
GUIDE - Speckle Analysis in *DCCLab*

Gabriel Genest (gabriel.genest.2@ulaval.ca)

Speckles are optics phenomena often undesired when we do microscopy or we work with lasers in general. However, some scientists want to use speckles, because they can be useful when acquiring high contrast and precision images, like HiLo microscopy. Knowing statistics about speckles is quite complex in theory, because the mathematics because difficult with complicated integrals, but numerical methods exist to help us in this task. This document will go over the details of what are speckles, how can we statistically describe them and how can we analyze them numerically from an image.

Contents

1 Introduction	3
1.1 Why speckle analysis?	3
2 Speckle statistics	4
2.1 Average size of speckles	6
2.1.1 Free space geometry	6
2.1.2 Imaging geometry	10
3 Computation of speckles' average size	11
3.1 Morphology	11
3.1.1 Algorithms well defined	11
3.1.2 Can count speckles	11
3.1.3 Can generate a distribution of sizes	12
3.2 Autocorrelation	12
3.2.1 Fast algorithm	12
3.2.2 No parameters required	13
4 Python implementation of speckle size analysis	14
4.1 Reading the image	14
4.2 Computing autocorrelation	14
4.2.1 Preprocessing	14
4.3 Finding the FWHM of the autocorrelation array	15
4.3.1 The mean method	15
4.3.2 The linear fit method	16
4.4 Giving information to the user	17
4.4.1 Individual pieces of information	17
4.4.2 All the info at the same time	17
4.5 Example of the code usage	17
4.6 GUI implementation: what to know and how to use	19
5 Conclusion	21
Bibliography	22
A The Delta function	23
B Some moderately advanced notions of statistics	23
B.I Central Limit Theorem	23
B.II Change of variable in statistics	24
B.III Sum of probability distributions	25
C Random phasors	25
C.I What is a phasor	25
C.II Sum of random phasors	25
C.III Joint probability distribution of the real and imaginary parts of the phasors sum	26
C.IV Joint probability distribution of the amplitude and the phase of the phasors sum	26
C.V Probability density of the speckle pattern intensity	27
D Sum of speckle patterns	28

1 Introduction

This document aims to explain the purpose of speckle analysis and how it is achieved. This guide also explains the code created and how to implement or use it. It is considered that the reader knows the basis of *Python* programming ¹ and is at ease with basic notions of statistics and calculus².

1.1 Why speckle analysis?

The concept of speckles ³ is really important in *HiLo* microscopy ⁴ and it is mandatory to know some information about them, such as the average size and the contrast.

There are not a lot of already existing modules or software able to compute those information. This is why the *DCCLab* wrote *Python* code able to extract important speckles features. The code developed aims to be easy to use for scientists, even if they don't consider themselves good with *Python* or programming in general. The code hierarchy and construction also aims to be easy to understand, hence easy to implement in other projects or easy to work with if one wants to add new features.

¹Resources will be presented in case one does not feel at ease with *Python*

²Annexes will talk more in depth about those subjects in case the reader wants a little reminder.

³See Section 2 for more info

⁴See <https://github.com/DCC-Lab/Documentation/blob/master/GUIDE/GUIDE-MicroscopeHiLoZebrafish.pdf> for more info about *HiLo*, in French only

2 Speckle statistics

Laser speckle is a phenomenon quite recent. It is known and documented since the 60's. There are mainly two groups of scientists: those who want speckles and those who want to remove them. As you may think, the *DCCLab*, at least the *HiLo* project, is part of the first group. Speckles arise when a bunch of random phasors, wavefronts or photons interfere with each other and create a pattern of bright and dark spots. Figure 1 shows an image of laser speckles.

