SORBONNE UNIVERSITY

M1 INTERNSHIP REPORT

Speckle Statistics

Ozgen Tunc Turker

supervised by
Hilton B. de AGUIAR

July 3, 2020

Contents

List of Figures

Speckle Statistics

ABSTRACT

This project is worked as an internship project between the dates 14/02/2020 and 28/06/2020 at the Laboratoire Kastler Brossel, in the Complex Media Optics Lab Group. Because of the Covid-19 the internship mostly is performed remotely.

The subject is mainly about finding a more efficient way to use the speckles in the non-linear microscopy imaging. Because speckles are obeying Rayleigh distribution, non-linear applications cause to loss most of the useful part of the speckle. To finding a efficient use of the speckles, tailoring the statistics of it to a sub-Rayleigh statistics can be a solution. In this project this possibility is investigated. Unfortunately, the project can not be completed in the period of the internship.

Chapter 1

Introduction

Optical microscopy is a centuries old technique in which illuminated sample and system of lenses are used. With Ernst Abbe's words "The microscope image is the interference effect of a diffraction phenomenon"[12]. Conventional optical microscopy is limited with "diffraction limit" which is the consequence of the natural phenomenon *diffraction*. Diffraction limit which is is also called *Abbe diffraction limit*, was first introduced by Ernst Abbe in 1873. *Abbe diffraction limit*:

$$d = \frac{\lambda}{2n\sin\theta} \tag{1.1}$$

Optical microscopes are not able to resolve dimensions smaller than d as it is defined in equation 1.1 where λ is the wavelength of the light used for illuminating the specimen and n is reflective index of the medium in which light is carried and θ is the half angle of the light cone enters the objective of the microscope. Figure 1.1

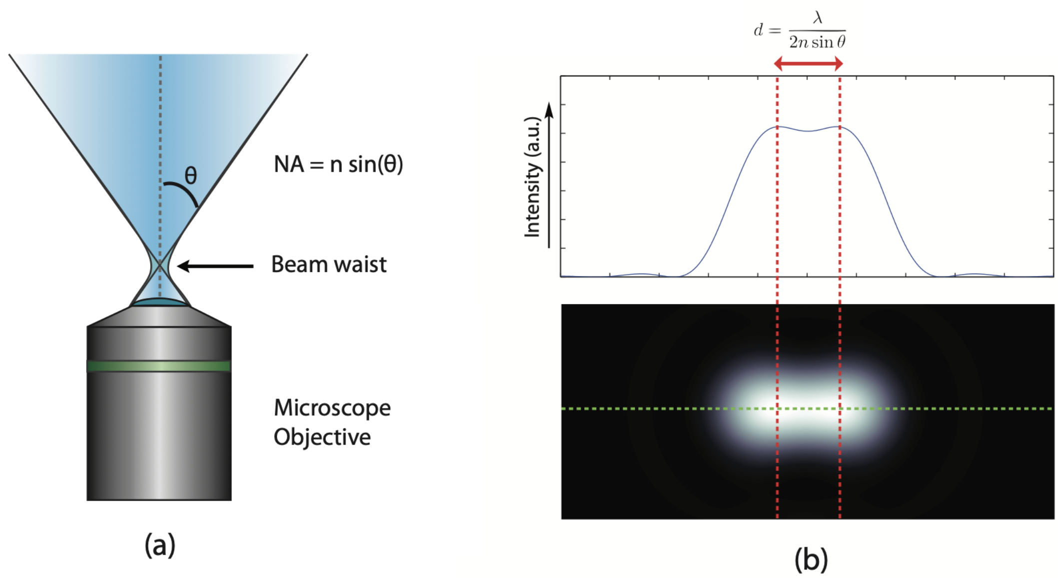


Figure 1.1: Demonstration of the diffraction limit [2].

But in any field, discovering the beyond the limits is the aim of the science. Hence in optical microscopy there are some techniques to achieve to go the beyond the diffraction limit, they are called “Super-resolution microscopy" techniques. *4Pi*, *Structured illumination microscopy (SIM)*, *Stimulated emission depletion (STED)* are some known techniques [11].

The project done in this internship is only about blind structured illumination technique which can be classify as a sub-technique under the Structured illumination microscopy (SIM).

1.1 Microscopy Imaging

Microscopy can be seen as a technique that brings the high spatial frequency objects to the low frequency region. Mathematically, spatial frequency domain is Fourier domain and the passage from spatial coordinates to the spatial frequency domain is Fourier transformation.

On the other hand, imaging is expressed mathematically as follows:

$$I(x, y) = \big[\rho(x, y) \times I_{ill}(x, y)\big] * PSF(x, y) \tag{1.2}$$

Here, $I(x, y)$ is the image function, $\rho(x, y)$ spatially distributed object, $I_{ill}(x, y)$ is the function of illumination and PSF is the point spread function which is the impulse respond of the system. Effect of the PSF on an image can be seen in the figure 1.2.

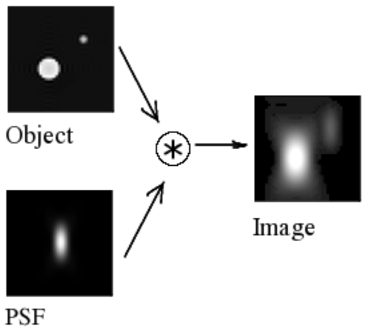


Figure 1.2: Convolution of object and point spread function

If image is Fourier transformed, $\tilde{I}(k_x, k_y)$ is obtained as follows

$$\tilde{I}(k_x, k_y) = \big[\tilde{\rho}(k_x, k_y) * \tilde{I}_{ill}(k_x, k_y)\big] \times OTF(k_x, k_y) \tag{1.3}$$

Here OTF called optical transfer function, is Fourier transform of PSF. As PSF is the impulse respond of the system, OTF is defined in Fourier plane and from the area it covers, the resolution capability of the system is known.

As well as the Fourier transformed image ($\tilde{I}(k_x, k_y)$) rises the information about spatial resolution, by changing the convolution to the multiplication it helps to get rid of the foreknown PSF in the formulation.

As it is shown in the figure 1.3, in the conventional microscopy, maximum k values in the spatial frequency space is limited by the half of the wavelength of the illumination light.

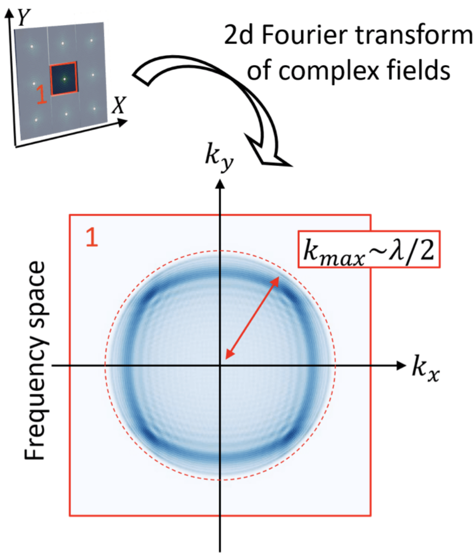


Figure 1.3: Representation in k-space [8].

By manipulating frequency domain helps us to reveal or hide the details in the object. A good example of is given by Bahaa E. A. Saleh and Malvin Carl Teich, figure 1.4.

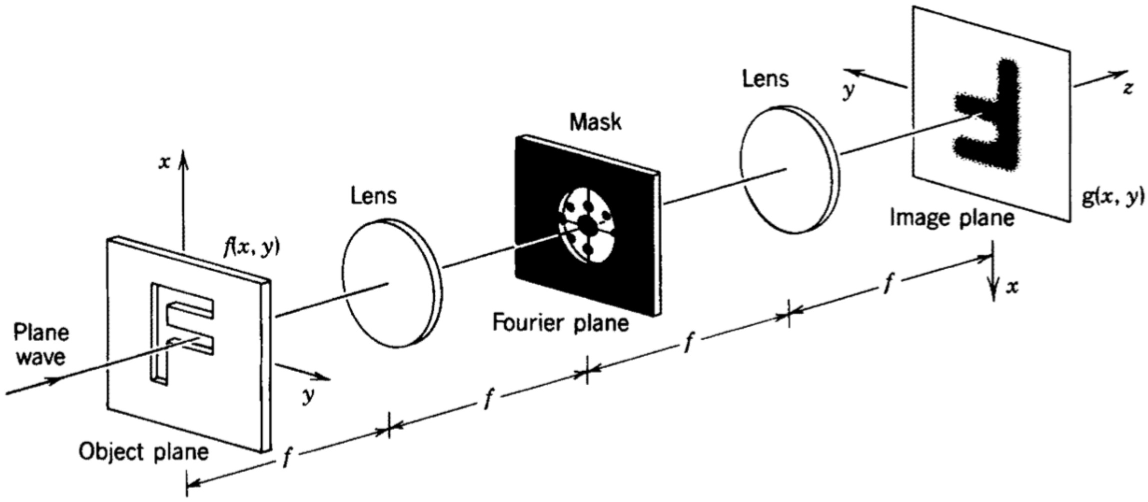


Figure 1.4: Demonstration of the manipulation in Fourier domain [10].

