Universiteit Utrecht

Bachelor thesis

Wavefront Shaping

Developing a demonstration setup

Author:
Suzan Marsman
Study: Physics & Astronomy

Supervisors:
Sanli Faez
Utrecht University
and
Jeroen Bosch
Utrecht University

8 July 2016

Abstract

Wavefront shaping is a promising technique invented circa ten years ago which enables us for example to focus light inside or behind a scattering material. To make this technique well-known we want to make a demonstration setup which can be used in presentations. The principle of our setup is that it reflects a laser beam off a Spatial Light Modulator (SLM), which modulates the spatial phase pattern of the light. We send the light through a scattering sample behind which we want to focus. By modulating the light with the SLM, we can change how the light comes out of the sample and in this way we can for example focus it in a certain spot. The setup we built is compact and portable, reproducable for an affordable price, and very safe to use, which are important assets for using it during a presentation. With our setup we can increase the intensity in the spot where we focused on with a factor of 6. This is an enhancement, but you can't see it by eye, which would be nice for a presentation. The main limitation for increasing the enhancement appears to be the signal-to-noise ratio.

Contents

Chapter 1

Introduction

The propagation of light makes it possible to see objects. In history a lot of devices like telescopes, microscopes and cameras were invented to improve the vision and to make it possible to see very small or far things. After those inventions, we were limited to the transparent media. But we never thought we would be able to look through media in which the light scatters. However, circa ten years ago, scientists developed a very promising technique which makes it possible to control the propagation of light through such a medium, called wavefront shaping [1]. In wavefront shaping you modulate the light before you send it into the medium in such a way that it comes out in the way you want to. This way, we are able to focus the light behind or even inside a medium.

Focusing behind or inside a sample is just one of the applications of wavefront shaping. Since the technique was developped, it is used in the fields of deep tissue microscopy [2], endoscopy [3], optical trapping [4], super-resolution imaging [5], nano-positioning [6] and cryptography [7].

Our goal is to make a demonstration setup to show people wavefront shaping. Therefore we want a setup which clearly shows the effect of wavefront shaping. Furthermore, the setup should be portable and safe, so that we can bring it to presentations. We also want to make it reproducible for an affordable price, so that for example institutes like schools can reproduce the setup to learn people about wavefront shaping.

We build a setup which sends a laser beam through a scattering sample, e.g. a piece of paper, and modulate the light in such a way that we focus the light behind it. This is a good application of wavefront shaping for the demonstration setup, because people can see the effect of wavefront shaping by eye. Namely, they can see that the spot we focused on is much brighter than the other spots.

Chapter 2

Theory

Light can travel through many mediums. Through homogeneous transparent mediums, like glass, it will travel in a straight line. On the interface between transparent mediums with different refractive indices the angle of refraction will be different from the angle of incidence (as long as the angle of incidence is not 0 degrees) and the angle will be determined by Snell's law [8]:

$$\frac{\sin(\theta_i)}{\sin(\theta_r)} = \frac{n_1}{n_2} \tag{2.1}$$

with θ_i the angle of incidence, θ_r the angle of refraction and n_1, n_2 the refractive indices of the respective medium.

But the path of light is not always this predictable. If you look at opaque mediums, the light scatters. For example if you look through mist, you can't see as far as when there is no mist. This is the result of the light hitting the water drops which change the direction of the light. Because the water drops are distributed randomly in space, the light scatters in random directions. Close by you can still see very clearly through mist, since the distance between water drops is relatively large and the light doesn't hit a lot of water drops, but if you look further the light has hit more water drops and has diffused more. The average distance traveled by light without hitting a water drop is called the mean free path.

In other opaque mediums like ground glass or paper, the principle is the same. In a ray-optics picture the light hits the particles in the material and as a result changes direction. By changing the local phase, the intensity in the speckles changes due to interference. The size and shape of inhomogeneity of the medium determines how strongly the light scatters as it propagates.

2.1 Light interference

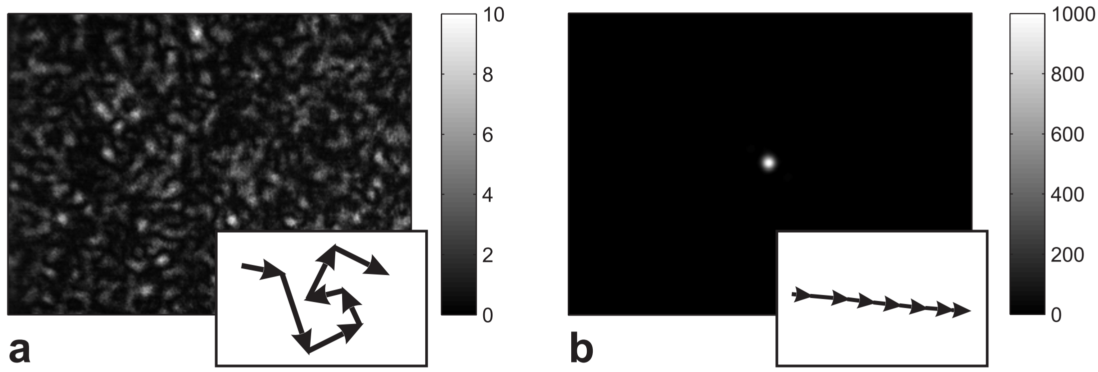


Figure 2.1: Interference in an opaque lens. **a** Transmitted intensity of an unshaped incident beam. Scattered light forms a random interference pattern known as laser speckle. Inset (phasor plot), at each point many waves interfere randomly, resulting in a low overall intensity. **b** Transmitted intensity of a wavefront that is optimized for focusing at a single point. The intensity in the focus is approximately 1000 times as high as the average intensity in **a**. Inset, in the focus all waves are in phase. Figure and caption from [9].

If you shine monochromatic light at an opaque medium, it comes out as a speckle pattern, see figure 2.1. This is the result of the light scattering inside the medium. As a result of the interference of the scattered light, some spots behind the medium have a high intensity and other spots have a low intensity. These are the spots of the speckle pattern. Since the light can scatter in so many ways depending on the inhomogeneity of the material and the spatial phase of the light, the speckle pattern looks totally random, but it actually is completely determined by the medium and incoming light. As long as the sample and the light source are stable in time the speckle pattern is always the same.

By modulating the spatial phase pattern of the light you send in, you can change the path the light follows and the phase of the light leaving the medium. Because the light comes out with a different phase, the intensity in the speckles changes as a result of interference. This is called wavefront shaping. Using this method the intensity of the light in a single speckle can be enhanced by a factor larger than 100 [1].

2.2 Speckle enhancement

If we want to optimize the intensity in a single speckle it is easiest to talk about it quantitatively. It is common to use the quantity enhancement η which is defined as [9]

$$\eta = \frac{\hat{I}_\beta}{\langle \hat{I}_\beta \rangle} \tag{2.2}$$

with $\hat{I}_\beta$ the intensity in the target speckle spot after optimizing and $\langle \hat{I}_\beta \rangle$ the reference intensity, which is the ensemble average of the intensity in a speckle before optimization.