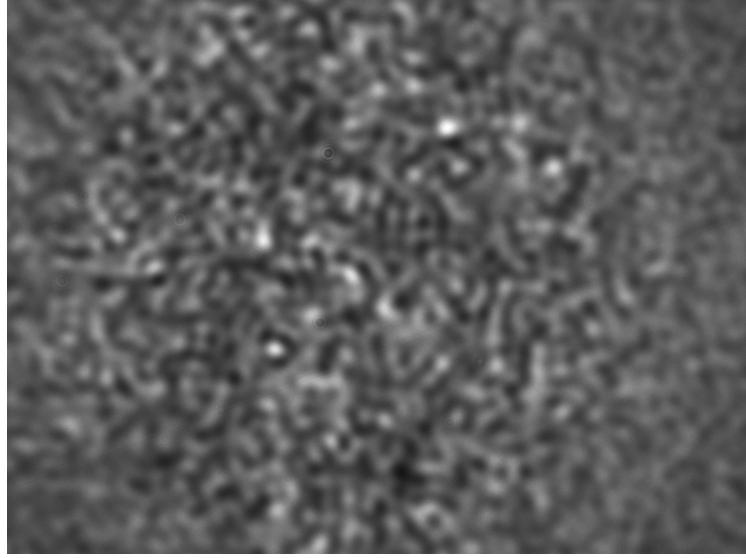


Figure 1: Example of speckle pattern. Copyright *DCCLab*.

In more technical terms, laser speckles can be seen as a random walk of phasors in the complex plane[1]. Figure 2 shows two examples of random walks where one results in constructive interference (figure 2a) and one results in destructive interference (figure 2b).

We will now focus on the probability distribution of the speckle amplitude, phase and intensity. In terms of amplitude A and phase θ , the joint probability distribution is:

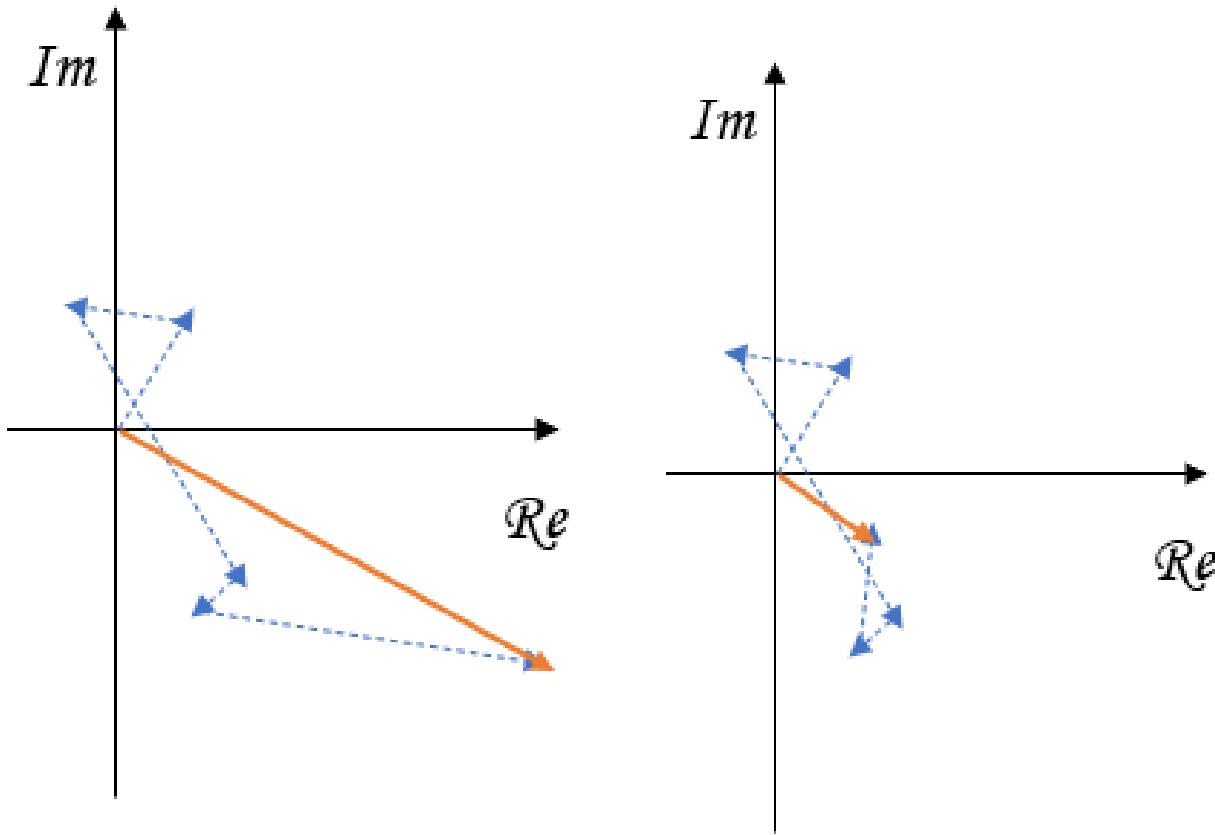
$$p_{A,\theta}(A,\theta) = \frac{A}{2\pi\sigma^2} \exp\left\{-\frac{A^2}{2\sigma^2}\right\} \quad (1)$$