Here, the Fourier transformation and inverse Fourier transformation are physically done by lenses. In the Fourier plane (Fourier domain), larger k values are masked by a filter. So the information in the higher spatial frequency is lost and less detailed image is constructed in the image plane.

1.2 Structured illumination microscopy (SIM)

To go to super-resolution microscopy, the k-space seen in the figure 1.1 is needed to expand. In 1995, Mats Gustafsson became the first person came up with a new idea which ended by the ability of doubling the resolution of the diffraction-limited system. His idea was illuminating the specimen with predetermined illumination pattern which can encode the higher spatial frequencies [5]. What this states mathematically is that the $I_{ill}(x, y)$ in the equation 1.2 is not a uniform illumination as it is used in conventional microscopy but a periodic function which is well defined before.

The method used for the SIM can be understood as *Moiré Fringes*. In the figure 1.5, (a) shows how Moiré fringes are created from line patterns, (b) shows the same the same shape observed in figure 1.3 Fourier plane of the conventional microscope, (c) shows 3 components in the Fourier domain this is because illumination pattern is sinusoidal, (d) shows convolution of the illumination and the object as it is seen in the equation 1.3 without OTF, finally (e) shows this effect seen in (d) for different phase illuminations. To conclude it can be said that structured illumination microscopy shifts the OTF of the system. It is also mentioned that with this technique resolution is doubled than the conventional microscopy's [4].

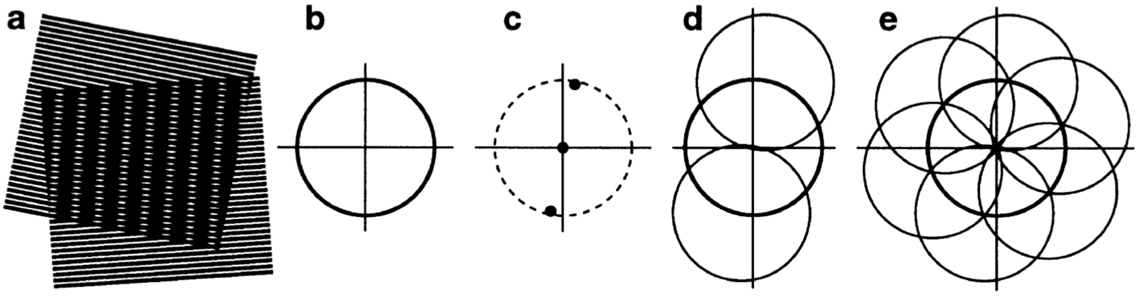


Figure 1.5: Theoretical explanation of the SIM [4]

Before passing to the Blind-SIM (blind structured illumination microscopy), it is better to mention the drawback of the SIM. SIM requires highly determined illumination pattern and such level of knowledge is not an easy thing to built and to maintain [7], a small errors ends up with a yielding artefacts on the final image [9]. That's why the specimen should be thin or its reflective index should be small [6].

1.3 Blind Structured Illumination Microscopy (Blind-SIM)

Blind-SIM primarily refers to lack of knowledge about the illumination light, indeed no knowledge about it. For a illumination pattern with any a priori knowledge, speckles become very well candidates.

1.3.1 Speckles

Speckles are granular patterns which can be seen on a reflective surface when a laser beam hits on that surface. First it can be said that this is because of the roughness of the reflective surface but any surface can be considered rough respect to the coherent wave-front. Moreover speckle can be seen when laser light is passing through a stationary medium or a particle suspension. So the reason behind it is the different optical paths taken by coherent light rays [3]. The granular shape of the speckle can be seen in the figure 1.6.

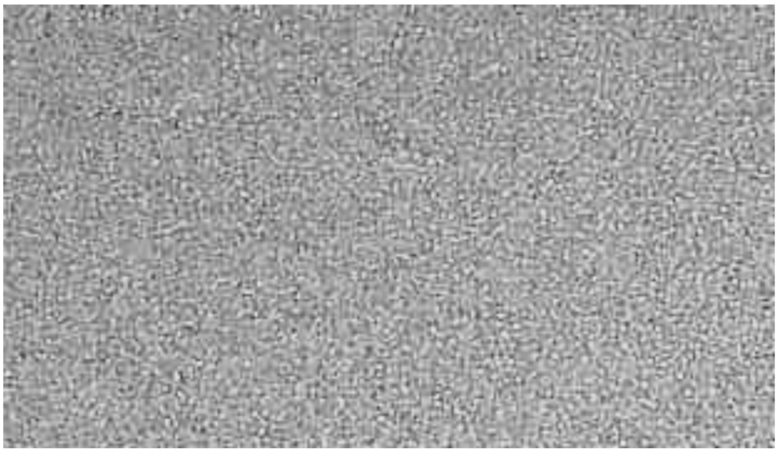


Figure 1.6: Magnified speckle pattern [3].

1.3.2 Speckle Statistics

Speckle appears when the signal has "multitude of independently phased additive complex components" composition. [3]. So it can be represented as random phasor sums. To do so let's first express random phasor sums as a typical sinusoidal signal

$$\boldsymbol{A} = A(x, y, t)\cos(2\pi\nu_0 t - \theta(x, y, t)) \tag{1.4}$$

Where $A(x, y, t)$ is amplitude, ν_0 is carrier frequency and $\theta(x, y, t)$ is phase. Complex representation of such signal can be

$$\boldsymbol{A} = A(x, y, t)e^{-i(2\pi\nu_0 t - \theta(x,y,t))} \tag{1.5}$$

Because carrier frequency is same for all components of the speckle (coherent light), it is acceptable to write the it as follows:

$$\boldsymbol{A} = A(x, y, t)e^{i\theta(x,y,t))} \tag{1.6}$$

$$\boldsymbol{A} = Ae^{i\theta} \tag{1.7}$$

Now it can be easily expressed by phasors:

$$\boldsymbol{A} = Ae^{i\theta} = \frac{1}{\sqrt{N}}\sum_{n=1}^{N}\boldsymbol{a_n} = \frac{1}{\sqrt{N}}\sum_{n=1}^{N}a_n e^{i\phi_n} \tag{1.8}$$

Here is the time to do some assumptions,

1. a_n and ϕ_n are statistially independent

2. ϕ_n is uniformly distributed between $[-\pi, \pi]$

Now checking the first and second moment, as it is calculated by separating real and imaginary parts it is seen that real part has a $\cos\phi_n$ and imaginary part has a $\sin\phi_n$ which leads to 0 for first moment of the both parts.

$$Re\{\boldsymbol{A}\} = \frac{1}{\sqrt{N}}\sum_{n=1}^{N} a_n \cos\phi_n = R \tag{1.9}$$

$$Im\{\boldsymbol{A}\} = \frac{1}{\sqrt{N}}\sum_{n=1}^{N} a_n \sin\phi_n = I \tag{1.10}$$

If second moment are also written in separately,

$$\sigma_R^2 = E[R^2] = \frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{M} E[a_n a_m]\cos\phi_n\cos\phi_m \tag{1.11}$$

$$\sigma_I^2 = E[I^2] = \frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{M} E[a_n a_m]\sin\phi_n\sin\phi_m \tag{1.12}$$

$$\tag{1.13}$$

a=m for having non zero result. So

$$\sigma_R^2 = \frac{1}{N}\sum_{n=1}^{N} E[a_n^2] = \sigma_I^2 \tag{1.14}$$

$$\tag{1.15}$$

When N goes infinity, from the central limit theorem, joint probability distribution of the R and I can be taken as a Gaussian distribution.

$$P_{R,I}(R, I) = \frac{1}{2\pi\sigma^2}\exp\big(-\frac{R^2+I^2}{2\sigma^2}\big) \tag{1.16}$$

here $\sigma^2 = \sigma_R^2 = \sigma_I^2$. As it is known $A = \sqrt{R^2+I^2}$ and $\theta = \arctan(I/R)$ transformation from equation 1.16 to the joint probability of amplitude and phase ($P_{A,\theta}$) can be done by jacobian

$$P_{A,\theta} = P_{R,I}(R, I)||J|| = P_{R,I}(A\cos\theta, A\sin\theta)||J|| \tag{1.17}$$

$$||J|| = \begin{vmatrix} \cos\theta & -A\sin\theta \\ \sin\theta & A\cos\theta \end{vmatrix} \tag{1.18}$$

because $||J||$ is 1:

$$P_{A,\theta}(A, \theta) = \frac{A}{2\pi\sigma^2}\exp\big(-\frac{A^2}{2\sigma^2}\big) \tag{1.19}$$

for $A >= 0$ and $-\pi <= \theta < \pi$. By finding its marginal statistics, Rayleigh distribution is found.