In a strongly scattering medium the maximal enhancement that can be achieved can be calculated by [10]

$$\eta = \alpha(N-1) + 1 \tag{2.3}$$

with N the number of independent controllable channels (in our case the number of superpixels) and $\alpha \in (0, 1]$. If you can modulate the light for both phase and amplitude separately, $\alpha = 1$. This is the highest achievable level of control, because in this case we have full control of the incident field of light. In our setup, we use binary phase modulation, which means that we optimize the modulation for only two different phases. For this control level $\alpha = \frac{1}{\pi}$ [10]

Chapter 3

Set-up

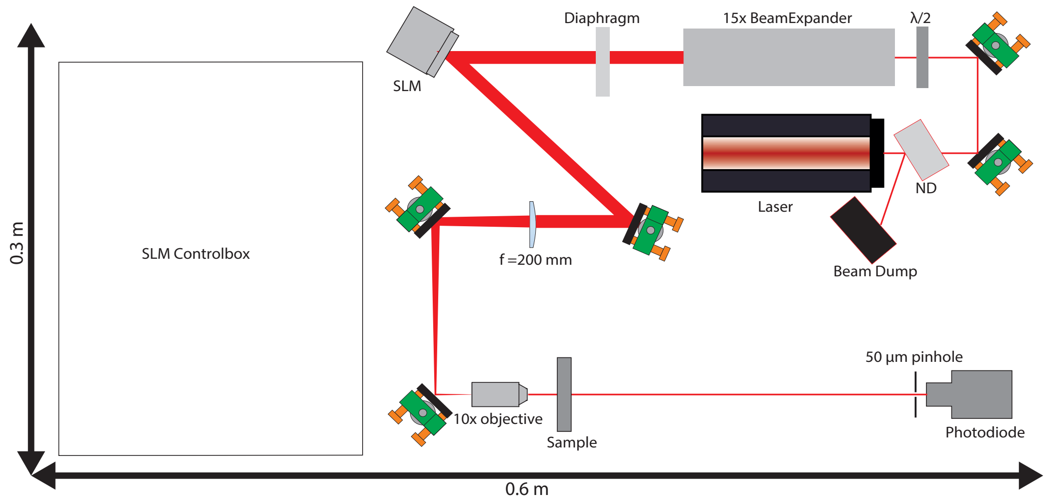


Figure 3.1: Schematic of the setup. The red line is the path of the light. Our light source is a 640 nm laser. ND, Neutral Density filter. SLM, Spatial Light Modulator.

To demonstrate wavefront shaping we built a setup which is able to focus light through an opaque sample, for example ground glass or paint. In this chapter we will discuss the design choices we made and the elements used. Afterwards we will explain the control program and finally we will tell something about the safety concerns.

Because we want to use the setup during presentations it should meet these constraints:

1. The demonstration setup should be small and movable. For this purpose we chose a breadboard of only 0.3 x 0.6 m.
2. It should be safe to be around. There might be a lot of people in the room, which do not all know the safety precautions about working with lasers. Therefore we need to make sure that all the stray light is properly shielded.
3. The effect of wavefront shaping should be really clear. We want that the speckle we focus on has a sufficient enhancement so that we can clearly see that this speckle is much brighter than the ones around.

The design is shown in Figure 3.1. The laser beam goes through a ND filter, which reflects 90% of the light into a beam dump and lets 10% of the light go through. Then the light goes through a halfwave plate, is expanded by the Beam Expander and cut by a diaphragm. Then it is reflected off the Spatial Light Modulator (SLM), which is able to change the phase per pixel separately. The SLM is imaged on the sample by a lens (f = 200 mm) and a 10x objective, NA 0.25. After the sample there will be a speckle pattern, which depends on the sample and the way the SLM modulated the light. A photodiode measures the intensity in approximately one speckle because of the 50 μm pinhole which is in front of it, in which we maximize the intensity.

3.1 Laser

Since we want to show the wavefront shaping process to people we use a laser with visible light. This way people can directly see the effect of wavefront shaping. We use an iBeam smart laser with a wavelength of 640 nm, which is visible red light. The laser beam has a diameter of 1.0 mm. It is known from lasers of this type that they run most stable when they run at a high power. That's why we choose to let the laser run at 129 mW and attenuate the power with an ND filter. The laser is controlled through a serial connection.

3.2 Spatial Light Modulator

One of the most important processes in wavefront shaping, is evidently the shaping of the wavefront. This is possible with a Spatial Light Modulator (SLM). Two of the most common SLMs are a Digital Micromirror Device (DMD) and a Liquid Crystal Display (LCD).

A DMD has a chip which consists of circa one million micromirrors. These micromirrors all have two orientations which we will call the on state and the off state. By creating a binary amplitude pattern, combined with an optical Fourier-filter a DMD can modulate the amplitude and the phase of light.

An LCD consists of around a million liquid crystal cells. Liquid crystals have a rod-like molecular structure which gives them a certain orientation. If there is no electric field applied on the cells, they are in a twisted nematic phase (see figure 3.2a). If light comes in, the polarization of the light follows the orientation of the rods and rotates 90 degrees. But by applying an electric field on the cell the rods orientate themselves in the direction of the electric field (see figure 3.2b). Now if the light comes in, it doesn't rotate its polarization anymore. For different magnitudes of the electric field and therefore different orientations of the rods the light also gets different phase delays. This way an LCD modulates the amplitude and phase of the light. With a certain configuration of polarization of the light, it is possible to do a phase-mostly modulation [9].

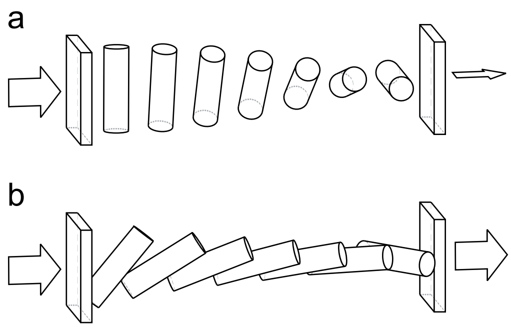


Figure 3.2: Operating principle of a transmissive TN LCD. **a** No voltage is applied and the rod-like liquid crystal molecules are ordered in a helix. The polarization of incident light follows the twist. **b** A voltage is applied between the two electrodes. The rods orient in the direction of the field and the twist disappears. The polarization of the incident light is not rotated. Figure and caption from [9].

If we compare the DMD with the LCD, the biggest differences are speed and resolution. The DMD is much faster than the LCD. The speed of the LCD is mostly limited because the liquid crystals need a certain time to settle, circa 50 ms. On the other hand, most LCDs have a higher resolution than DMDs. We finally chose to use an LCD because we had one available and it was fast enough for our setup.

The SLM we used is the Holoeye LC-R 2500 which has a reflective Liquid Crystal on Silicon (LCoS) chip, which works similar to an LCD. The screen is 19.5×14.6 mm^2 and has 1024×768 liquid crystal cells. The area per pixel is 19×19 μm^2. In our optimization method, we group the pixels in rectangular groups of pixels and call them superpixels, which we will treat as a single pixel. We use superpixels because we don't have time to modulate the light for every pixel separately. Also, the separate pixels slightly influence the pixels close to them. By using superpixels, most of the pixels close to a pixel are in the same superpixel and therefore should have the same settings. Furthermore the size of the superpixels also influences the signal-to-noise ratio.