where σ represent the variance of the distribution. A more complete development is presented in [1] and in the following annexes. The probability distribution presented in equation 1 is the joint probability of A and θ , but we can find the probability distribution of A or θ alone. We have:

$$p_A(A) = \int_{-\pi}^{\pi} p_{A,\theta}(A,\theta) d\theta = \frac{A}{\sigma^2} \exp\left\{-\frac{A^2}{2\sigma^2}\right\} \quad (2)$$

$$p_\theta(\theta) = \int_0^\infty p_{A,\theta}(A,\theta) dA = \frac{1}{2\pi} \quad (3)$$

The distribution of amplitudes, $p_A(A)$, is also known as a Rayleigh distribution. This is not a random fact! Rayleigh distribution appears when we work with complex numbers; in fact, when the complex numbers are independently and identically distributed Gaussian variable (i.e. they follow a Gaussian probability distribution, also known as a normal distribution), and when they have a zero mean and equal variance, the absolute value, here A , follows a Rayleigh distribution[2]. It is also worth noting that the distribution of phases, $p_\theta(\theta)$, is also known as a uniform



(a) Example of a random walk of phasors where the final intensity is large. The orange arrow represent the sum of the different phasors with different amplitudes and phases. The result is mostly a constructive interference because of the large final phasor.

(b) Example of a random walk of phasors where the final intensity is small. The orange arrow represent the sum of the different phasors with different amplitudes and phases. The result is mostly a destructive interference because of the small final phasor.

Figure 2: Examples of random walks in the complex plane with a small large and a small final vector. Based on [1].

distribution. This means that each θ in the range $-\pi \leq \theta \leq \pi$ has a probability of $\frac{1}{2\pi}$. With that, we can see that speckles are made of a sum of amplitudes following the Rayleigh distribution and with phases of uniform probability, we say that they create a fully developed speckle pattern! Other authors explore the mathematics when the phase is not uniform, also known as partially developed speckle patterns, or the amplitude does not follow a Rayleigh distribution, but such cases will not be discussed here. If the reader is interested in other interesting, but with harder math, [1] is a great resource.

It can also be interesting to consider the intensity instead of the amplitude and the phase, because this is actually what we measure in the lab when we work with laser speckles. We don't measure the electric field, its amplitude and phase, but the intensity. With some relatively basic statistical manipulations, presented in the annexes, one can find that the probability distribution

of the intensity of the speckle pattern follows an exponential law:

$$p_I(I) = \frac{1}{\bar{I}} \exp \left\{ -\frac{I}{\bar{I}} \right\} \quad (4)$$

where \bar{I} is the average intensity. Equation 4 applies only in the case where there is a large number of random phasors.

In some cases, we can acquire a sum of speckle patterns. Such cases arise when we acquire for too long, when the frame rate is smaller than the scale of Brownian motion[3]. Speckles pattern can also appear when a fiber is present[1]. We can derive the supposed intensity distribution in such cases:

$$p_{I_{tot}}(I_{tot}) = \begin{cases} \sum_{n=1}^N \frac{\bar{I}_n^{N-2}}{\prod_{p=1, p \neq n}^N (\bar{I}_n - \bar{I}_p)} \exp \left\{ -\frac{I_{tot}}{\bar{I}_n} \right\}, & \bar{I}_m \neq \bar{I}_n \forall m, n \in \{1, 2, \dots, N\} \\ \frac{I_{tot}^{N-1}}{\Gamma(N) I_0^N} \exp \left\{ -\frac{I_{tot}}{I_0} \right\}, & \bar{I}_m = \bar{I}_n = I_0 \forall m, n \in \{1, 2, \dots, N\} \end{cases} \quad (5)$$

where $I_{tot} = \sum_{n=1}^N I_n$. The case where the mean of every pattern is equal, the second case of equation 5, correspond to the Gamma distribution. This makes sense, because the sum of exponential distribution with same mean is a Gamma distribution. In both cases, when the number of patterns, N , is large, the central limit theorem is considered and we can approximate both cases to a Gaussian distribution, which simplifies computation since the first case is quite ugly and complicated.