$$P_A(A) = \int_{-\pi}^{\pi} P_{A,\theta}(A, \theta)d\theta \tag{1.20}$$

$$P = \frac{A}{\sigma^2}e^{-\frac{A^2}{2\sigma^2}} \tag{1.21}$$

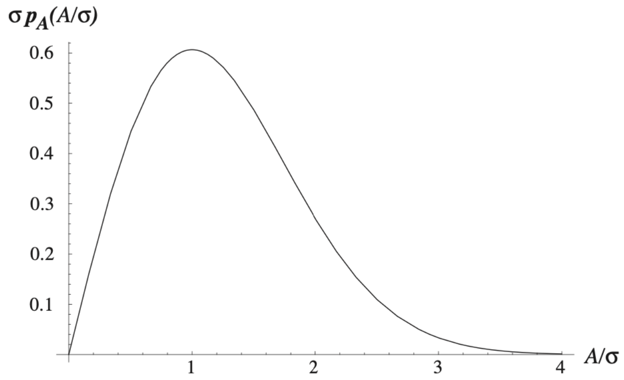


Figure 1.7: Rayleigh distribution [3]

In the following chapters σ^2 is taken 1 and A^2 is defined as intensity I

1.4 Aim & Motivation of The Project

In the project, tailoring the speckle statistics and obtaining any target statistics rather than a Rayleigh is aimed. The project is not realised experimentally but in principle its setup consists only one SLM and a lens. SLM is a spatial light modulator device which creates speckle pattern according to the phase input. So if random phase is given to SLM then it creates speckle pattern which its intensity obeys Rayleigh statistics.

Motivation of the project is to use speckle patterns in the non linear imaging. Because of the steep slope of the Rayleigh distribution (e^{-I}), a non-linear application comes to the loosing lots of speckle grains. In the figure 1.8 the loss can be easily seen when a random speckle intensity squared.

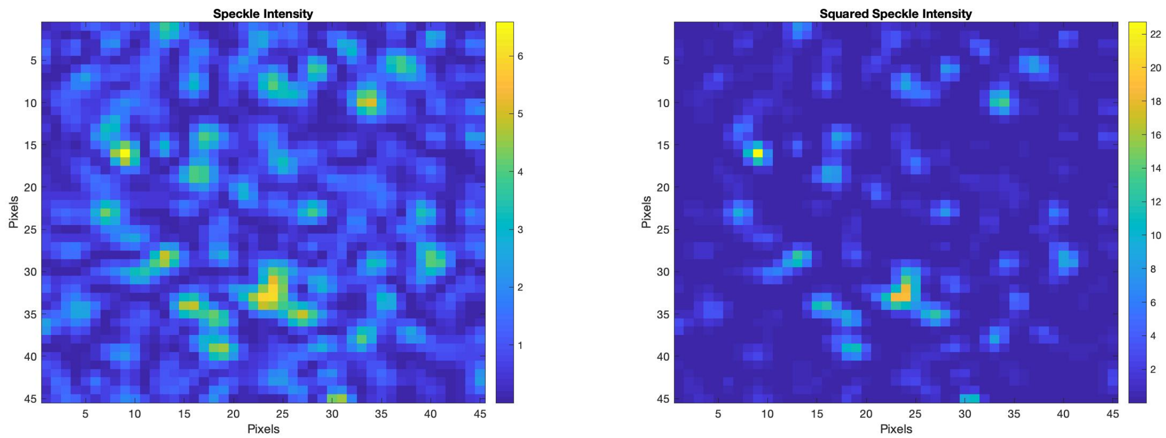


Figure 1.8: Speckle patterns are created from random phases in Matlab, in the images intensity of the speckle pattern is shown. On the right same speckle intensity is squared and loss of grains are obvious.

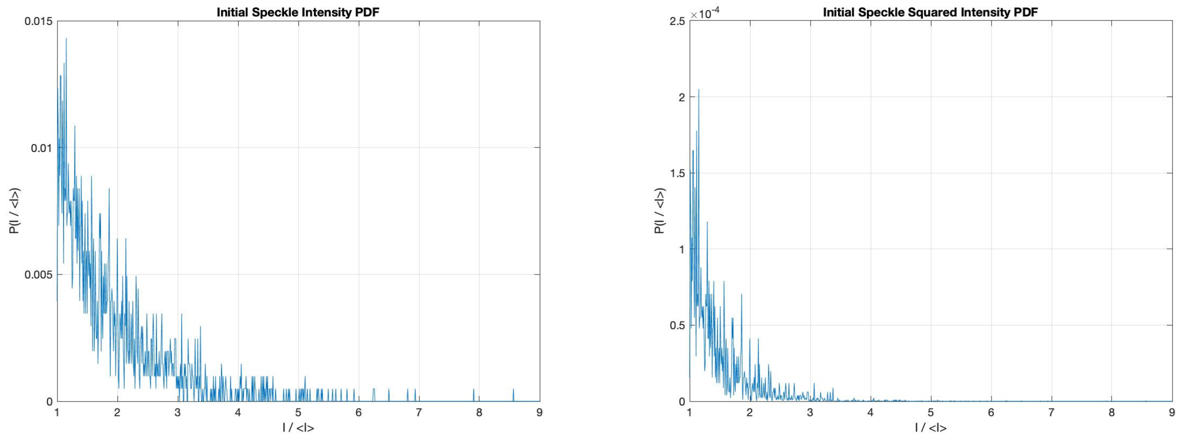


Figure 1.9: Intensity PDF's of the speckles above. It is again seen that most of middle intensity point is disappering when the statistics squared

Contrary to that the same application with the So to use the speckles in non-linear imaging they are needed to be tailored to the $e^{-0.5I}$ which is called sub-Rayleigh.

Chapter 2

Code

All the codes of the project are based on the paper "Customizing speckle intensity statistics" [1]. Full code is given in the appendix A.

Step-by-step explanation of the code is as follows:

1. Creating initial a speckle from random phases
   This step is done by transformation matrix which is noted from the SLM used in the real experiment of the paper.

2. Intensity of the speckle is calculated and transformed to the $e^{(-0.5I)}$ This step is done according to the transformation formula [1]

$$\int_0^I e^{-I'} dI' = \int_{\bar{I}_{min}}^{\bar{I}} F(\bar{I}') d\bar{I}' \tag{2.1}$$

   solving it for $\tilde{I}$.

3. Next is to retrieve a phase for created "target" intensity. For the non-linear optimization is used. Non-linear optimization package used in the code is NLopt.

4. Finally phase retrieved from the non-linear optimization applied to the SLM by a transformation matrix and speckle which obeys wanted statistics created.

The figure 2.1 is taken from the paper [1], there are some statistics targeted and results are seen.

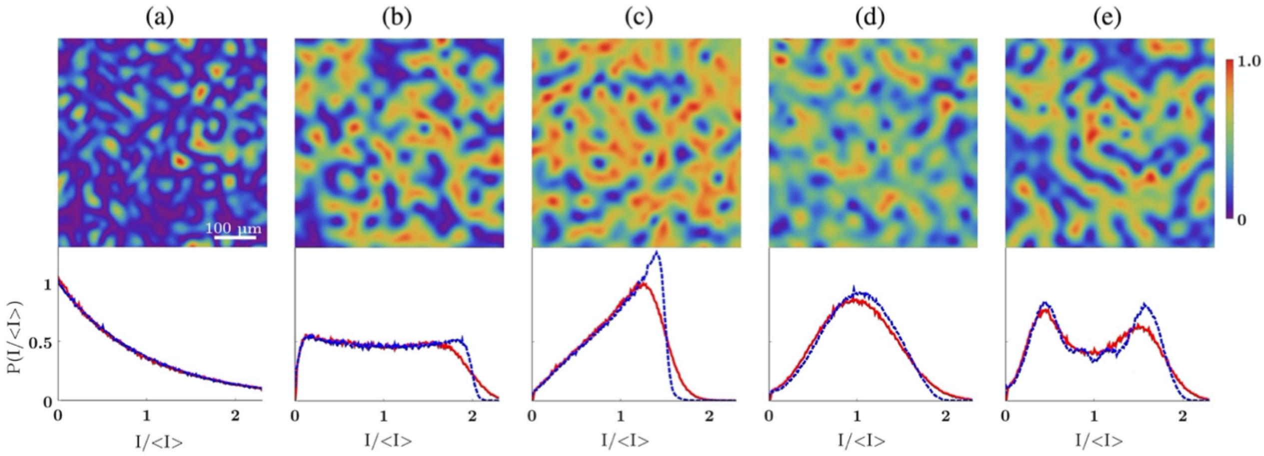


Figure 2.1: From left to right: Rayleigh, uniform, linear, unimodal and bimodal distributions. Blue line is numerical curves and the red ones are experimental result [1].