To use the SLM optimally, we expand the diameter of the laser beam from 1 mm to 15 mm with a 15x beam expander. Between the beam expander and the SLM we have placed a diaphragm to regulate exactly how much of the SLM we use. Because of the way the chip of the SLM is mounted, the sides are not as stable as the inner part of the SLM. For that reason we decided to not use the upper- and lower-10% of the SLM. We cut the beam diameter with a diaphragm to about 12 mm.

3.3 Scattering sample

To make the wavefront shaping work we need to project the SLM on the sample. We decided that we wanted one SLM pixel per diffraction limited spot. We know from the specifications of the SLM that one pixel is 19 x 19 μm^2. We can calculate the size of a diffraction limited spot by using the formula [8]

$$D = \frac{\lambda}{2 \times NA} \tag{3.1}$$

with λ the wavelength of the light and NA the numerical aperture of the focusing objective. In our setup we use a laser with wavelength 640 nm and we chose to use a NA of 0.2. By using this, we can calculate that the size of the diffraction limited spot is $\frac{640}{2\times0.2} = 1600\text{nm} = 1.6\mu\text{m}$. This means that we need a magnification of $\frac{0.0016}{0.019} = 0.0842$, which is the inverse of a $\frac{1}{0.0842} = 11.875$x magnification. We chose to use a 10x objective which is close enough.

3.3.1 Sample Constraints

We can use all kinds of samples. We already mentioned ground glass and paint, but a lot of opaque material will do the trick. The most important things that should be taken into account concerning the choice of a sample are transmission and stability. If the sample doesn't have enough transmission, the intensity of a single speckle is too low to optimize. If the sample changes too much over time, for instance because it's moving or because it's living material like skin in which the structure of the tissue keeps changing, wavefront shaping won't be possible, because in the time the setup uses to do wavefront shaping the internal structure of the sample will be changed so much that the wavefront is not optimized anymore for the changed sample.

3.4 Photodiode

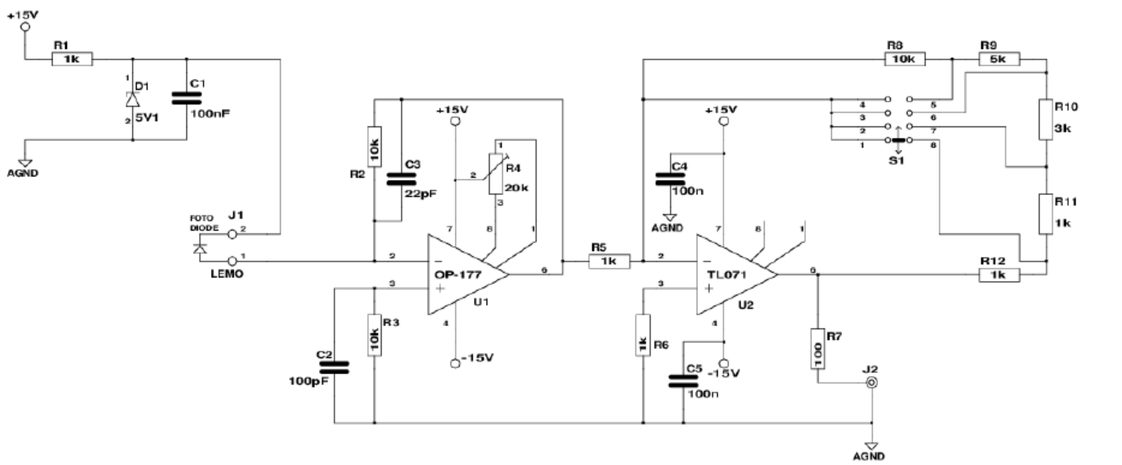


Figure 3.3: Schematic of the custom-made amplifier. We use an amplification of 2 $\times$ 10^7

To measure the intensity, we use a Thorlab's DET210 photodiode, which is a high-speed photo detector. We only measure the intensity of one speckle, because that's what we are focusing on. We can calculate the size of the speckle using the formula

$$size = \frac{\lambda d}{a} \tag{3.2}$$

in which λ is the wavelength of the light, d is the distance from the sample and a is the size of the beam in the focus. Therefore we estimate the size of the speckle at 70 mm from the sample to be $\frac{640\times10^{-9}\times70\times10^{-3}}{1*10^{-3}} = 44.8\times10^{-6}$ m. We therefore mounted a 50 μm pinhole in front of the photodiode.

The responsivity for light with a wavelength of 640 nm of this photodiode is approximately 0.40 A/W. Before the light hits the sample it has a power of around 5 mW. In the sample the light gets scattered into circa a million speckles and therefore the power of one single speckle is estimated to be 5×10^{-6} mW. Since this signal is far too low to convert with the Analog-to-Digital converter in the Red Pitaya, we connected the photodiode to an amplifier, see figure 3.3 for the schematic. The amplifier has an amplification of 2×10^{7}. So that the signal is such that the Red Pitaya can measure the analog signal and convert it to a digital signal.

A disadvantage of this construction to amplify the signal is that the amplifier not only amplifies the signal, but also brings along new noise. This is a big contribution to our noise level and reduces the signal-to-noise ratio a lot.

3.5 Red Pitaya

The photodiode measures the intensity of the speckle and convert it to a corresponding voltage, which is an analog signal. A computer can only have digital signals as input. Therefore we need an AD-converter. We chose to use the 14-bit AD-converter in a Red Pitaya. It has a sampling rate of 125 Ms/s, which is definetely fast enough. If it measures, it will automatically measure 16384 points. With a sampling rate of 125 Ms/s that would cover a time length of 131 μs which is not enough because of the refreshing rate of the SLM. Therefore we chose to adjust the sampling rate to 122.070 ksps, which covers a time length of 134 ms, so that we measure $\frac{134}{16} = 8.375$ refreshed screens of the SLM. The Red Pitaya has two modes, Low Voltage(LV) and High Voltage(HV). Using LV it can measure from -1 to 1 Volts and using HV it can measure from -20 to 20 Volts. Because it has a 14-bit AD-converter, it can measure $2^{14} = 16384$ different values. This means that at LV it makes steps of $\frac{1-(-1)}{16384} \approx 0.000122$ V which equals 0.122 mV and at HV it makes steps of 2.44 mV. After some tries of wavefront shaping, we saw that the voltage differences when changing the colour of a superpixel are of the order of a mV, for that reason, we chose to run the Red Pitaya at LV.

The Red Pitaya is connected to a computer, which will analyze the data and control the SLM.

3.5.1 Offset Red Pitaya

While wavefront shaping we noticed the Red Pitaya has a little offset in the voltage measurement. We mostly noticed this because when the power levels get really small, the Red Pitaya will measure a negative value. For that reason we set up a little experiment to determine this offset.