2.1 Average size of speckles

The average size of speckles is not something very easy to understand. Most of the documentation read mentions only *what* should be used, but not *why* or *how* it should be used. We have to dig quite deep! Before going into the maths, let's talk about some propagation geometries.

2.1.1 Free space geometry

The free space geometry considers the case of a coherent light (e.g. a laser!) illuminating a planar rough surface. Said surface is at a distance z from the observer (see figure ??) and the wavelength of the light is λ . We assume that the rough surface is stationary in time (we don't consider any Brownian movement) and we also assume a paraxial propagation (only small scattering angles are considered). We can then write the complex amplitude in the observation plane $\mathbf{A}(x, y)$ in terms of the complex amplitude in the plane of the scattering surface $\mathbf{A}(\alpha, \beta)$ with the help of the Fresnel diffraction integral[1]:

$$\mathbf{A}(x, y) = \frac{\exp\{j k z\}}{j \lambda z} \exp \left\{ j \frac{k}{2z} (x^2 + y^2) \right\} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{A}(\alpha, \beta) \exp \left\{ j \frac{k}{2z} (\alpha^2 + \beta^2) \right\} \exp \left\{ -j \frac{k}{\lambda z} (x\alpha + y\beta) \right\} d\alpha d\beta \quad (6)$$

This equation is quite complicated to solve. We are now interested in the autocorrelation⁵, Γ_A , of the speckle field at the observer plane at two points, (x_1, y_1) and (x_2, y_2) . The autocorrelation function is defined by[1]:

$$\Gamma_A(x_1, y_1; x_2, y_2) = \overline{A(x_1, y_1) A^*(x_2, y_2)} \quad (7)$$

⁵For now, the reader should ask why we are interested in this quantity. Spoiler alert, we will be able to measure the mean speckle size with the autocorrelation!

Then, using equation 6 with equation 7, we have this (awful) equation[1]:

$$\begin{aligned} \Gamma_A(x_1, y_1; x_2, y_2) &= \frac{1}{(\lambda z)^2} \exp \left\{ j \frac{k}{2z} (x_1^2 + y_1^2 - x_2^2 - y_2^2) \right\} \\ &\times \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[\Gamma_A(\alpha_1, \beta_1; \alpha_2, \beta_2) \exp \left\{ j \frac{k}{2z} (\alpha_1^2 + \beta_1^2 - \alpha_2^2 - \beta_2^2) \right\} \right. \\ &\left. \times \exp \left\{ -j \frac{2\pi}{\lambda z} (\alpha_1 \alpha_2 + \beta_1 \beta_2 - x_1 \alpha_2 - y_1 \beta_2) \right\} d\alpha_1 d\beta_1 d\alpha_2 d\beta_2 \right] \end{aligned} \quad (8)$$

where $\Gamma_A(\alpha_1, \beta_1; \alpha_2, \beta_2)$ is found with equation 7, but in terms of $\alpha_1, \beta_1, \alpha_2$ and β_2 . We can make the assumption that the correlation extent is small, in other words the correlation decreases fast, and it can be approximated by a delta function (see annexes for more info about the delta function). We can then write[1]:

$$\Gamma_A(\alpha_1, \beta_1; \alpha_2, \beta_2) = \kappa I(\alpha_1, \beta_1) \delta(\alpha_1 - \alpha_2, \beta_1 - \beta_2) \quad (9)$$

where κ is a constant with units of $[L]^2$ and $I(\alpha_1, \beta_1)$ is the intensity at (α_1, β_1) . We can use this definition to simplify equation 8[1]:

$$\Gamma_A(x_1, y_1; x_2, y_2) = \frac{\kappa}{(\lambda z)^2} \exp \left\{ j \frac{k}{2z} (x_1^2 + y_1^2 - x_2^2 - y_2^2) \right\} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\alpha, \beta) \exp \left\{ -j \frac{2\pi}{\lambda z} (\alpha x_1 + \beta y_1 - \alpha x_2 - \beta y_2) \right\} d\alpha d\beta \quad (10)$$