Chapter 3

Applications

3.1 Creating Target Intensity

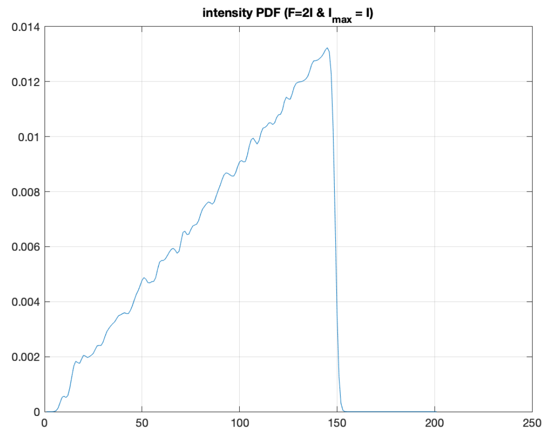
First step after creating the speckle shown in the 1.8, intensity of the speckle is calculated. This intensity is the left hand side of the equation 2.1 which is Rayleigh distribution with $\sigma = 1$. By checking

$$\int_{\bar{I}_{min}}^{\bar{I}_{max}} F(\bar{I}')d\bar{I}' = 1$$

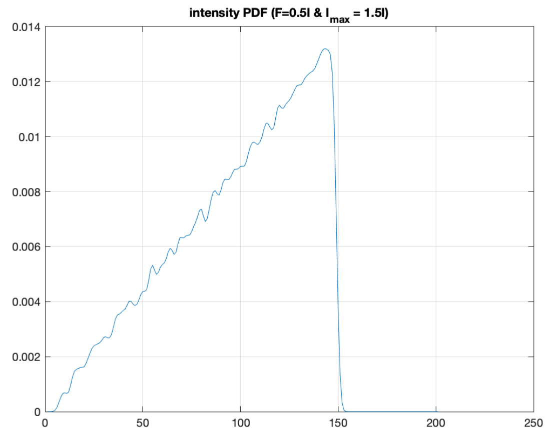
and

$$\int_{\bar{I}_{min}}^{\bar{I}_{max}} \bar{I}'F(\bar{I}')d\bar{I}' = \langle I \rangle = \langle \bar{I} \rangle$$

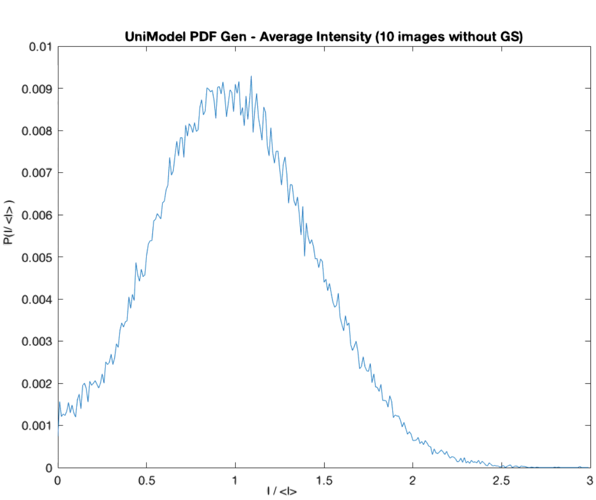
holds, different intensity PDFs are calculated: $F = 2I$, $F = 0.5I$, $F = 3I^2$, $F = 4I^4$, $F = exp(-0.5I)$.



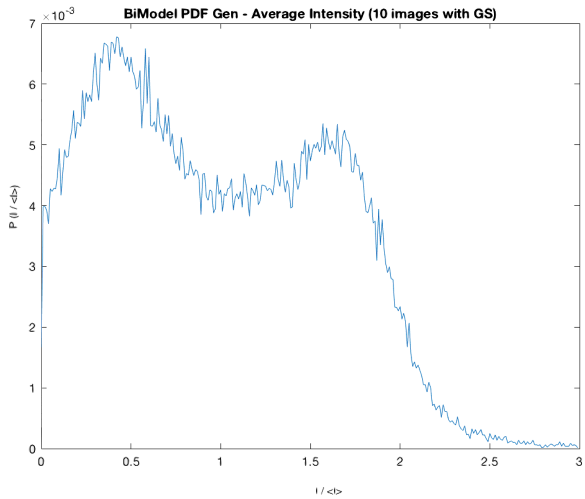
Target intensity is $2I$ with max value I



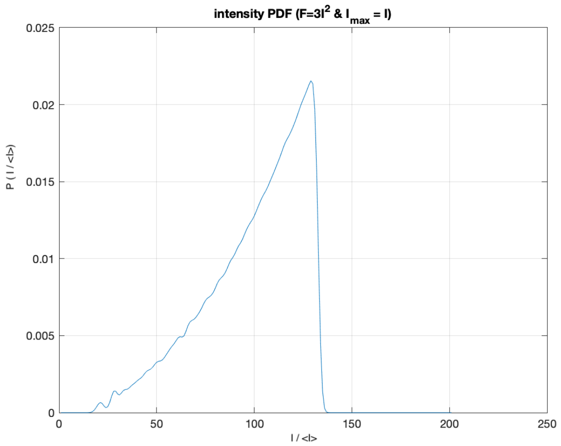
Target intensity is $0.5I$ with max value $1.5I$



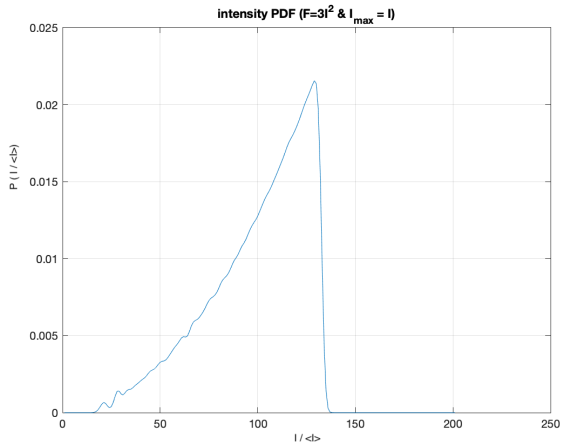
Unimodal Distribution



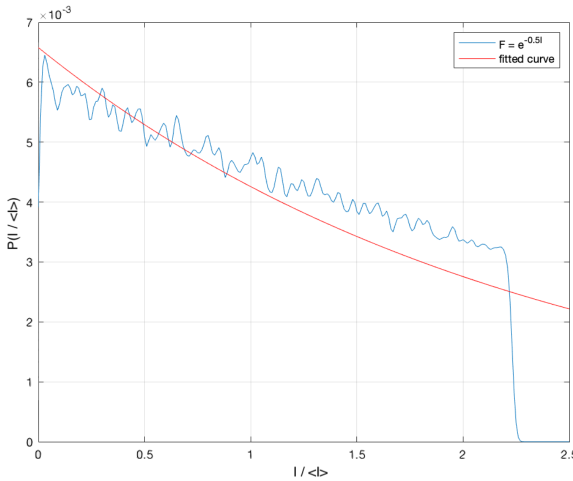
Bimodal Distribution



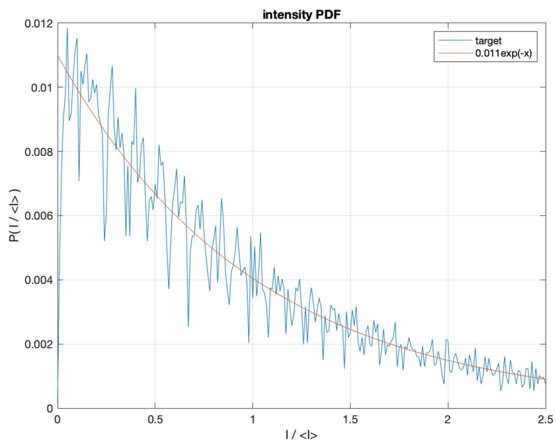
Quadratic distribution



Cubic Distribution



Sub-Rayleigh distribution



Rayleigh Distribution

Figure 3.1: Target intensity PDFs

So all of these different target intensity PDFs are generated from the Rayleigh intensity of the initial speckle.

A problem is encountered with the linear distribution that slope and the end points of the curved can not be changed. So for the two different targets $F = 2I$, $F = 0.5I$ same plot is obtained. In the figure 3.2 it is seen easily.

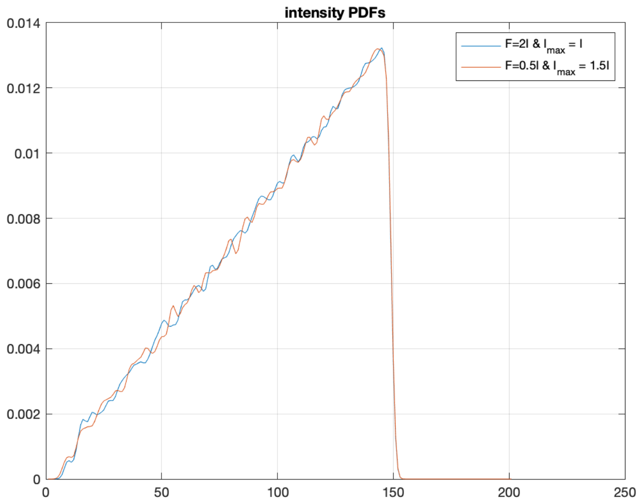


Figure 3.2: The problem of not able change the slope.