To find the offset we used a function generator, which we connected to the Red Pitaya and the oscillator to be able to say whether the offset was due to the Red Pitaya or the function generator. We let the function generator generate a square function with several amplitudes. For each amplitude, we measured the minimal and maximal value of the function on the oscilloscope and the Red Pitaya. In figure 3.4 we see the results of this measurement.

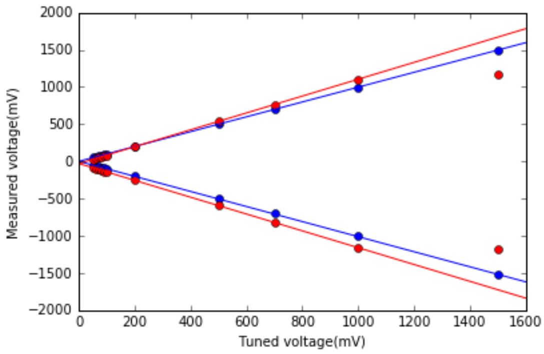


Figure 3.4: Measured voltage corresponding to a certain peak-to-peak intensity. Blue dots are the measurements of an oscilloscope and the red dots are the measurements of the Red Pitaya. The lines are the corresponding linear approximations

We decided that we trust the oscilloscope the most and therefore calculated the offset with respect to the data of the oscilloscope. From our data, we find a linear approximation would be the best approximation to calculate the offset. We don't include the last measurements of the Red Pitaya in the linear approximation, because for these intensities the Red Pitaya is already saturated. We find a constant offset of 29.75 mV and a linear offset of 0.89. We adjust our data for this offset.

3.6 Control Program

The computer gets the data from the photodiode via the Red Pitaya and will analyze these. It will compare the intensities which depend on the configuration of the SLM and will controll the SLM accordingly. To do this we use the programming language Python. See the Appendix for the code. We will first tell you about the connection with the Red Pitaya. Then we explain how we control the SLM and show you the methods. Lastly we will say something about the optimization speed.

3.6.1 Connecting to Red Pitaya

We can control the Red Pitaya by sending SCPI-commands to it. We first connect the Red Pitaya to the computer. Do this with the step-by-step plan of the appendix. Before you run the code always check that the IP-address in this line is correct:

```
rp_s = scpi.scpi('131.211.54.62').
```

To measure with the Red Pitaya we wrote the method ReadDiode() which tells the Red Pitaya to start measuring, send the data to the computer, take the second half of the data sent by the Red Pitaya and return the maximum value of the data. We only use the second half of the data sent by the Red Pitaya because the first half of the data sent by the Red Pitaya are the data of a previous measurement before which apparently are still in the memory of the Red Pitaya.

3.6.2 Controlling the SLM

The SLM is basically a second screen, which is connected to the computer with a DVI-cable (Digital Visual Interface). This makes it quite easy to control the SLM with Python, because we can simply create a window with the same amount of pixels as the SLM and place it at the second screen with

```
win.master.geometry('%dx%d+%d+%d' % (1024, 768, xSLMscreen, ySLMscreen)).
```

In which xSLMscreen and ySLMscreen correspond to the coordinates the second screen has to the settings of the computer's screen resolution. These are dependent on the size of the computer screen you are using. Then we only have to draw black and white squares on the window to colour the superpixels. The greyscale of a pixel will correspond to a certain phase delay.

3.6.3 Methods

The program has three different methods for wavefront shaping, named SLMGrating1() (based on the continuous sequential algorithm), SLMGrating2() and SLMGrating3() (both based on the stepwise sequential algorithm) [9].

SLMGrating1() starts with a completely white window. There is a nested loop, one for the x-coordinates and one for the y-coordinates, to go through all the superpixels. For every superpixel the method first checks whether the pixel is inside a circle in the middle of the SLM with a diameter corresponding to the beam diameter, if not, it skips that superpixel. If it is in the circle, it measures the intensities when the superpixel is white and when the superpixel is black. It compares the two intensities and set the superpixel in the colour corresponding to the highest intensity. Then it goes to the next superpixel.

SLMGrating2() also starts with a completely white window and also goes through all the pixels, checking whether it is in the circle and measuring the intensities when the superpixel is white and when it is black. The difference is the fact that it compares the two states, but it doesn't immediately set the superpixel in the colour corresponding to the highest intensity. Instead, after starting the method it creates a zero matrix, with a zero for every superpixel. If the intensity is higher when the superpixel is black, it changes the 0 corresponding with that superpixel in a 1. This continues for every superpixel. At the end, the method colours the superpixels which have a 1 in the matrix black and set the optimized pattern.

SLMGrating3() changes as well as SLMGrating2() the superpixels at the end in contrast to change them right away after measuring. The difference between SLMGrating2() and SLMGrating3() is that SLMGrating3() starts with generating a random matrix with zeroes and ones. And the SLM doesn't start with a completely white window, but with a window corresponding to the random matrix. Then again if the intensity is higher when the superpixel is white, it saves a 0 and if the intensity is higher when the superpixel is black, it saves a 1 as the corresponding matrix element.

3.6.4 Optimization speed

The speed of the experiments is bounded by two factors, the construction of the modulated wavefront and the connection with the Red Pitaya. We will discuss both in this section.

The construction of the wavefront has a few time consuming steps. At first the computer needs to send the data to the Spatial Light Modulator (SLM), the device that modulates the light, which takes about 10 ms. Then the SLM needs to settle which takes about 50 ms. Lastly, the SLM has a refreshing rate of 60 Hz, which means that the SLM will refresh circa every 17 ms. This means we will have to measure at least 17 ms and preferably synchronously with the refresh rate.

The Red Pitaya also results in a limit in the speed. Firstly, there are only a certain number of decimation options. The decimation reduces the sampling rate with a certain factor. Because the refreshing rate of 60 Hz we had to choose to measure for 134 ms. Furthermore, sending the measurement data from the Red Pitaya to the computer takes about 430 ms.

Adding all these time scales we end up with circa 0.7 s per measurement. Since we do the wavefront shaping binary in phase, it takes two measurements per superpixel. Therefore we can optimize around 43 pixels per minute.

3.7 Safety

Especially for a demonstration setup it's important to be aware of the safety risks. We will have a lot of people around the setup and don't want any of them to get hurt. The laser we used is a class 3B laser, which means that it can damage the eyes by looking in it directly, but diffuse reflections are not harmful. Also the laser is not hazardous for the skin.

To bring the laser power back to a reasonable value we put a reflective ND filter with an optical density of 1.0 right behind the laser. This ND filter will thus transmit 10% of the light, which we will use, and it will reflect 90% of the light, which we will send directly into a beam dump which will absorb the beam. This way the safety is kept in proportion as long as you do not look directly in the laser. Additionally we tried to cover the highest orders of diffraction, which have the highest intensities, as much as possible.

These precautions make sure that people won't be able to look into a beam which has a higher power than 5 mW, which is acceptable to look at as long you don't stare at it. (I still have to check whether the setup stays below the 5 mW-level)

Chapter 4

Measurements and Results

4.1 Sample

For the experiments we have used ground glass. Ground glass is normal glass which doesn't have a flat surface, but a rough surface. The light scatters when going through, because of the rough surface, and the outgoing light is a speckle pattern. The big advantage of ground glass is that it is a really stable sample with a high transmission.