where we used the new definition of equation 9 and inserted it in equation 8. We then used the property of the delta function where we can change the variable (see annexes). We also changed (α_1, β_1) to (α, β) . We can also propose another simplification: we are (almost) only interested in the modulus of the correlation function, hence we can *remove* the leading exponential in equation 10. We then have the following simplified equation[1]:

$$\Gamma_A(\Delta x, \Delta y) = \frac{\kappa}{(\lambda z)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\alpha, \beta) \exp \left\{ -j \frac{2\pi}{\lambda z} (\alpha \Delta x + \beta \Delta y) \right\} d\alpha d\beta \quad (11)$$

where $\Delta x = x_1 - x_2$ and $\Delta y = y_1 - y_2$. We can see that equation 11 is similar to a Fourier transform aside from the constants in front of the integrals.

Now that the correlation function is found, we can compute the correlation coefficient:

$$\mu_A(\Delta x, \Delta y) = \frac{\Gamma_A(\Delta x, \Delta y)}{\Gamma_A(0,0)}$$

With the correlation function found earlier, we can write⁶:

$$\mu_A(\Delta x, \Delta y) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\alpha, \beta) \exp \left\{ -j \frac{2\pi}{\lambda z} (\alpha \Delta x + \beta \Delta y) \right\} d\alpha d\beta}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\alpha, \beta) d\alpha d\beta} \quad (12)$$

Then, we can find an expression for the autocorrelation function of the intensity⁷, Γ_I , which is different from Γ_A :

$$\Gamma_I = \bar{I}^2 [1 + |\mu_A(\Delta x, \Delta y)|^2] = \bar{I}^2 \left[1 + \left| \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\alpha, \beta) \exp \left\{ -j \frac{2\pi}{\lambda z} (\alpha \Delta x + \beta \Delta y) \right\} d\alpha d\beta}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\alpha, \beta) d\alpha d\beta} \right|^2 \right] \quad (13)$$

⁶The mathematical derivation to achieve equation 12 can be found in [1].

⁷*Idem*

The reader may wonder why we are seeing these equations. The answer is that we will see how it is related to the mean speckle size, but first we will take a look at two different geometries where the correlation is quite nice.

First of all, let's consider a square $L \times L$ of uniform distribution intensity. The equation for the intensity is:

$$I(\alpha, \beta) = I_0 \text{rect}\left(\frac{\alpha}{L}\right) \text{rect}\left(\frac{\beta}{L}\right)$$

where $\text{rect}(x)$ represent the rectangular function defined by:

$$\text{rect}(x) = \begin{cases} 1, & |x| \leq \frac{1}{2} \\ 0, & \text{elsewhere} \end{cases}$$

It can be found that the Fourier transform of the rectangular function is the cardinal sine:

$$\text{sinc}(x) \equiv \frac{\sin(\pi x)}{\pi x}$$

Thus, we can write:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_0 \text{rect}\left(\frac{\alpha}{L}\right) \text{rect}\left(\frac{\beta}{L}\right) \exp\left\{-j\frac{2\pi}{\lambda z}(\alpha\Delta x + \beta\Delta y)\right\} d\alpha d\beta = L^2 I_0 \text{sinc}\left(\frac{L\Delta x}{\lambda z}\right) \text{sinc}\left(\frac{L\Delta y}{\lambda z}\right)$$

It is quite easy to find the autocorrelation function:

$$\Gamma_l(\Delta x, \Delta y) = \bar{I}^2 \left[1 + \text{sinc}^2\left(\frac{L\Delta x}{\lambda z}\right) \text{sinc}^2\left(\frac{L\Delta y}{\lambda z}\right) \right] \quad (14)$$

Another free space geometry interesting to study is for a circular scattering spot with uniform intensity. The diameter of the spot is D and the intensity is:

$$I(\alpha, \beta) = I(\Omega) = \text{circ}\left(\frac{2\Omega}{D}\right)$$

where $\Omega = \sqrt{\alpha^2 + \beta^2}$. The circular function $\text{circ}(x)$ is simply:

$$\text{circ}(x) = \begin{cases} 1, & x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