This problem is left a side in order to move on in the project. Because especially wanted statistics is the $e^{0.5I}$ which is also called sub-Rayleigh.

Contrary of the problem mentioned, the difference in the quadratic and cubic distributions is clearly seen in the figure 3.3

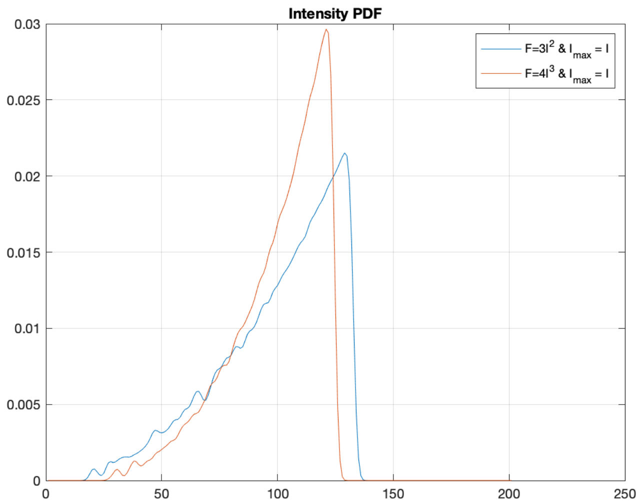
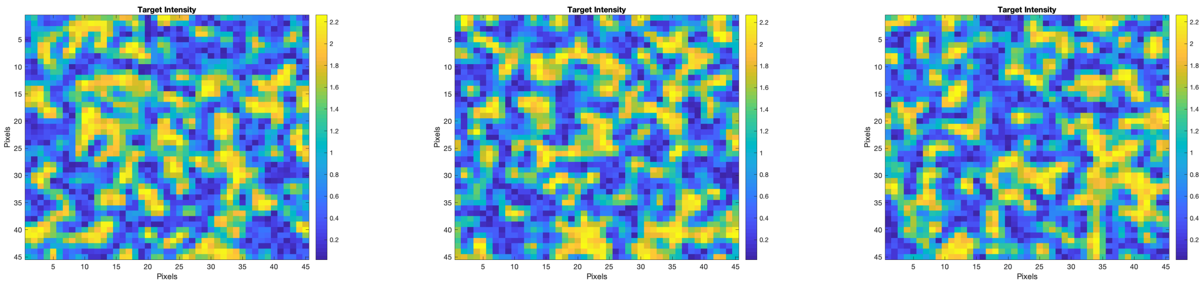


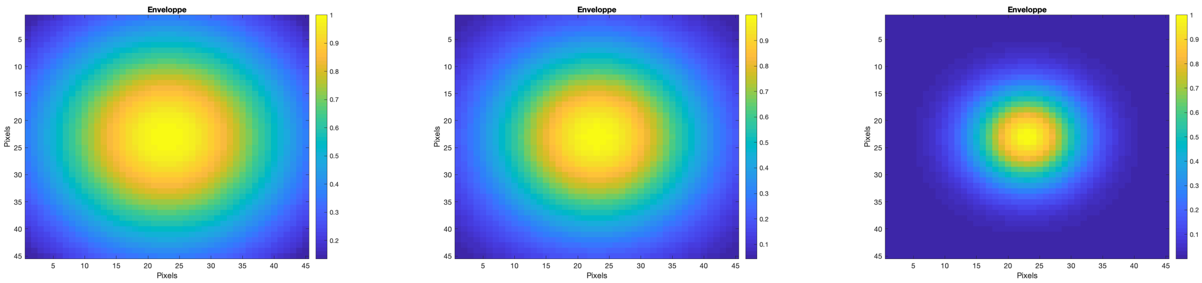
Figure 3.3: Difference between cubic and quadratic distribution.

3.2 Envelope

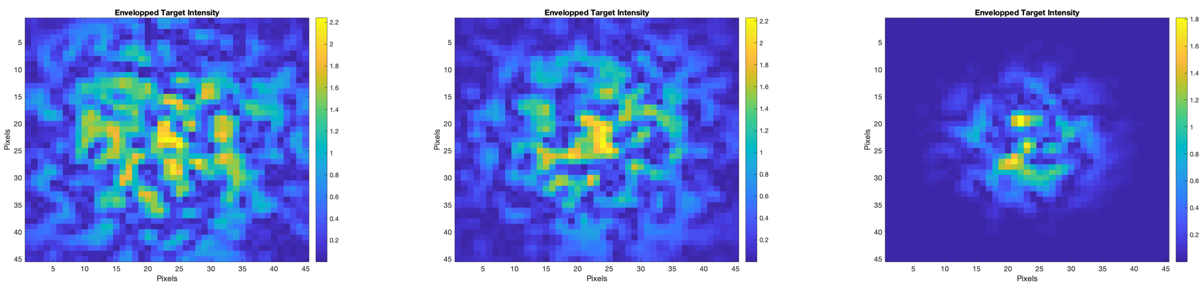
In order to obtain some shape closer to the one obtained in the experiment an envelope should be applied to the speckle. Since in the experimental setup it is impossible to use all the pixels of the SLM small portion of it is used and because of the nature of the light there is always a distribution close to the Gaussian. In numerical simulation a Gaussian distribution applied to the speckle and rest of the calculation is done with that envelope.



Target intensities of $e^{-0.5I}$
each of them created from
different initial speckle



Different Gaussian envelopes



Target intensities after
envelope applied

The rightmost speckles are most real looking ones.

3.3 Phase Retrieval and Target Speckle

Now non-linear optimization is used to get the phase from the intensities above. As it is found out in the figure 3.4, intensity is increasing as envelope is getting narrower. This is expected from the energy conservation it is expected to have higher intensities with small envelope.

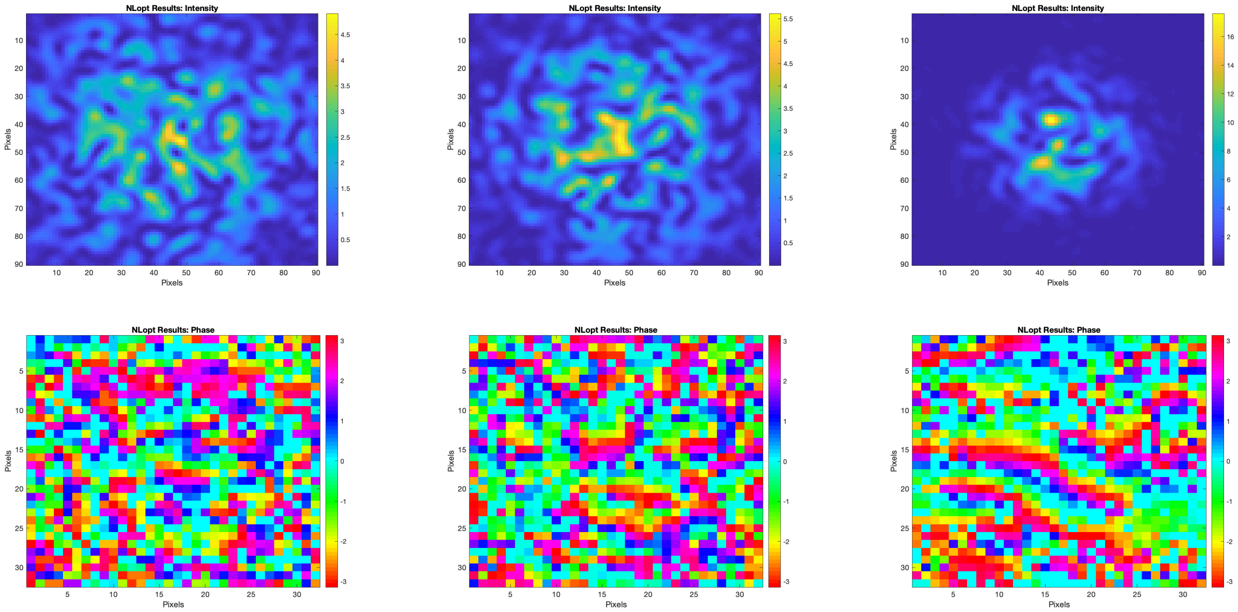


Figure 3.4: Phase Retrieval and Target Speckle

In the SLM phase, it can be obviously seen that in the smallest envelope there are some patterns. These are high frequency components arise because of the existence of the smaller envelope. Again to hold the energy conservation more higher frequency terms intervene. This high frequency terms are also seen in the auto-correlation functions.

3.4 Auto-correlation

Until this part all of the steps above is realised 50 times. To have a better understanding average of these 50 realization is taken. In the figure 3.5, both of the one realization and the average over 50 realizations are shown. There is slight difference between them. Noises seen in the tail of the auto correlation functions of one realization are seem to be disappeared in the averaged graph.