4.2 Enhancement measurement

To see whether the demonstration setup is able to focus the light behind the sample, we did a measurement where we tried to achieve an enhancement as high as possible. This section will show the measurement and the results of this measurement.

4.2.1 Conditions

We measured using the method SLMGrating1(). We used a 30 x 30 superpixel grid, but we only wanted to use the superpixels the laser shined on (see the 251 black superpixels in figure 4.1). We corrected the data points for the offset of the Red Pitaya by using the constant and linear offset calculated before.

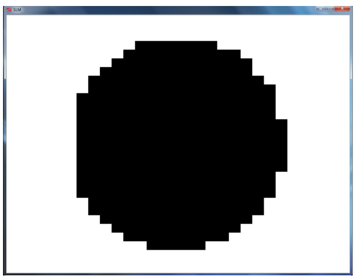


Figure 4.1: The window on the SLM, when all superpixels used in the measurement are black.

4.2.2 Results

We measured the intensity for all the 251 superpixels when it was white and when it was black leaving the others constant. We therefore had $2 \times 251 = 502$ measurements. Because we use the method SLMGrating1() we built up the window on the SLM and we see in figure 4.2 that the intensity is increasing step by step. The intensity we started with is circa 0.1 V and the intensity we ended with is circa 0.6 V. The enhancement we achieved is therefore 6.

4.3 Stability measurements

We also wanted to test the stability of our setup. We therefore did an experiment which is able to show us the impact of the noise on the signal. The modulation signal is the difference between the highest intensity and the lowest intensity in the observed speckle spot achievable by changing the

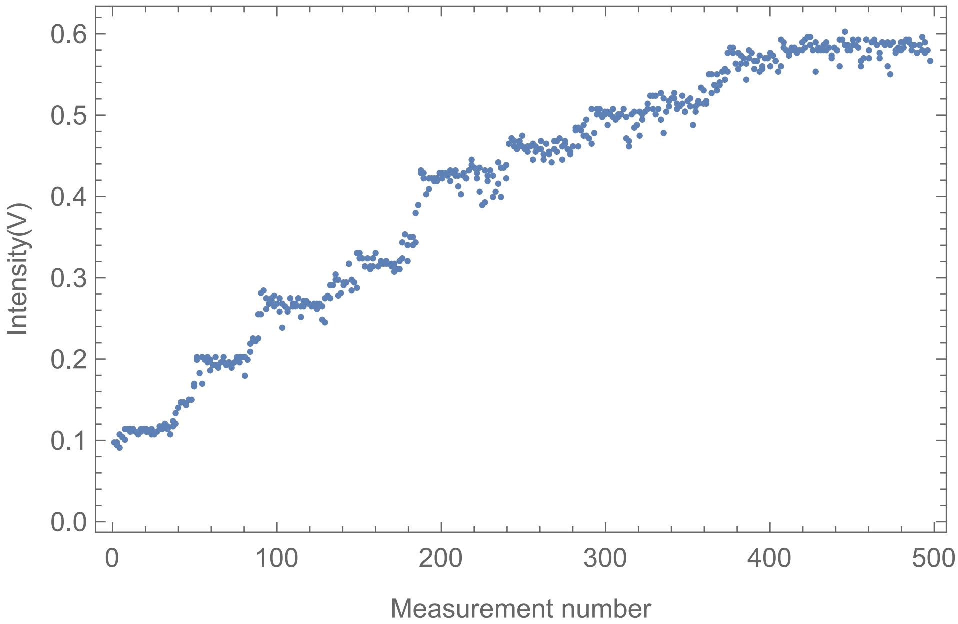


Figure 4.2: Transmission enhancement measurement. We measured for every superpixel the maximal intensity corresponding to the white superpixel and the maximal intensity corresponding to the black superpixel. (2 x 251 measurements). We see an enhancement of 6.

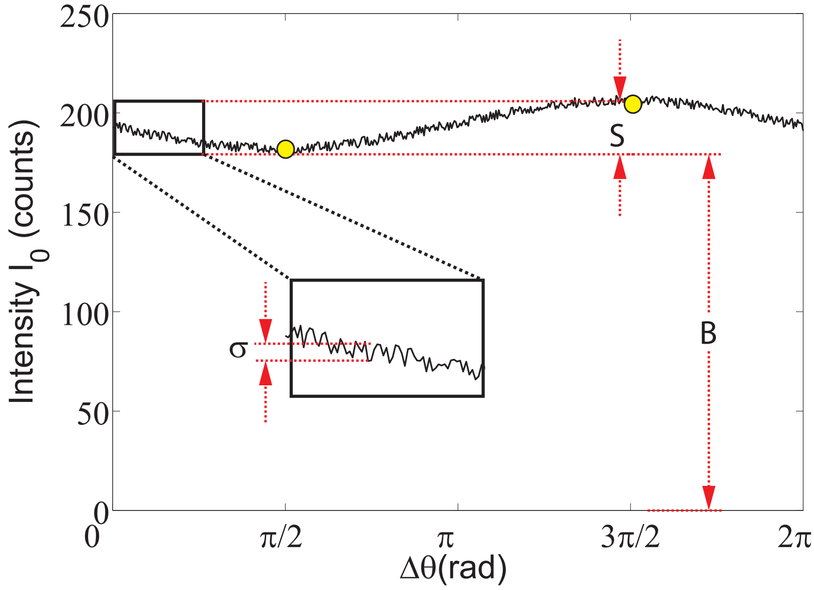


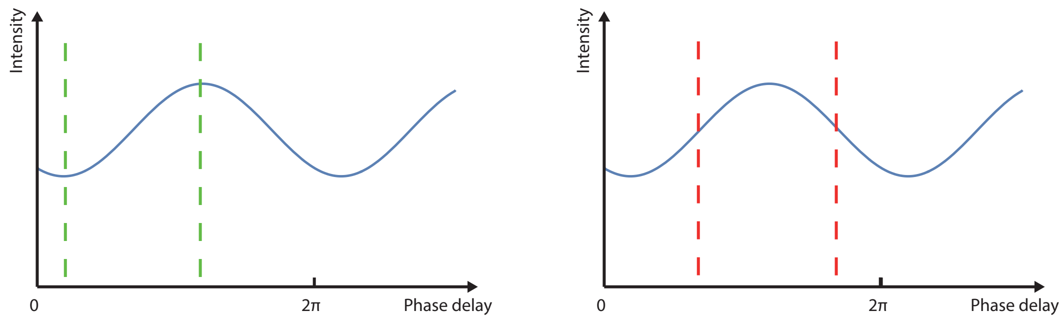
Figure 4.3: The effect of the noise on phase estimation. The target intensity I_0 is shown versus the phase delay $\Delta\theta$. The modulation signal S and the background B at the observed speckle spot are shown during the phase modulation $\Delta\theta$ of a single superpixel. The standard deviation of the noise is represented by σ. Figure adopted from [11].

phase delay. The standard deviation of the noise is the amplitude of the noise in the signal (see figure 4.3).