Before computing the autocorrelation function of the speckle intensity, we must find the Fourier transform of the circular spot:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{circ}\left(\frac{2\Omega}{D}\right) \exp\left\{-j\frac{2\pi}{\lambda z}(\alpha\Delta x + \beta\Delta y)d\alpha d\beta\right\} = 2\pi \int_0^{\frac{D}{2}} \Omega J_0\left(\frac{2\pi r\Omega}{\lambda z}\right) d\Omega = 2A \left[\frac{J_1\left(\frac{\pi D r}{\lambda z}\right)}{\frac{\pi D r}{\lambda z}} \right]$$

where A is the area of the scattering spot given by $A = \pi(D/2)^2$, J_0 is the first kind Bessel function of order 0, J_1 is the first kind Bessel function of order 1 and $r = \sqrt{(\Delta x)^2 + (\Delta y)^2}$. It is left as an exercise to the reader that the autocorrelation function is:

$$\Gamma_I = \bar{I}^2 \left[1 + \left| 2 \frac{J_1\left(\frac{\pi D r}{\lambda z}\right)}{\frac{\pi D r}{\lambda z}} \right|^2 \right] \quad (15)$$

Let's now see how we could find the size of the speckles. One approach is to compute the equivalent area of the normalized covariance function. The normalized covariance function is related to the autocorrelation by:

$$\rho_I(\Delta x, \Delta y) = \frac{\Gamma_I(\Delta x, \Delta y) - \bar{I}^2}{\text{Var}(I)} \quad (16)$$

where $\text{Var}(I)$ is the variance of the intensity (also written as σ_I^2), defined by $\text{Var}(I) = \bar{I}^2 - \bar{I}^2$ ⁸. Then, the equivalent area is defined by:

$$\mathcal{A}_c = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho_I(\Delta x, \Delta y) d\Delta x d\Delta y \quad (17)$$

With the help of equation 13 inserted in equation 17, we also have:

$$\mathcal{A}_c = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |\mu_A(\Delta x, \Delta y)|^2 d\Delta x d\Delta y$$

We can now use both examples done earlier and show that

$$\mathcal{A}_c = \frac{(\lambda z)^2}{A}$$

for either the rectangular spot and the circular spot. Since the geometry is 2D, one cannot define a unique way to compute the 1D width. A good approximation is the square root of the equivalent area.

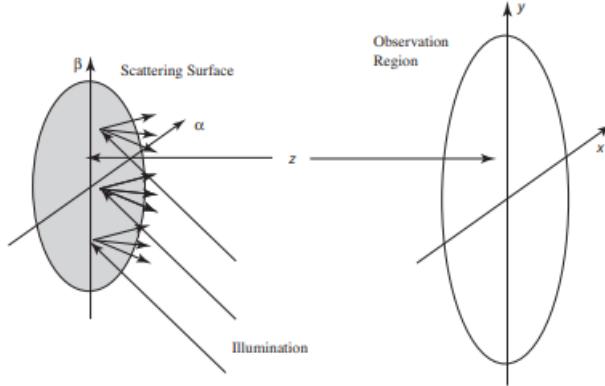


Figure 3: Freespace geometry (taken from [1])

⁸The first term is the mean of the squared intensity, while the second is the square of the intensity mean. They must not be interchanged, otherwise the standard deviation could be imaginary!

2.1.2 Imaging geometry

This part will be very short and sweet. An imaging geometry (like the one shown in figure 4) can be as simple as a coherent light (like a laser!) illuminating a rough object. Then, the diffused light goes through a lens and is focused at the image plane. Let z_o be the distance between the object (i.e. the rough surface) and the lens, and z_i be the distance between the lens and the image. Also, let's suppose that the lens has a focal length of f . To find the autocorrelation function at the image plane, we can use an approximation proposed by [1]. First of all, we must make two assumptions 1) the wavefront incident on the lens has phase variations approximately uniform over 2π 2) The speckle size incident on the lens is very small compared to the lens' aperture. Then, we can treat the lens as δ -correlated effective scattering source. This allows us to only consider the propagation from the lens to the image plane. In most cases, the lens has a circular aperture and the results obtained in the previous section holds, except for the fact that z_i is used instead of z . The intensity autocorrelation function is then given by equation 15. A more *in depth* explanation of the assumptions and other comments are made in [1].

