= \sqrt{\frac{1}{2}(t_\parallel^2 + t_\perp^2)} = \sqrt{\frac{1}{2}\left(\left(\frac{2n_\mathrm{i}\cos\theta_\mathrm{i}}{n_\mathrm{t}\cos\theta_\mathrm{i} + n_\mathrm{i}\cos\theta_\mathrm{t}}\right)^2 + \left(\frac{2n_\mathrm{i}\cos\theta_\mathrm{i}}{n_\mathrm{i}\cos\theta_\mathrm{i} + n_\mathrm{t}\cos\theta_\mathrm{t}}\right)^2\right)}.
\end{aligned}
$$

The reflected ray's amplitude, without polarisation, is multiplied by r and the transmitted ray's amplitude is multiplied by t.

4.4 Detection

The objective lens is an integral part of the set-up as it allows to collect light in a large aperture and focus it onto the small optical fibre. It bends the light and changes the phase, i.e. it changes the optical path length due to being made of glass. A physical lens has undesirable side effect such as spherical aberration, dispersion and coma unique to each lens' design. Instead, we simulate an idealised thin lens using the rules of geometrical optics. An ideal convex thin lens is characterised by four properties.[3]

1. All rays passing through the centre of a lens do not change direction.
2. All rays parallel to the optical axis, once past the lens, converge on the optical axis at the focal length.
3. All rays radiating from a point that pass through the lens converge in a point on the other side of the lens.
4. The relative phase of rays emitted from one point is preserved in the other focal point, regardless of where they pass through the lens.

These simple rules generally hold for physical convex thin lenses when using small angles, and are also goals that multi-lens objectives strive towards. Simulating the effects of an ideal lens on a ray, however, is non-trivial, but can be simplified with some considerations. We use an approach that exploits rules 3 and 4 and our specific set-up to avoid simulating the actual effects of a lens.

Rule 3 tells us that all rays that hit a specific point on the fibre optic must, before the lens, disregarding the rest of the set-up, radiate out of a single point. This virtual point is called a projection. Reasoning the other way around: All rays that, disregarding all other set-up, radiate from this projected point and pass through the lens, must hit the fibre optic at a single known point. So instead of calculating the transmission angles through the lens, we can trace a ray backwards and compute if and where it would have originated from the projection had the rest of the set-up not been there.

But we still need to know the phase difference of rays that hit the real detector. Rule 4 now tells us that the relative phase, or OPL difference, is the same for all rays that would travel

between the two points. But there may still be a constant phase difference between these two points. And this phase difference may well differ between other pairs of detector and projection points. In most cases this phase difference will be small, but correcting for it is computationally easy.

Imagine that using rule 2 we send a ray from the projected sensor along the optical axis. It will converge with other parallel rays onto the focal point of the lens. The rays are parallel on the projection side of the lens and if they are extended further in that direction, they will never converge or diverge. But this is equivalent to rays that originate from a point source that is infinitely far away. Taking it all together, multiple in-phase rays emitted from the projected sensor, which is perpendicular to the optical axis, will converge to the focal point. The rays are in phase at the projection and stay in phase as they traverse in a straight line, so they must also have been in phase at the infinitely far away origin point. And invoking rule 4 again, the rays must also be in phase on the focal point. The only change in phase comes from the optical path length difference between the focal point and the detector. The optical path length from the projection to the focal point is effectively zero. So we can take a ray that hits the lens, try to trace it back to the projection, find if and where it hits the real detector, and add the optical path length between there and the focal point to the final optical path length of the ray.

There is still a problem here though. When a ray is very near the edge of the detector, it gets detected. But for a slightly modified setup, for example one where the cylinder diameter is a nanometre larger, this ray falls just outside the detector. If this ray contains any significant amplitude, this will suddenly change the signal which can add many unwanted peaks and noise to the detector sensitivity. To prevent this, we need a smooth falloff on the edge of the detector, which prevents jumps in detected amplitude. In order to preserve te edge of the detector, we chose to apply a parabolic falloff to the detector. Rays that get detected on a radius r from the detector centre which has a maximum radius of R, the detected amplitude is scaled down by a factor $1 - (r/R)^2$.