4.3.1 Conditions

For this experiment we chose a random superpixel in the middle of the SLM. For this superpixel we look at the modulation signal while alternating between a white and a black superpixel, while keeping the other superpixels constant. We repeated the measurement 20 times, so that we could statistically analyze our data. This way, we analyze the modulation signal and compare it to the noise.

When we would change the phase delay in the superpixel continuously, the intensity in the observed speckle spot would be a sinusoid depending on the phase delay. Because we use binary



(a) (b)

Figure 4.4: Intensity as a function of phase delay. In figure 4.4a, we see the best case scenario, wherein the white and black superpixel correspond to the highest and lowest intensity in the observed speckle. In figure 4.4b, we see the worst case scenario, wherein the white and black superpixel both give the same intensity in the observed speckle.

phase modulation wavefront shaping we only measure two values of the sine function. As a result of this there is a best case scenario and a worst case scenario for the wavefront shaping technique. Let's assume that the difference in phase delay for a white and a black superpixel is π, which is a reasonable assumption. Then the best case scenario would be that one of them results in the maximal intensity and the other results in the minimal intensity (see figure 4.4a). In this case choosing the right colour for this superpixel, and therewith the right phase delay, the enhancement would really benefit. In the worst case scenario both a white and a black superpixel would be in the equilibrium of the intensity sine function (see figure 4.4b). Choosing between the two of them wouldn't give us an intensity difference at all and therefore choosing between a white and a black superpixel would not increase the enhancement. Because the signal varies for each superpixel, depending on the optimal value of the phase delay and the measured values, we did the experiment for two different superpixels.

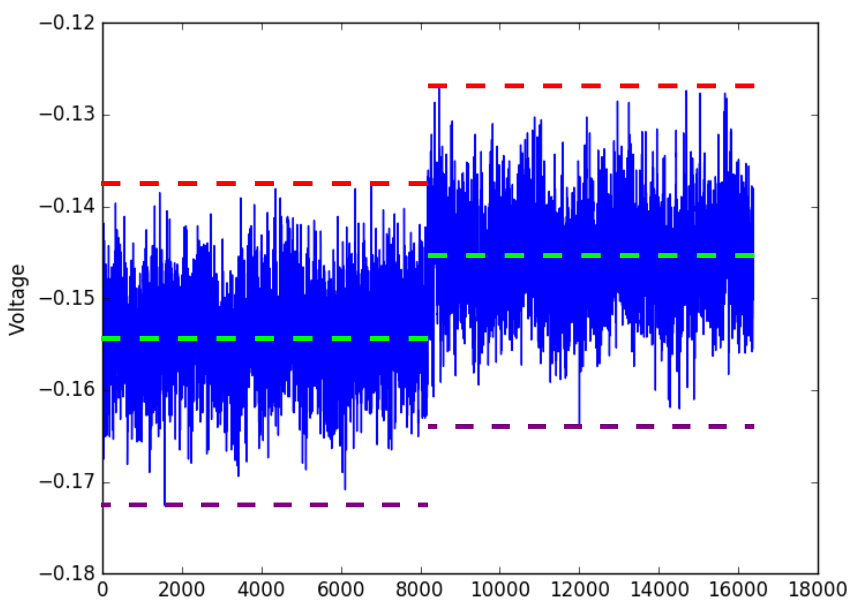


Figure 4.5: Measurement from the Red Pitaya while changing 1 out of the 251 superpixels from white to black. The purple lines gives the minimal value we measure, the green lines the average value and the red lines the maximal value.

We measured the signal with the Red Pitaya. We chose to analyze the last 8139 datapoints, because this covers a time buffer length of 66.67 ms, which is 4×16.67, the period of the refreshing of the SLM. We want a full amount of periods of the signal, because we then don't have to trigger

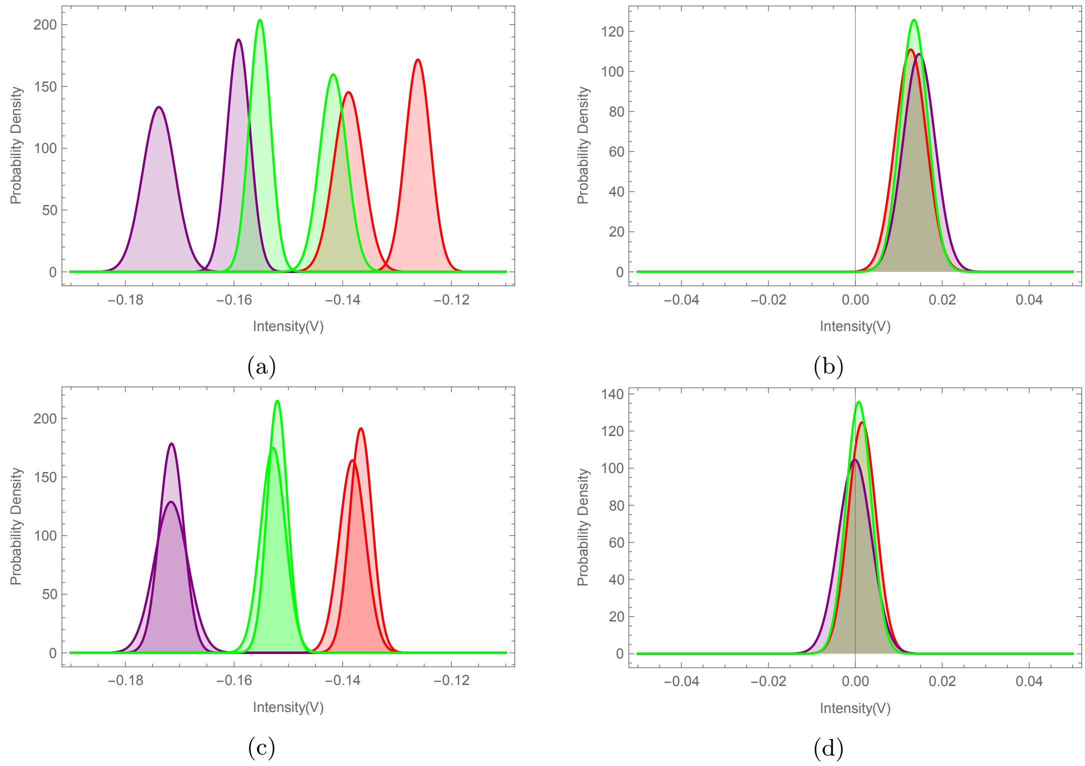


Figure 4.6: Measurements. Purple is corresponding to the minimal value, green is corresponding to the average and red is corresponding to the maximal value of the datapoints. **a)** Probability density graph of varying the first random pixel. One of the peaks is corresponding to the white superpixel, the other to the black superpixel. **b)**Probability density graph of the difference of a white pixel and a black pixel of the first random pixel. **c)** Probability density graph of varying the second random pixel. One of the peaks is corresponding to the white superpixel, the other to the black superpixel. **d)**Probability density graph of the difference of a white pixel and a black pixel of the second random pixel.

the signal to the SLM refreshing rate. We took the maximal value, the average and the minimal value, so that we could compare these three different ways to analyze the data (see figure 4.5).