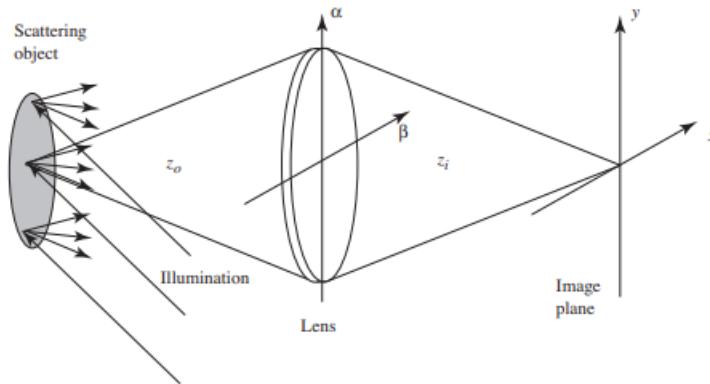


Figure 4: Imaging geometry (taken from [1])

We now have seen a theoretical way of measuring the "mean speckle size" via the equivalent autocorrelation size. In the next section, we will see a more *practical* way to find the mean speckle size, but it will also involve the autocorrelation.

3 Computation of speckles' average size

The average size of speckles is an important parameter that needs to be find. There are two great families of methods to find the average speckle size from a speckle image:

1. Morphology;
2. Autocorrelation.

This sections will analyze both families⁹. Both have specific advantages and disadvantages, so that is why it is pertinent to see them in depth.

3.1 Morphology

The first family is morphology. This regroups methods using classical morphology techniques, such as segmentation, blob detection and Hough transform. Morphology techniques are quite used in image processing and are included in a handful of programming language modules, like *OpenCV*[4], *scikit-image*[5]. The main advantages are (the order is not important):

1. Algorithms well defined (implemented in modules);
2. Can count the speckles;
3. Can generate a distribution of sizes.

I will now go over those 3 advantages, one by one.

3.1.1 Algorithms well defined

This one may not be obvious, but it is an important part. Morphology algorithms are widely used in a wide variety of contexts. Segmentation is a huge part of image analysis and there are some algorithms that can be implemented, like watershed segmentation. Segmentation algorithms are known and already implemented in modules. *Scikit-image* offers some algorithms for example. Instead of more complex methods like segmentation, some prefer to use basics manipulations of opening and closing[6]. Image opening is used to remove small object in the foreground, while closing removes small holes. Such methods can be combined to find and extract shapes in an image. This is the basis of blob detection. Another interesting approach involves the Hough transform. This transform is specific in finding circular objects, like speckles. The Hough transform uses a parameter space to find and recreate circular patterns in an image. It is used in computer vision, but could also be applied to speckle analysis.

3.1.2 Can count speckles

Morphology methods are specialized in object finding in images. They are used to distinguish different foreground and background objects. Thus, they can be used to count said distinguished objects. Some methods can do it by their nature, like segmentation and Hough transform, but basic methods must be modified and improved to count objects.

⁹For more info about the method that was used in the *DCCLab*, please refer to <https://github.com/DCC-Lab/Documentation/blob/master/HOWTO/HOWTO-MeasureMeanSpeckleWidthOnInFocusImagesOfSpeckles.pdf>.

3.1.3 Can generate a distribution of sizes

Like the previous advantage mentioned, morphology allow the counting of objects. Another step would be to have the different size of each object. Morphology techniques allow it and it can be quite easy to generate a distribution of sizes. It is fairly easy with Hough transform since it generates circles. Getting the radius of each circle is not complicated. It is also easy and natural to access sizes with opening and closing, because it uses kernels and if we use circular kernels of specific radii, we can extract some sort of size distribution. With segmentation, it can be a little tricky because it finds all sorts of possible shapes.