Now we finally know which rays hit the detector. We know their optical path lengths and their amplitudes. This information must be converted into a light spectrum, since this is what the physical detector detects. For this purpose we must effectively combine the electric field contribution of all rays. The first step is to find the phase, which is obtained by multiplying the optical path length by the wave number of the light in the spectrum. The electric field vector of the ray has a length equal to the amplitude of the ray and the phase dictates the direction. It is convenient to represent this vector as a complex number. Summing the electric field of all rays yields the total electric field received at the selected wavelength. Squaring the electric field gives the intensity as a function of the wavelength[3]

$$I(k) = \left| \sum_j E_j \mathrm{e}^{\mathrm{i}k\Lambda_j} \right|^2$$

where E_j is the amplitude of the j^{th} ray and Λ_j is its optical path length. This intensity is the signal that the detector provides.

5 Signal Processing

As explained in previous sections, our goal is to find the change in phase in the signal as the diameter of the cylinder changes. We perform the simulation for many different cylinders and in the signal processing step we extract this phase from each signal. At the heart of this algorithm is the Fourier transform. The Fourier transform decomposes a continuous signal of infinite length

into its constituent frequencies with corresponding phase offsets. It produces a new complex function over the FT dual of the quantity that was transformed over. The dual of time t is the angular frequency ω and the dual of the (angular) wave number k is real space x. The signal is a function of intensity over wave number so a Fourier transform produces a function over real space. For each position in real space it gives us the intensity and phase of the signal returned.

This works slightly differently for non-continuous signals of finite length. For such a signal like the one produced by a spectrometer with a finite number of discrete pixels, or our simulation, we need to use a discrete Fourier transform (DFT). The DFT makes some assumptions about the signal so it can take in and put out a discrete finite signal. The most important assumption is that input signal is a unit of an infinite signal that is build up by chaining these units. From this follows the second assumption, namely that the original signal is a sum of sinusoids with each a wavelength (in k-space) that fits an integer number of times in a signal unit, with two pixel as the smallest wavelength. For each of these component sinusoids the DFT calculates the amplitude and phase at which it occurs in the original signal. Using an inverse Fourier transform the original signal could be recovered losslessly from this data.

Since the DFT assumes the signal is periodic, the signal should be continuous when chained end to end. A large discontinuity in the edges adds much unwanted modulation to the final result. We can mitigate this by making the edges meet in a continuous fashion. This is achieved by multiplying the signal by a windowing function. It brings the edges of the signal close to zero, while leaving the middle of the signal intact. The modulation is strongly reduced while adding minimal noise. We chose to use the Hamming window specifically because it produces a very clean DFT and phase slope.

The signal we receive from the detector is the intensity of the light, which can of course never be less then zero. So the midpoint of the oscillation in the spectrum is always above zero. Applying the windowing function directly to the signal would distort the midpoint, creating unwanted modulations. So first we must try to bring the midpoint near zero. One may try to do so by subtracting the signal mean which would work for signals which contain many oscillations. But for signals with very few oscillations, like we have with small neurons, there is likely an unequal number of oscillations on either side of the midpoint, which causes the mean to not be on the midpoint. Subtracting the mean then does not bring the midpoint to zero and the problem remains. A more effective approach for these signals is to find the midpoint by taking the average of the minimum and maximum intensity of the signal, and subtracting that.

If we were to take a Fourier transform of this signal, which was windowed after the midpoint was subtracted, we would likely find a spike at $x = 0$. This represents the total sum of signal that the DFT was taken of and it does not hold any phase information. We can completely remove this spike from the DFT result by subtracting the mean of the signal from the signal, after the windowing function. This finally yields a DFT that can be used for further processing.