4.3.2 Results

For every method we measured 20 times. From these 20 measurement points we can calculate the mean and the standard deviation. Assuming the data are distributed normally we can compare the data by comparing the normal distributions corresponding to our data. To calculate the chance of picking the colour which results in a lower intensity in the speckle we focus on (assuming the average method chooses the colour which results in a higher intensity), we will calculate the normal distribution of the difference of the white and black superpixel. Where μ is calculated by subtracting the means from each other and σ_μ is calculated by adding the standard deviations quadratically. We then calculate the chance that we choose the wrong colour, by taking the integral of the difference from $-\infty$ to 0.

From the first superpixel we measured we see in figure 4.6a that there is a significant intensity difference between the white superpixel and the black one, because for each measuring method we see that the normal distributions are barely overlapping. On the other hand from the second superpixel we see in figure 4.6c that there isn't a significant intensity difference between the white superpixel and the black one, because for each measuring method we see that the normal distributions are mostly overlapping.

In figure 4.6b we plotted the probability density of the difference of the white and black superpixel of the first superpixel and from this we can conclude that the intensities corresponding to the white and the black superpixel are significantly different. The chances to choose the wrong colour for the superpixel are very low (see table 4.1). On the other hand, in figure 4.6d we plotted the probability density of the difference of the white and black superpixel and from this we can conclude that the intensities corresponding to the white and the black superpixel aren't significantly

Table 4.1: Results

	Minimal value			Average value			Maximal value		
Superpixel	μ (V)	σ_μ (V)	P	μ (V)	σ_μ (V)	P	μ (V)	σ_μ (V)	P
1	0.0128	0.0036	0.00003	0.0147	0.0037	0.00001	0.0135	0.0032	0.00019
2	0.0016	0.0032	0.51	-0.0001	0.0038	0.39	0.00007	0.0029	0.31
3	0.0100	0.0037	0.0032	0.0090	0.0053	0.0447	0.0100	0.0031	0.0006

Table 4.2: Analyzed data. μ is the mean of the difference of the white and black superpixel, σ_μ is the standard deviation of μ and P is the chance of choosing the wrong colour.

different. The chance to choose the wrong colour for the superpixel are close to 0.5 (see table 4.1), which would be the case by random guessing.

Chapter 5

Discussion and Conclusion

5.1 Conclusion

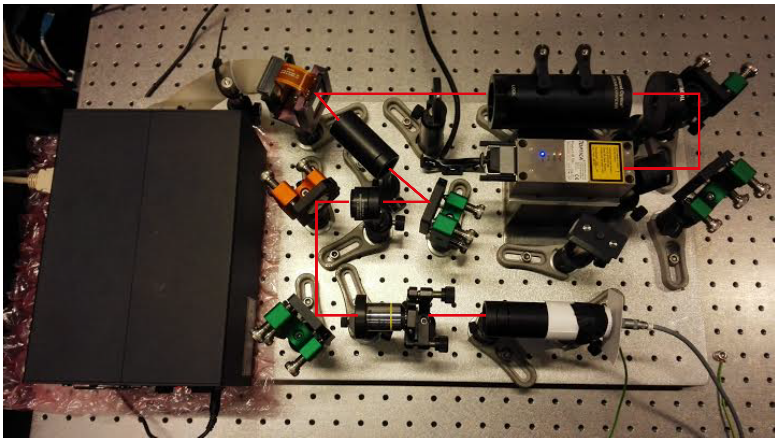


Figure 5.1: Photo of the setup. The red line shows the path of the light.

The goal was to make a demonstration setup for wavefront shaping. We built a setup (see figure 5.1) which can achieve an enhancement of 6, which isn't visible by eye, but definitely a good start. We can conclude from the stability measurement that the main problem is the signal-to-noise ratio, because for this reason a lot of the superpixels don't contribute to the enhancement at all. We can't really conclude whether it's better to take the maximal value, the average or the minimal value of the measured signal (see section 4.3.1), because the methods have similar standard deviations (see table 4.1).

Furthermore the setup is portable, because it's on a breadboard which is only 0.3 x 0.6 m^2. The setup is reproducible for an affordable price and there isn't any stray light which compromises the safety of the setup.

5.2 Possible improvements to the setup

Although we achieved an enhancement of the intensity, we couldn't make it visible by eye. There are some possible improvements. We concluded that the main problem of the setup is the signal-to-noise ratio at the moment. So to increase the enhancement we achieved we should increase the signal-to-noise ratio. We investigated where most of the noise comes from and found out that it is mostly due to the amplifier of the photodiode. We can think of two possible solutions for that. Firstly, we can increase the laser power. Because we don't need such a big amplification when we have a higher laser power, we don't get so much noise from the amplifier. Secondly, we can try to find another photodiode and/or amplifier which doesn't produce so much noise.

The setup is already quite affordable, but we still have some options to make it even cheaper. The laser can be replaced by a cheaper one, we only used this one because we had one available. The beam expander can be replaced by two lenses, which do the same, but it is slightly harder to align them in the setup. Also the SLM can be interchanged by a pocket projector, which is also an LCD and therefore can modulate the spatial phase pattern.

For safety reasons it would also be a good idea to build a box around the setup. This will also increase the stability of the setup, because it would get less stray light and the air flow would be minimized.

Appendix A

Connecting to the Red Pitaya

To connect the Red Pitaya to the computer use the following steps.

1. Make sure both the Red Pitaya and the computer are connected with the same network. To be sure, you can for example use a switch and connect both to it.
2. Go to www.redpitaya.com/quick-start/, log in (if you don't have an account you can very easily make one), and follow the steps.
3. You can now find the IP-address of the Red Pitaya under LAN IP.
4. Go to Putty and fill in the IP-address of the Red Pitaya
5. Log in with Putty, username: root, password: root
6. Run the following command using the remote connection:

```
$redpitaya>$ systemctl start redpitaya$\_$scpi $\&$
```

When they are connected you can run the program on the computer and get the data from the photo diode via the Red Pitaya.