Full width at half maximum of auto-correlation function shows the average speckle grain size. To have a higher resolution this value is wanted to be as small as possible.

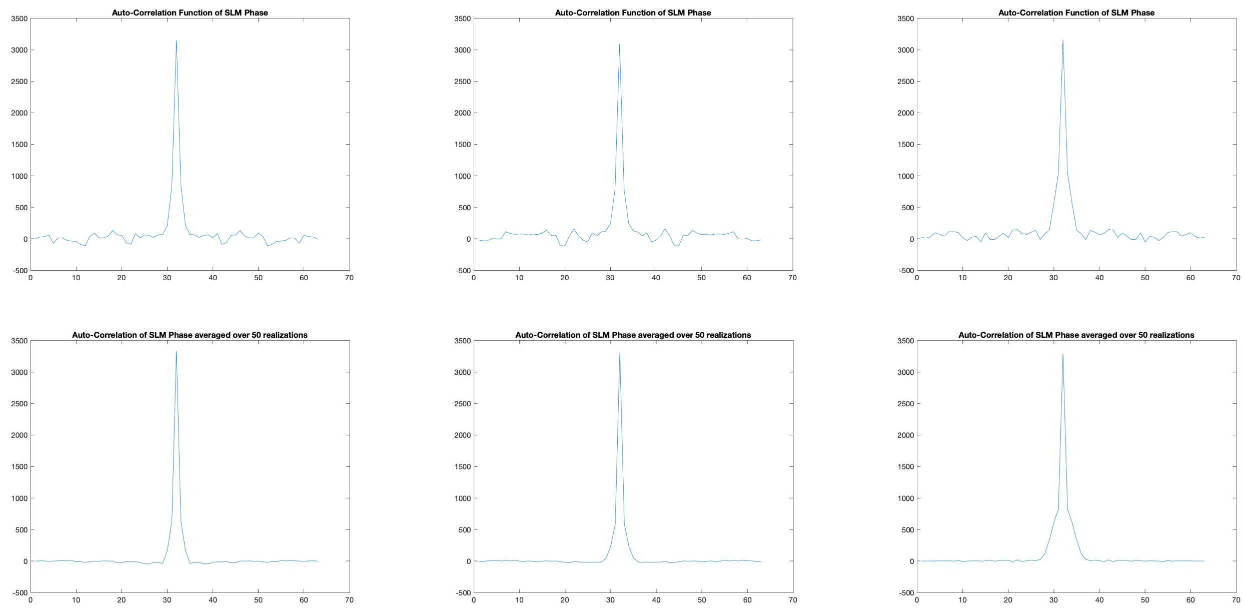


Figure 3.5: Auto-correlation functions.

Here it is seen that when envelope gets narrower auto-correlation function is getting larger. This is again because of the envelope. But to see whether it is still preferable or not it is needed to make a comparison with Rayleigh. This part has not done yet in the internship.

3.5 Comments & Further Work

Our goal was to have an other statistics than Rayleigh in order not to lose speckle by using the speckle patterns in non-linear imaging. Sub-Rayleigh distribution ($exp(-0.5I)$) is selected as a target statistics. So one way to see the non-linear effect on these distributions semi log plot can be plotted as figure 3.6.

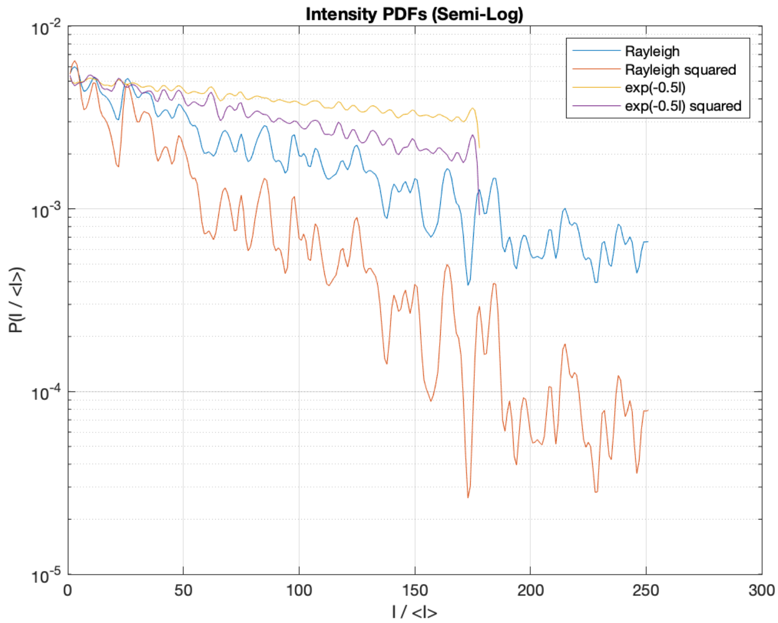


Figure 3.6: Comparison between Target intensity PDFs: Rayleigh, Rayleigh squared, sub-Rayleigh and sub-Rayleigh square distributions on a semi-log graph.

It can be seen that Rayleigh square is the one which has the steepest slope. The difference between Rayleigh square and $exp(-0.5I)$ squared is the reason why $exp(-0.5I)$ is a preferred distribution that the Rayleigh. In the figure 3.6, plots are target intensities, the result after the step 2 (figures 3.1) so they are not result of the whole algorithm.

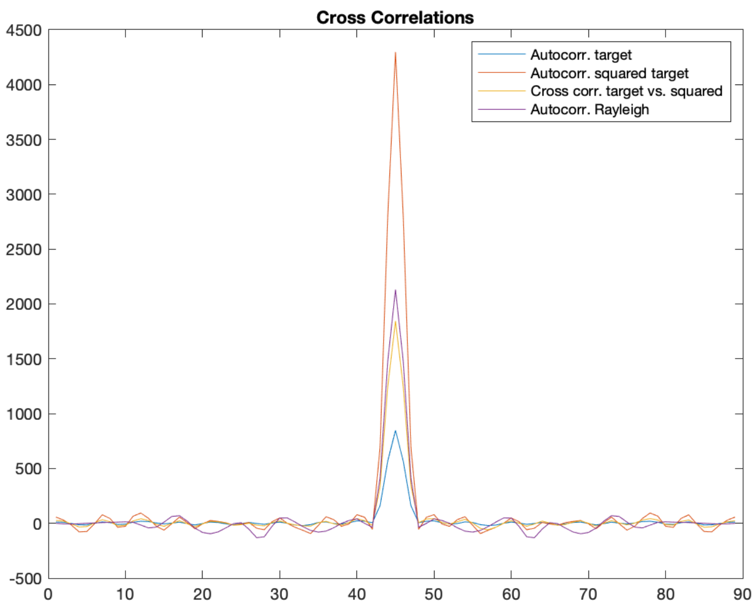


Figure 3.7: Comparison between correlation functions.

Similarly compared correlation functions in figure 3.7 are also giving information about the profit to use $e^{-0.5I}$ instead of Rayleigh. But again these are only comparison for the target intensities. To have the see the really effect comparison of the auto correlation should be applied the final speckle created in the figure 3.4. The reason why the auto-correlation comparison and the intensity PDFs comparison are not done by the speckles in the figure 3.4 is that the intensity PDFs calculated by these speckles are incorrect especially in the smallest envelope. The origin of the problem may be the way of calculating the intensity PDFs. As it seen in the appendix A.3, intensity PDFs are calculated by counting all the pixels in a specific intensity step by step from 0 to maximum. But when the intensity is increasing with the envelope is getting smaller, pixels have wider range to have a specific intensity. For example when the intensity goes from 0 to 3 pixels can get 300 numbers between 0 and 3.00 (steps by 0.01) but when intensity range is 0 to 16, there are 1600 numbers. So low intensity pixels are always around 0 but high intensity pixels are not always in the same spot over 50 realizations. That situation leads a huge peak at the 0 as it is shown in the figure 3.8, in which all the PDFs are completely unrealistic.