The majority of the returned signal comes from the sample, which has produced the oscillating interference spectrum. Because it is the majority of the signal, it forms a peak in the Fourier transform where the location of the peak represents the size of the neuron. We are after this size but the resolution of the DFT is on the micrometre scale, while we want to measure sub-nanometre diameter changes. It is for that reason that we need to look at the phase of the returned signal. The phase of each component in the signal is represented in the DFT as the phase of the complex number that represents said component, and the amplitude is represented as the magnitude of the complex number. Since our interest is in the component with the highest amplitude, we want the phase of the component that forms a peak in the DFT. So the phase that corresponds to a particular simulation is that of the component with the highest magnitude in the DFT.

Next we want to see how the phase changes for different diameters of cylinder. For a range

of cylinders the ray tracing simulation is repeated, the spectrum is created and preprocessed for taking the Fourier transform. For each cylinder we find the peak in the DFT result and take the phase of that component. Finally we have a range of phases we can analyse. The primary method of analysis will be by taking a derivative of this phase over the cylinder diameter. This is called the phase slope and also the sensitivity of the detector. Its unit of mrad/nm means that for every nanometre the cylinder's size increases, the phase we find increases by a certain number of milliradians.

6 Results & Conclusion

In our setup the light rays are focused by a microscope objective. The location of this focus is important to get as much relevant signal returned as possible. Relevant signal is that portion of the light that bounced either off the bottom or off the top of the neuron. Both reflections must be present to be able to measure the distance between them. The focus position z is defined as the height of the microscope objective, with the zero point when the light rays focus to a point on the glass–water interface. The rays are refracted and defocused by the various objects in the scene which makes analytically deriving an optimal focus position difficult. We choose to measure the focus at which we get a good signal. In fig. 6 we vary the focus and measure how much signal is returned. There are three peaks in the intensity of the returned signal. In the first peak the focus is on the glass–water interface at $z = 0\,\mu\text{m}$. The majority of the signal intensity is from the reflection off this interface which does not hold any information about the diameter of the neuron. The second peak between $1.0\,\mu\text{m}$ and $1.5\,\mu\text{m}$ is where the rays focus close to the centre of the cylinder. This enables enables many rays to approach both the near and far side of the cylinder at normal incidence. When those rays are reflected they are headed in the reverse direction which takes them back to the detector. This signal has reflections from both sides of the cylinder so it holds information about the diameter of the cylinder. The third peak in intensity is very wide, from around $2.0\,\mu\text{m}$ to $3.3\,\mu\text{m}$. At this focus the rays reflect off the far side of the cylinder. But they do so after being refracted by it. This refraction heavily defocuses the rays which is why the peak is significantly wider than the previous two. The signal mostly reflected off this far side and not off the near side so it holds no information about the diameter.

We've learned that to extract the diameter information from the light beam it needs to be focused on the middle peak. Furthermore we assume that the maximum intensity in this peak is the optimal focus position. Next need to know how this focus position depends on the diameter of the cylinder. This is again determined experimentally: For a range of cylinder diameters, we measure the received intensity for focus positions between 0.4 and 0.8 times the diameter. For each of the different diameters we fit a Gaussian to the intensity over focus position plot, where the peak of the Gaussian will end up at the optimal focus position for that cylinder. The position of this peak was then plotted against the respective cylinder diameter in fig. 7. We then fit a line to this plot and found that the optimal focus position can be found near

$$z = 0.581d + 0.069\,\mu\text{m}$$

where z is the focus position and d is the diameter of the cylinder, in case the cylinder is $5\,\text{nm}$ off the glass. For all subsequent results the rays are focused to this position, unless noted otherwise.