Appendix B

Python code

Make sure you can reach the external packages, redpitaya_scpi and graphics, used in this code, you can find them on the internet.

```
# -*- coding: utf-8 -*-

import sys
from random import randint
from graphics import * #make sure you add this file
import time
import redpitaya_scpi as scpi #make sure you add this file
from numpy import mean
from math import ceil

rp_s = scpi.scpi('131.211.54.62')
xp = 1024 #number of SLMpixels in x direction
yp = 768 #number of SLMpixels in y direction
xsp = 10 #number of superpixels in x direction
ysp = 10 #number of superpixels in y direction
xssp = ceil(xp/xsp) #number of pixels in one superpixel in the x
    direction
yssp = ceil(yp/ysp) #number of pixels in one superpixel in the y
    direction
xSLMscreen = 1920 #x-coordinate of SLMscreen (see at screenresolution)
ySLMscreen = 0 #y-coordinate of SLMscreen (see at screenresolution)
rgbbg = 255 # colour from background (0 is black, 255 is white)
rgbo = 0 # other colour
SLMRadius = 300 # Radius of the circle used on the slm in SLMpixels
win = GraphWin("SLM", xp, yp) #creates the window
win.master.geometry('%dx%d+%d+%d' % (1024, 768, xSLMscreen, ySLMscreen)
    )#makes sures the window will be at the upperleft corner


Matrix = [] #starts a zero matrix corresponding to the number of
    superpixels
for i in range(ysp):
    Matrix.append([0]*xsp)

def main():
    rp_s.tx_txt('ACQ:DEC 1024') #sets the decimation of the Red Pitaya
    time.sleep(1)
    win.setBackground(color_rgb(rgbbg,rgbbg,rgbbg)) #makes SLMscreen
        white

def DrawPixel(x, y,colour): #Draws a pixel in the corresponding colour
    c = Rectangle(Point(xssp*x,yssp*y),Point(xssp*x+xssp,yssp*y+yssp))
    c.setFill(colour)
    c.setOutline(colour)
```

```
    c.draw(win)

def RandomGrating(): #sets a random grating on the SLM
    for x in range(xsp):
        for y in range(ysp):
            if randint(0,1) == 0:
                DrawPixel(x,y,colour_rgb(rgbo,rgbo,rgbo))

def InRadius(x,y): #bool whether a pixel is in the circle in the middle of the SLM or not
    if min((((x*xssp)-(xp/2))**2 + ((y*yssp)-(yp/2))**2)**0.5, (((x*xssp+xssp)-(xp/2))**2 + ((y*yssp)-(yp/2))**2)**0.5,(((x*xssp)-(xp/2))**2 + ((y*yssp+yssp)-(yp/2))**2)**0.5,(((x*xssp+xssp)-(xp/2))**2 + ((y*yssp+yssp)-(yp/2))**2)**0.5) < SLMRadius:
        return True
    else:
        return False

def ReadDiode():#gives the maximal intensity of the diode
    rp_s.tx_txt('ACQ:DEC 1024')
    rp_s.tx_txt('ACQ:START')
    rp_s.tx_txt('ACQ:TRIG NOW')

    while 1:
        rp_s.tx_txt('ACQ:TRIG:STAT?')
        if rp_s.rx_txt() == 'TD':
            break


    rp_s.tx_txt('ACQ:SOUR1:DATA?')
    buff_string = rp_s.rx_txt()
    buff_string = buff_string.strip('{}\n\r').replace("  ", "").split(',')
    buff = map(float, buff_string)
    buff2 = buff[-8000:]
    return mean(buff2)

def CheckPixel1(x, y):#Compares the intensity of a black and white intensity and keeps the pixel in the state where the the intensity is the highest
    time.sleep(1)
    white = ReadDiode()
    print white
    DrawPixel(x,y, color_rgb(rgbo,rgbo,rgbo))
    time.sleep(1)
    black = ReadDiode()
    print black
    time.sleep(1)
    if white > black:
        DrawPixel(x,y, color_rgb(rgbbg,rgbbg,rgbbg))

def SLMGrating1():#Checks all the pixels
    for x in range(xsp):
        for y in range(ysp):
            if InRadius(x,y):
                CheckPixel1(x,y)

def CheckPixel2(x,y):#Compares the intensity of a black and white intensity and returns the pixel white.
    white = ReadDiode()
    DrawPixel(x,y, color_rgb(rgbo,rgbo,rgbo))
```

```
        black = ReadDiode()
        DrawPixel(x,y, color_rgb(rgbbg,rgbbg,rgbbg))
        if black > white:
            Matrix[x][y] = 1

def SLMGrating2():#Checks all the pixels and draws the matrix
    for x in range(xsp):
        for y in range(ysp):
            CheckPixel2(x,y)
    DrawMatrix(Matrix)

def DrawMatrix(matrix):#Draws a matrix (1 = black, 0 = white)
    for x in range(xsp):
        for y in range(ysp):
            if matrix[x][y] == 1:
                DrawPixel(x,y,color_rgb(rgbo,rgbo,rgbo))

def CheckPixel3(x,y):#Does the same as CheckPixel2, but starts with a
    matrix in stead of just a white screen
    if Matrix[x][y]==0:
        white = ReadDiode()
        DrawPixel(x,y, color_rgb(rgbo,rgbo,rgbo))
        black = ReadDiode()
        DrawPixel(x,y, color_rgb(rgbbg,rgbbg,rgbbg))
        if black > white:
            Matrix[x][y] = 1
    if Matrix[x][y]==1:
        black = ReadDiode()
        DrawPixel(x,y, color_rgb(rgbbg,rgbbg,rgbbg))
        white = ReadDiode()
        DrawPixel(x,y,color_rgb(rgbo,rgbo,rgbo))
        if white > black:
            Matrix[x][y] = 0

def SLMGrating3():
    print Matrix
    for x in range(xsp):
        for y in range(ysp):
            CheckPixel3(x,y)
    print Matrix
    DrawMatrix(Matrix)

main()
# Choose method you want to use, choose SLMGrating1(), SLMGrating2() or
    SLMGrating3()
SLMGrating1()
```

Bibliography

[1] I. M. Vellekoop and A. Mosk, "Focusing coherent light through opaque strongly scattering media," *Optics letters*, vol. 32, no. 16, pp. 2309–2311, 2007.

[2] X. Xu, H. Liu, and L. V. Wang, "Time-reversed ultrasonically encoded optical focusing into scattering media," *Nature photonics*, vol. 5, no. 3, pp. 154–157, 2011.

[3] T. Čižmár and K. Dholakia, "Exploiting multimode waveguides for pure fibre-based imaging," *Nature communications*, vol. 3, p. 1027, 2012.

[4] T. Čižmár, M. Mazilu, and K. Dholakia, "In situ wavefront correction and its application to micromanipulation," *Nature Photonics*, vol. 4, no. 6, pp. 388–394, 2010.

[5] E. Van Putten, D. Akbulut, J. Bertolotti, W. Vos, A. Lagendijk, and A. Mosk, "Scattering lens resolves sub-100 nm structures with visible light," *Physical review letters*, vol. 106, no. 19, p. 193905, 2011.

[6] E. Van Putten, A. Lagendijk, and A. Mosk, "Nonimaging speckle interferometry for high-speed nanometer-scale position detection," *Optics letters*, vol. 37, no. 6, pp. 1070–1072, 2012.

[7] R. Horstmeyer, B. Judkewitz, I. M. Vellekoop, S. Assawaworrarit, and C. Yang, "Physical key-protected one-time pad," *Scientific reports*, vol. 3, 2013.

[8] *University Physics*. Pearson, 2012.

[9] I. M. Vellekoop, *Controlling the propagation of light in disordered scattering media*. PhD thesis, University of Twente, 2008.

[10] I. M. Vellekoop, "Feedback-based wavefront shaping," *Optics Express*, vol. 23, pp. 12189–12206, May 2015.

[11] H. Yılmaz, W. L. Vos, and A. P. Mosk, "Optimal control of light propagation through multiple-scattering media in the presence of noise," *Biomed. Opt. Express*, vol. 4, pp. 1759–1768, Sep 2013.