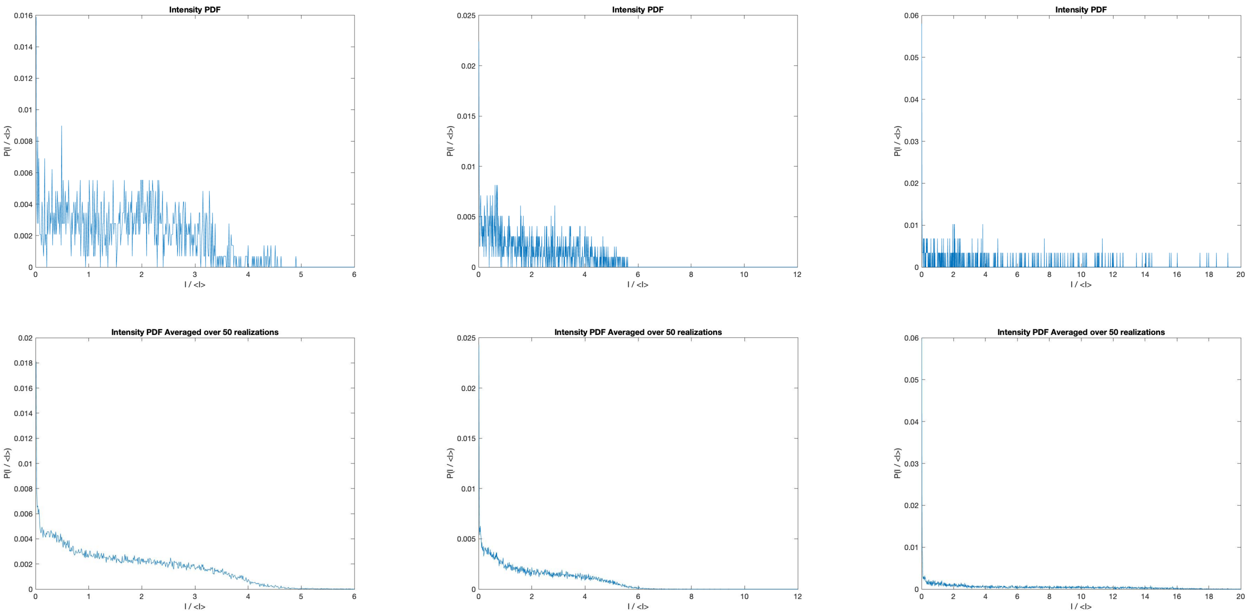


Figure 3.8: Intensity PDFs.

Because they are unrealistic comparison between the different statistics are not showing expected results, figure 3.9.

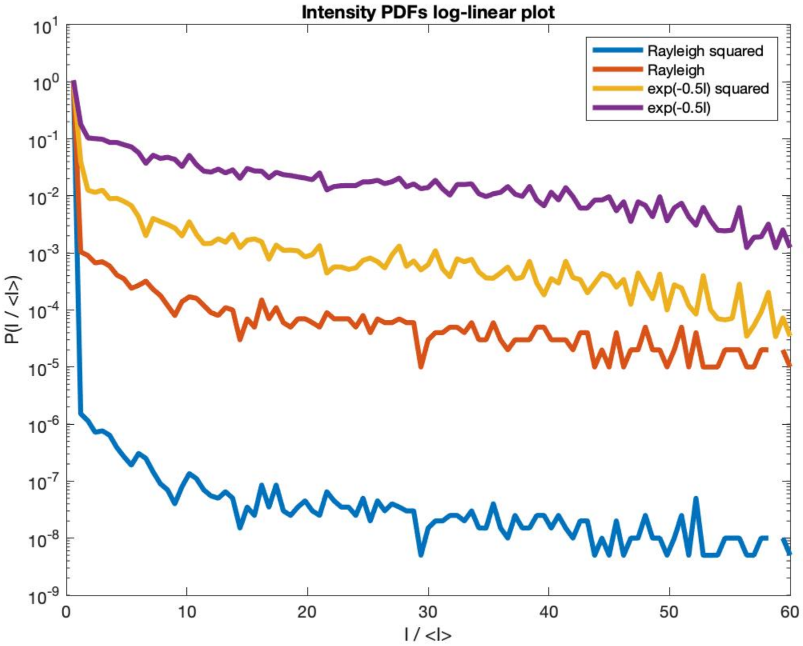


Figure 3.9: Comparison between resultant intensity PDFs.

That's why a new way to calculate intensity PDFs is planned, which is averaging the intensity of one specific pixel over number of realization. It is thought that to have an accurate statistic 2025 realizations should be done. Because there are $45 \times 45 = 2025$ pixels.

Because the system is thought to be ergodic the averaging it in the position direction or the time direction (here, realization direction) should be the same. This new idea has not been tried.

Bibliography

[1] Bender, N., Yilmaz, H., Bromberg, Y., and Cao, H. Customizing speckle intensity statistics. *Optica 5*, 5 (2018), 595–600.

[2] Dijkstra, R. R. Brightness characterization of single core and core/shell yb3+, er3+-doped nayf4 upconversion nanoparticles. Master's thesis, University of Twente, 2012.

[3] Goodman, J. W. *Speckle phenomena in optics: theory and applications.* Roberts and Company Publishers, 2007.

[4] Gustafsson, M. G. Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *Journal of microscopy 198*, 2 (2000), 82–87.

[5] Gustafsson, M. G., Agard, D., Sedat, J., et al. I5m: 3d widefield light microscopy with better than 100nm axial resolution. *Journal of microscopy 195*, 1 (1999), 10–16.

[6] Labouesse, S., Allain, M., Idier, J., Bourguignon, S., Negash, A., Liu, P., and Sentenac, A. Fluorescence blind structured illumination microscopy: a new reconstruction strategy. In *2016 IEEE International Conference on Image Processing (ICIP)* (2016), IEEE, pp. 3166–3170.

[7] Liu, P. *Statistical and numerical optimization for speckle blind structured illumination microscopy.* PhD thesis, 2018.

[8] Liu, X., and Velten, A. The role of wigner distribution function in non-line-of-sight imaging.

[9] Mudry, E., Belkebir, K., Girard, J., Savatier, J., Le Moal, E., Nicoletti, C., Allain, M., and Sentenac, A. Structured illumination microscopy using unknown speckle patterns. *Nature Photonics 6*, 5 (2012), 312–315.

[10] Saleh, B. E., and Teich, M. C. *Fundamentals of photonics.* john Wiley & sons, 2019.

[11] Vangindertael, J., Camacho, R., Sempels, W., Mizuno, H., Dedecker, P., and Janssen, K. An introduction to optical super-resolution microscopy for the adventurous biologist. *Methods and applications in fluorescence 6*, 2 (2018), 022003.

[12] Zernike, F. How i discovered phase contrast (nobel lecture), 1953.

Appendix A

The Matlab code

A.1 Main Code

```
rng('shuffle')

addpath('/usr/local/lib/matlab')
addpath('Functions')
sigma = 10; %for the Gaussian envelope
n=10;        %number of realization
m= 500;      %number to manipulate the max number of Intensity PDF
c = 1;       %changing the step of Intensity PDF calculation


Rayleigh = zeros(n,m/c);
IntensityPDF = zeros(n,m/c);
PhaseCorr = zeros(n,63);

Data=load('TMatrix10.mat');
matrix=Data.Matrix;            %transformation  matrix

for i=1:n
    fprintf('%i\n',i)
    clearvars -except i n x x_ray y matrix m c e;

   SLM.Length=32;%The total length of one side of the square SLM
       array
    targetL=90;%length of target region;

    TMatrix = refreshing_TMatrix(matrix,targetL,SLM);

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    %%%%%%%%%%%%%%Compressing TMatrix%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    TMatrix.dL=2;
    TMatrix.matrix=TMatrixCompress(TMatrix,TMatrix.matrix);
    TMatrix.ColumnLength=size(TMatrix.matrix,1);
    TMatrix.Length=abs(sqrt(TMatrix.ColumnLength));
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %creating initial speckle & transforming it to a target intensity
    target = target_gen(SLM,TMatrix);
    %With target intensity & initial speckle phase calculating (non-linear opt.)
    %SLM phase for wanted speckle & calculating wanted speckle
    number = numeric_solv(target,TMatrix,SLM,matrix,sigma,m,c);
    %converting Rayleigh distributions of initial speckles to intensityPDF
    rayleigh = getPDF(TMatrix,target.Int_ray,m,c,number.mask,number.env);

    %storing each realisation
    Rayleigh(i,:) = rayleigh.Plist;
    IntensityPDF(i,:) = number.Plist;
    PhaseCorr(i,:) = number.phase_corr;
end
%creating a file name
index=[sprintf('%i',sigma),sprintf('%i',c)];
shape = ''; %used it for different wanted statistics
%Ploting Intensity PDF, slm phase, speckle intensity, envlope, enveloped target, target, phase autocorr
ploting(number,index,shape);

%ploting average intensity PDF over n realizations
figure;
plot(number.Clist,averaging_data(IntensityPDF))
title(['Intensity PDF Averaged over ',sprintf('%d',n),' realizations']);
xlabel("I / <I>");
ylabel("P(I / <I>)");
grid;
```

```
saveas(gcf,[shape,'Averaged_PDF50_',index,'.jpg']);

%ploting average auto correlation PDF over n realizations
figure;
plot(averaging_data(PhaseCorr))
title(['Auto-Correlation of SLM Phase averaged over ',sprintf('%d',n
    ),' realizations']);
grid;
saveas(gcf,[shape,'Averaged_Auto50_',index,'.jpg']);

save(['exp-',shape,'_',index,'.mat'],"y","x","number","x_ray");


%Try to plot all statistics in semilof to see the steeps
plot(number.Clist,averaging_data(Rayleigh.^2),'linewidth',3)
hold on;
plot(number.Clist,averaging_data(Rayleigh),'linewidth',3)
plot(number.Clist,averaging_data(IntensityPDF.^2),'linewidth',3)
plot(number.Clist,averaging_data(IntensityPDF),'linewidth',3)
legend("Rayleigh squared","Rayleigh","exp(-0.5I) squared", "exp(-0.5
    I)")
set(gca, 'YScale', 'log');
title('Intensity PDFs log-linear plot')
xlabel("I / <I>");
ylabel("P(I / <I>)");
```