Now that we have the correct focus to produce a signal that contains the required information, we can proceed with the final simulations and signal processing to find the phase slope. For 151 cylinders with a diameter between $1.0\,\mu\text{m}$ and $2.0\,\mu\text{m}$ we perform ray tracing to get a spectrum from each. The spectrum runs from wavelengths $\lambda = 650\,\text{nm}$ to $\lambda = 900\,\text{nm}$. We center the

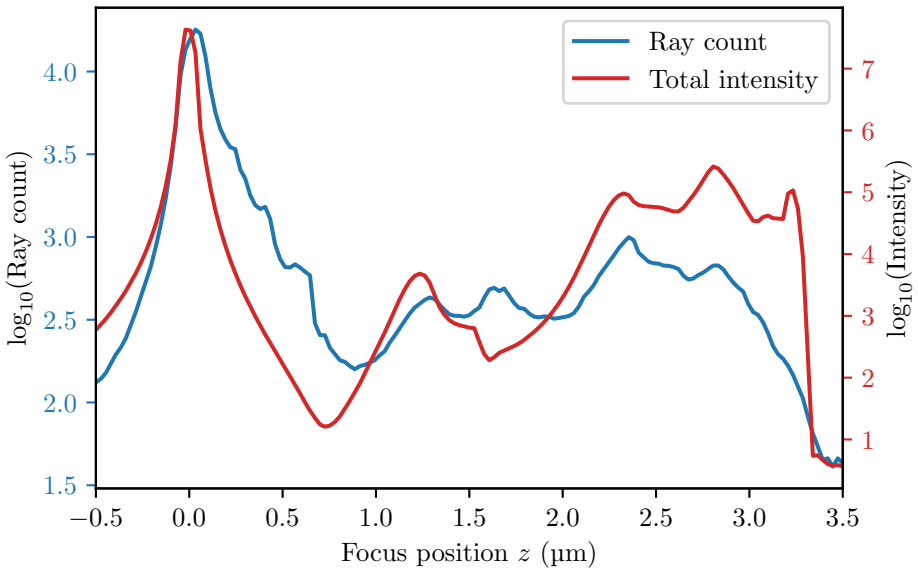


Figure 6: The number of rays captured on the sensor, and the total light intensity received by the detector, over the focus position. The cylinder has a diameter of 2 µm and sits 5 nm off the bottom of the glass. At $z = 0$ the rays focus to a point on the glass–water interface and all reflect back to the lens and detector. From 1.0 µm to 1.5 µm the rays nearly focus in the centre of the cylinder. Between 2.0 µm and 3.3 µm the rays have been diffracted and defocused by the curvature of the cylinder and focus on its top side, where they reflect back to the detector. Discrepancies where the ray count is higher that the total intensity are mainly from higher order reflections, which retain minimal amplitude. When the ray count is low while the intensity stays high, most of the reflected rays are from single reflections.

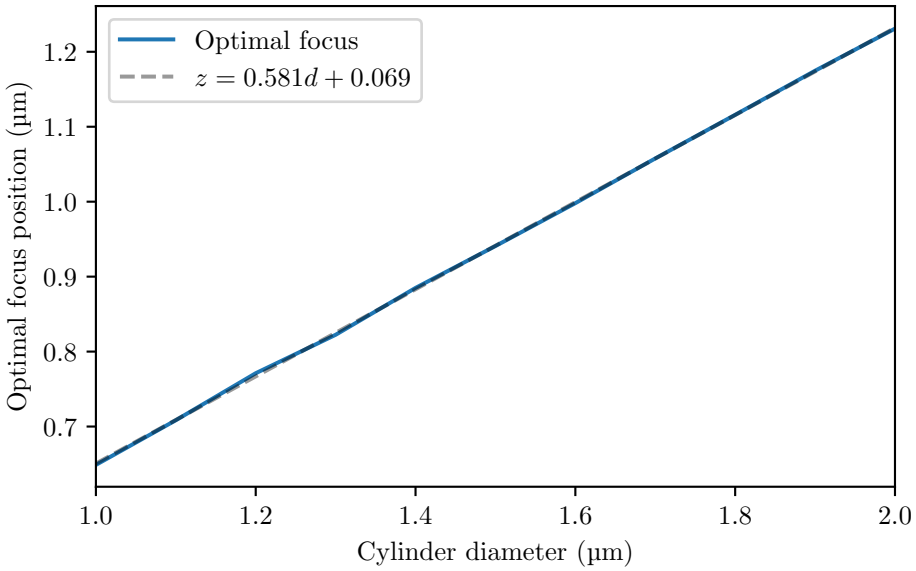


Figure 7: The ideal focus position for a cylinder with a particular diameter is the location of the middle intensity peak in fig. 6. The relation between focus and diameter is approximately linear so we made a linear fit to be able to interpolate and extend the focus to other cylinders.

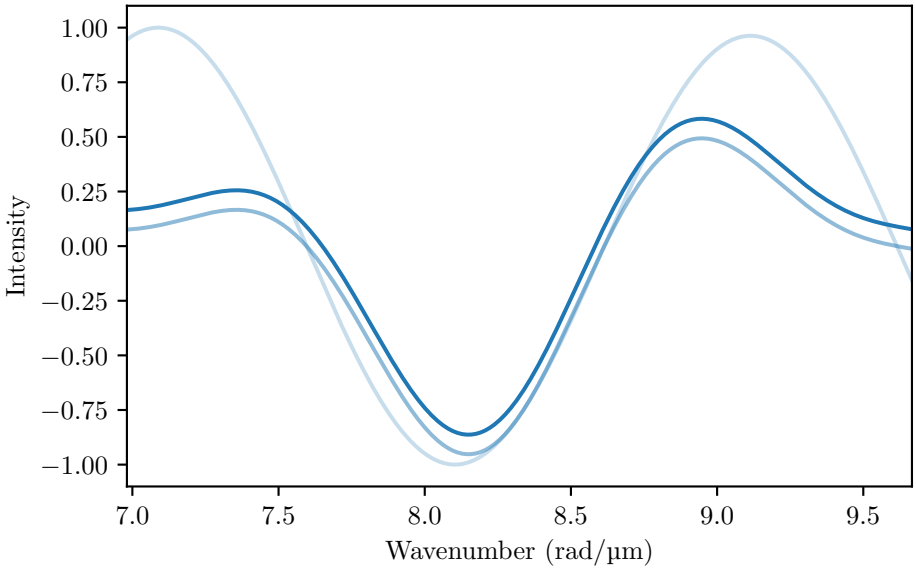


Figure 8: Light blue: Spectrum of neuron that is 1.4 µm in diameter, after repositioning it to fit between −1 and 1. The left and right edges don't match up. Medium blue: The same spectrum, but with the Hamming window applied. The left and right edges are now nearly matching. Dark blue: The previous spectrum, but with mean subtracted.

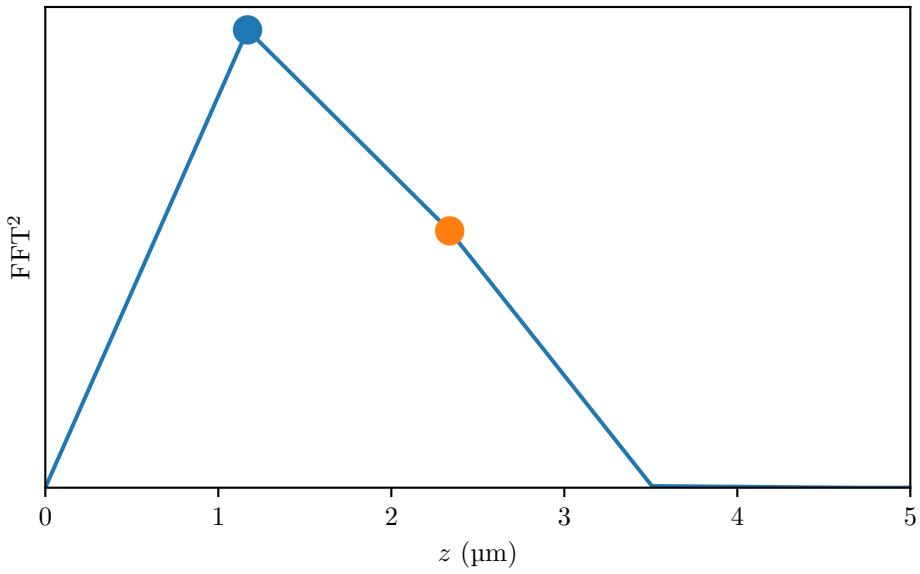


Figure 9: FFT of the preprocessed spectrum of a neuron with diameter 1.4 µm, which can be seen in fig. 8. The blue dot is placed on the first relevant pixel of the FFT, and the orange dot on the second. In the next plots these colors will refer to these points.