A.2 Creating Target Generating Part

```
function [Out] = target_gen(SLM, TMatrix)
    targetL=90;
    AveVal=2.5; %Here I want a average intensity of 2.5
    %what you have in a Rayleigh speckle pattern
    clearvars slmV DCor slm obsTemp obs Int dum RCor i Int_ray;
    norm=0;
    numb=1000;
    for i=1:numb
        slmV=exp(complex(0,2*pi*rand(SLM.Length*SLM.Length,1)));
        obsTemp=OBSVectorToMatrix(TMatrix,TMatrix.matrix*slmV);
        norm=norm+sum(sum(abs(obsTemp).*abs(obsTemp)))/numb;
    end
    norm=norm*AveVal; %So this is the norm I will normallize
    %all of my speckles to have
```

```
    %%%%set the speckle intensity correlation function%%%%%
    %%to prevent our target speckle patterns to prevent our%%
    %%%%%%%%%%%%%%target speckle patterns%%%%%%%%%%%%%%%%
    RCor=zeros(targetL/2);
    for i=1:numb
        RCor=RCor+abs(FT(abs(OBSVectorToMatrix(TMatrix,TMatrix.
            matrix*exp(complex(0,2*pi*rand(SLM.Length*SLM.Length,1)))
            )).^2))/numb;
    end
    RCor=RCor/max(max(RCor));
    DCor=RCor;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %This is your intial phase
    slm=exp(complex(0,2*pi*rand(SLM.Length*SLM.Length,1)));
    obs=OBSVectorToMatrix(TMatrix,TMatrix.matrix*slm);
    Int=abs(obs).^2;%This is you initial speckle pattern
    Int_ray=Int/mean2(Int);
    %Now we are going to generate the target with a G.S. algorithm
    for i=1:100 %this does not need to be so large
         Int=abs(IFT(exp(complex(0,angle(FT(Int)))).*DCor));
         Int=XPDFGen(Int);
    end
    %now we just need to renormalize everything and convert it to a
       vector


    Out.norm = norm;
    Out.slm = slm;
    Out.Int_ray = Int_ray; %initial Rayleigh
    Out.target = Int/mean2(Int);
end
```

A.3 Creating Wanted Speckle Part

```
function [Out] = numeric_solv(target,TMatrix,SLM,matrix,sigma,m,c)
    clear Clist Plist phase intensity env env_target phase_corr
       target2;
    targetL=90;
```

```
    TMatrix.Length = 45;
    TMatrix.ColumnLength = 2025;
    env = Gauss_2D(sigma,45); % envelope
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Mask is created to be used in the intensity PDFs calcul%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    mask = zeros(45,45);
    norml = 0;
    for i=1:length(mask)
        for j=1:length(mask)
            if (j-22.5)^2+(i-22.5)^2<=sigma^2
                mask(i,j)=1;
                norml = norml + 1;
            end
        end
    end
    mask = imresize(mask,2);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %weighting
    env_target = target.target.*env;
    target2=OBSMatrixToVector(TMatrix,env_target);
    target2=target2*(target.norm)/sum(target2);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%Ok now to the nlopt section of the code%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    vars.target=target2;
    vars.M=TMatrix.ColumnLength;%the number of camera d.o.f
    vars.N=SLM.Length*SLM.Length;%the number of SLM degrees
    phase=angle(target.slm)+pi; %to make it from 0 to 2pi
    vars.matrix=TMatrix.matrix;
    vars.Dmatrix=complex(0,eye(vars.N));


    opt.algorithm = NLOPT_LD_LBFGS;
    opt.lower_bounds = zeros(vars.N,1);
    opt.upper_bounds = 2*pi*ones(vars.N,1);
    opt.min_objective = @(x) myfunc(x,vars);
    opt.xtol_rel = 0.0001;
```

```
opt.maxeval=inf;%for the global
[xopt, fmin, retcode] = nlopt_optimize(opt, phase);

slm2=exp(complex(0,xopt)); %retrieved phase

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%previous one is compressed.
TMatrix2 = refreshing_TMatrix(matrix,targetL,SLM);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%Creating new speckle%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
speckle=TMatrix2.matrix*slm2;
intensity = abs(speckle).^2;
speckle=OBSVectorToMatrix(TMatrix2,speckle);
speckle=speckle./(mean(mean(intensity))).^(1/2);
%speckle is complex valued. dividing to RMS value??
speckle_phase =  angle(speckle);
phase=angle(slm2);
phase=reshape(phase,[32,32]);
intensity=OBSVectorToMatrix(TMatrix2,intensity);
intensity=intensity./mean2(intensity);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%Calculating the intensity PDFs%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
temp2=OBSMatrixToVector(TMatrix2,intensity.*mask);
temp2 = temp2/mean(temp2);
temp2=round(temp2,2);
Plist=zeros(1,m/c);
Clist=zeros(1,m/c);
a = 0;
for i=1:m
```

```
            a = a + sum(temp2==round((0.01)*(i),2));
            if mod(i,c)==0
                Plist(i/c)=a;
                Clist(i/c)=(0.01)*(i);
                a = 0;
            end
        % Plist(i) = sum(temp2==round((0.01)*(i),2));
        % Clist(i)=(0.01)*(i);
        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


        Out.Plist=Plist/(sum(Plist));
        Out.Clist = Clist;
        phase_corr = xcorr2(phase-mean2(phase),phase-mean2(phase));
        Out.phase_corr = phase_corr(round(length(phase_corr)/2),:);

        Out.intensity = intensity;
        Out.phase = phase;
        Out.env=env;
        Out.target = target.target;
        Out.env_target=env_target;
        Out.mask  = mask;
        Out.speckle_phase = speckle_phase;
end
```

A.4 Performing Target Intensity Transformation

```
function [out] = XPDFGen(IntArray)
        normO=sum(sum(IntArray));
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%This part calculates the pdf
            %%%%%%%%%%%%%%
        IntArray=IntArray.*(length(IntArray(:,1,1)).^2)./normO;
        %intensity of the initial speckle divided by the average
            intensity
        temp=reshape(round(IntArray,2),length(IntArray)^2,1);
        %make it vector and get rid of the 2nd digit after the point
        Plist=zeros(1001,1);
        for i=0:1000
            Plist(i+1)=sum(temp==round((i+1)/100,2));
```

```
    end
    %creating the PDF by summing how many number in that specific
        intensity
    con=exp(-([-4:0.5:4]).^2);
    Plist=conv(Plist,con,'same');
    %make it smoother
    Plist=Plist/sum(Plist);
    %normalize them
    IPlist=zeros(1001,1);
    for i=1:1001
        IPlist(i)=sum(Plist(1:i));
    end
    %%%%%%%%%%%%%%%%%%%%%%%%Now We transform the PDF to what we want
        %%%%%%%%%%
    IPlist=-2*log(1-0.5*IPlist);
    %%%%%%%%%%%%%%%%%%%%%%%%Now lets perform the Transform
        %%%%%%%%%%%%%%%%%%%%%
    out=round(IntArray,2);
    for i=1:1001
        I=round((i-1)/100,2);
        out(out==I)=IPlist(i);
        %specific intensities in the initial intensity matrix is
            replaced
        %by the corresponding transformed value
    end
    out=out.*(normO./sum(sum(out)));
end
```

A.5 Cost Function of Non-Linear Optimization

```
%%%This code will define the cost function we wish to%%%%
%%%%optimize. We can use a vector input in addition to%%%
%%%%%%%a matrix multiplication as an operation%%%%%%%%%%%
%%%%%%%However, we will need to convert this into a%%%%%%
%%%%%%scalar which is being optimized in the end%%%%%%%%%
function [val, gradient] = myfunc(phase,vars)
%vars is the object which contains all of the information
%you will need. Additionally the vector "phase" gives the
%value of the phase on the SLM at given points according%
%to the Tmatrix basis. Thse are the proverbial knobs we%%
%are going to tweak to get the desired target function%%

```

```
%start by defining the SLM field vector.
    slm=transpose(exp(complex(0,phase)));
    obsField = vars.matrix*slm;
    obsIntensity=abs(obsField).*abs(obsField);
    dVector=vars.target-obsIntensity;
    val=sum(abs(dVector).*abs(dVector));
    if (nargout > 1)

        slm=exp(complex(0,phase));
%       slm=transpose(exp(complex(0,phase)));
        Dmatrix=-4*repmat(dVector,1,vars.N).*real(vars.matrix.*(
            complex(0,1)*repmat(slm,vars.M,1)).*repmat(conj(obsField)
            ,1,vars.N));
        gradient=transpose(sum(Dmatrix));
    end
end
```