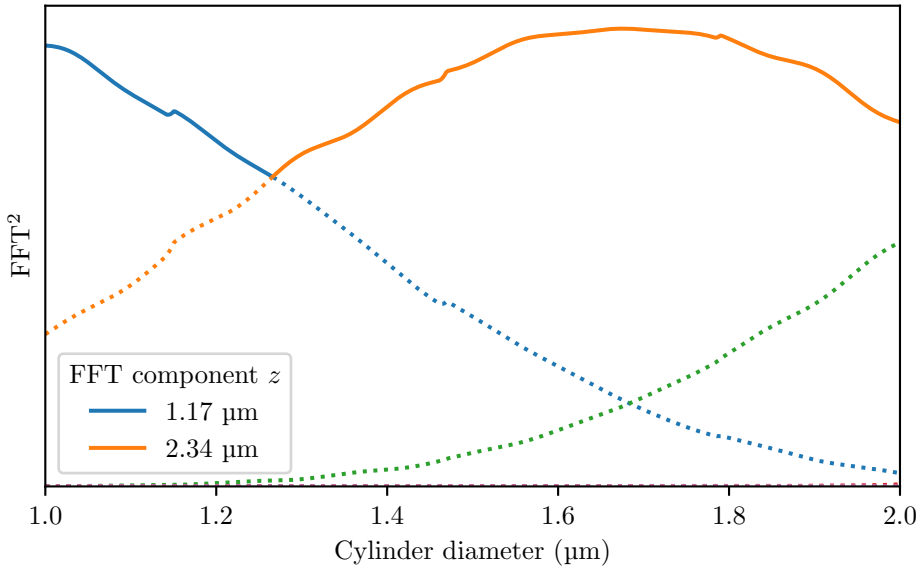


Figure 10: FFT for a range of cylinder diameters. Each line is one component, see fig. 9. The line is solid when it has the highest magnitude, meaning we will use this component's phase. Dotted lines are not the highest magnitude for a particular diameter of cylinder.

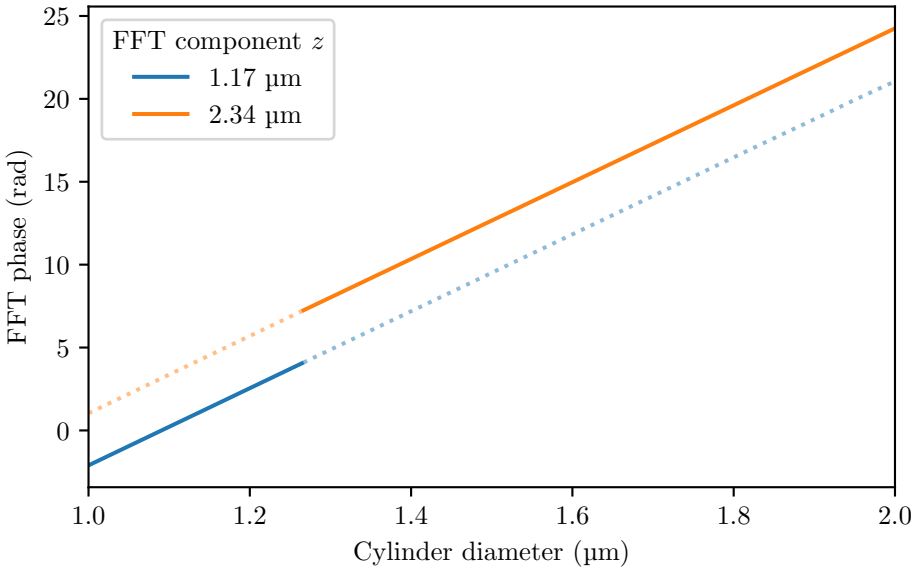


Figure 11: The phase of the first two relevant components in the FFT of each cylinder's spectrum. The lines are solid when this component has the highest magnitude of the FFT and dotted otherwise.

spectra to zero, apply the Hamming window and subtract their means. The intermediate spectra after each effect can be seen in fig. 8. Next we apply a fast Fourier transform (FFT), which is a fast implementation of the DFT, to each spectrum to get the amplitude and phase for each component of the spectrum, see figs. 9 and 10. For each FFT result we take the phase of the component with the highest amplitude, see fig. 11. If we then take the slope of that phase over the diameter, end up with the sensitivity of the detector for the different cylinder diameters. This is plotted in fig. 12 where the solid line as a whole is our final result. The colours just indicate what part of the phase slope is based on what component of the FFT.

The oscillation in the spectrum we receive from the detector should be close to $\cos(2ndk)$. If we omit the ray tracing and instead perform the signal processing with this ideal spectrum, we find the green dashed line. The black dashed line uses the prediction of eq. (2) for the phase slope. In equation eq. (2) we make an estimation for the phase slope. Using the refractive index of the neuron of $n = 1.39$ and wavelengths from 650 nm to 900 nm we estimate that the phase slope we will be 23.14 nm/mrad.

We find that the sensitivity that results from the simulation very closely matches what we would predict from this setup. There is however a small but significant oscillation in the sensitivity with an amplitude of around 0.5 mrad/nm. This would induce a variability of at most 2.2 % of the diameter change, or 0.022 nm for a 1 nm diameter change. The standard deviation of the sensitivity around the predicted mean is 0.23 mrad/nm. The standard deviation from ideal spectrum slope is 0.19 mrad/nm but half of this is caused by the spike around 1.15 µm. For the previous case its contribution is much smaller. The cause of these peaks is unknown.

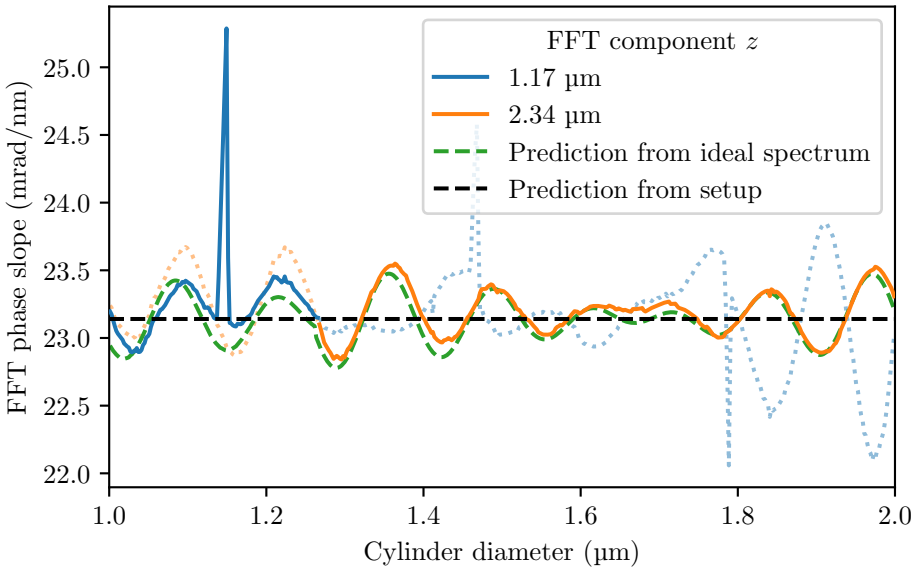


Figure 12: The phase slope or sensitivity for different sized cylinders. Each colour corresponds to a component of the FFT. The solid lines signify the component with the greatest amplitude, and therefore the slope that is used as the final sensitivity of the setup at this diameter. The green line is the phase slope we get if we use $\cos(2ndk)$ for the spectrum instead of the ray tracing result. The black dashed line is our prediction of the phase slope, based on the refractive index of the neuron and the wave number range of the spectra.

7 Discussion & Future Research

The intention of the simulation is to be very simple but still useful. Many aspects of physical light transport were ignored which may have a significant impact on the results produced. The simulation can be extended by adding these aspects which would make it more realistic, but also more complicated and potentially slower to perform.

One of the effects that was omitted is the diffraction limit. Light cannot focus to a point smaller than this limit. This is a wave based effect so it can be hard to simulate using rays. One way to emulate its effects is to intentionally defocus the ray bundle. It will cause less rays to be collected by the sensor and can significantly broaden the reflection peaks we see in fig. 6, which may also have the effect of making the phase slope less dependent on the precise focus position.

A similar effect with a different cause is aberrations from the objective. For the simulation of the lens we use an idealised lens following geometrical optics. No lens is perfect however and all introduce imperfections called aberrations into the beam. These may include general imperfect focus, primarily near the edges of the objective, or a focus depending on the wavelength of the light, caused by dispersion.

Dispersion is not included in our simulation. Dispersion is the dependence of the refractive index on the wavelength of the light. This would change the direction of the rays, which means we need to do a separate ray tracing simulation for each wavelength. That would cause simulation time to go up a thousandfold, for having a thousand different wave numbers in our spectrum, but it would be more accurate.

Another effect that was not included is the absorption of light. When a ray bounces off an object, part of the light is reflected and part is transmitted, but a part of the energy is also absorbed and turned into heat. The absorption could depend on the color of the light which would further complicate matters. With absorption, the same ray count would be detected, but the amplitude would be lower.

Finally a major omission is the proper polarisation of light. Polarised light has many interesting effects and is frequently used to aid in observing certain features of an object. Adding polarised light would not only make the simulation more realistic, but also add a new dimension of information that can be used or analysed.

Some techniques may also be able to be improved. The falloff of the detector was added in a late stage of the project and was not properly analysed. The inverse parabolic falloff was chosen over e.g. a bell curve because the parabolic falloff retains the finite size of the detector, which provided the most continuity with previous simulations.

The area that may require the largest overhaul is the data processing. The phase of the spectrum is obtained through a Fourier transform, but this transform has shown to add undesirable oscillations to the phase slope. One approach to get the phase from the spectrum may be to perform a more common fit of a sinusoidal wave to the spectrum, and extract the phase directly from this fit. Or maybe there are other ways still to obtain the nanometre scale diameter information that don't rely on the phase.

One major unexplained phenomenon is the sharp peaks in fig. 12. These may be analysed by examining the spectra that produces them. But a concrete explanation of the cause is not available yet.

After the simulation software is more finalised, it would be beneficial to optimize the program to run in less time while yielding the same or similar results. One should take care that the programmer time spent optimizing does not exceed the time gained by the optimizations.

References

[1] M. Brezinski. *Optical Coherence Tomography: Principles and Applications*. Springer, 2006. ISBN: 978-0-12-133570-0.

[2] W. Dexler and J.G. Fujimoto. *Optical Coherence Tomography: Technology and Applications*. 2nd ed. Springer, 2015. ISBN: 978-3-319-06419-2. DOI: 10.1007/978-3-319-06419-2.

[3] E. Hecht. *Optics*. 5th ed. Pearson, 2017. ISBN: 978-1-292-09693-3.

[4] A.F. Oskooi et al. "Meep: A flexible free-software package for electromagnetic simulations by the FDTD method". In: *Computer Physics Communications* 181 (2010), pp. 687–702.

[5] StarkEffects.com. 2006. URL: https://starkeffects.com/snells-law-vector.shtml (visited on 06/16/2021). This website only contains the final solution. I managed to re-derive these.

[6] Helmut Vogel. "A better way to construct the sunflower head". In: *Mathematical Biosciences* 44.3 (1979), pp. 179–189. ISSN: 0025-5564. DOI: 10.1016/0025-5564(79)90080-4.

[7] M. Wagner. *Deformations of Nerves and Neurons During Action Potentials*. Aug. 20, 2020.