With the advantages explained, it is time to dive into the bad and the ugly. There is one big disadvantage and it is the required parameters. Morphology techniques can be really powerful, but they require parameters that are mostly unknown when working with laser speckles. With the wrong parameters, the algorithms don't work properly or give bad results. Let's take the example of opening and closing. One way to get a distribution of sizes is to apply the algorithm with a certain kernel size, then increment the size. Rinse and repeat. However, if one does not know where to stop, one can count two speckles as one. If we take for example the watershed segmentation, one needs to know the minimum spacing between speckles, otherwise we have a result that merged multiple speckles. For the Hough transform, we need the minimum radius and/or spacing, if not also the maximum radius and/or spacing! One other downside of those methods is that speckle delimitation may not be clear. For example, does a certain speckle has a diameter of 10 or 15? This is caused by the contrast between background and foreground being kind of unclear, and morphology tends to fail more often in this case.

3.2 Autocorrelation

Autocorrelation is a specific variant of the cross-correlation where the two random variables are in fact the same. In our case, we compute the cross-correlation of the image with itself. The mathematical definition of cross-correlation and autocorrelation will be presented in annexes. For those interested in why the autocorrelation is used can take a look at section 2.1 for a theoretical derivation of the autocorrelation function and the equivalent area. I can count two major advantages of the autocorrelation method:

1. Fast!
2. Does not require any parameter

3.2.1 Fast algorithm

The computation of the cross-correlation is quite tedious. When doing it numerically, it takes a lot of computer juice to iterate over the data. When using 1D samples, it can be *ok* when there are not a lot of data. But when working with fairly large images (let's say 1000×1000 pixels²), you better get access to a super computer or that you have time to kill. Meanwhile, the special cross-correlation case called autocorrelation can be computed really fast with the use of the Wiener-Kinchin theorem. Long story short, this theorem stipulates that the inverse Fourier transform of the power spectrum density of the data is the autocorrelation, or alternatively that the Fourier transform of the autocorrelation gives the power spectrum density. This can be written as:

$$\rho_I(x,y) = \frac{\mathcal{F}^{-1}(|\mathcal{F}(I(x,y)|^2) - \bar{I}^2}{\text{Var}(I)} \quad (18)$$

where \mathcal{F} represents the Fourier transform, \mathcal{F}^{-1} represents the inverse Fourier transform. In practical cases where we have a speckle image, $I(x,y)$ represents the intensity value at pixel (x,y)

and the power spectrum density is simply given by $|\mathcal{F}(I(x, y)|^2$. The mathematical development to achieve equation 18 is not that hard and presented by a few authors, so it is left as an exercise to the reader. In our case where we use computer and numerical methods to compute the mean speckle size, the Fast Fourier Transform, a.k.a the FFT, will be used in order to get a fast computation of the autocorrelation.

3.2.2 No parameters required

This one advantage is not negligible when we want to automatize image processing, in this case the computation of the mean speckle size. The algorithm using the autocorrelation works by itself. Computing the autocorrelation of the image is the same for every single image that existed, exists or will exist. There is no parameter that needs to be known, like the maximum spacing between speckles. The mean intensity is something that is fairly easy to get, it is simply the mean pixel value. Same thing for the variance, it is simply the variance of the pixels.

The big disadvantage of the autocorrelation method is that it requires another step in order to get the mean speckle size. It was explained in the last section (section 2.1) that the speckle size was obtained by the equivalent area of the autocorrelation. This last step is to find the full width at half maximum, FWHM, of the central peak of the autocorrelation. Since we compute the autocorrelation of the an image (2D array), we have a 2D autocorrelation, which allows us to find an horizontal and a vertical speckle size. This part requires some sort of parameters, but in an ideal case where the data is not discrete, we should not require them.

4 Python implementation of speckle size analysis

This section is aimed at those who want to use *DCCLab*'s implementation of the speckle size analysis. The code follows an *object-oriented* structure, which follows this general structure:

1. Read a speckle image
2. Compute the autocorrelation with [18](#)
3. Find the FWHM of the autocorrelation in both dimensions
4. Give information to the user

This is really only what the code does. Some general information about the files and the classes will be presented in the next subsections, but the reader is encouraged to visit the GitHub and take a look at the documentation of the different classes and methods. The code is available [here](#).

4.1 Reading the image

The wanted speckle image is read in the `autocorrelation.py`, using the class `FileReader`, which can read most used image format like *PNG* or *TIFF*. It can also remove a background image from the target one.

This class is in `autocorrelation.py`, because images are read from the `Autocorrelation` class, which is the main one in the current file.

4.2 Computing autocorrelation

Located in the `autocorrelation.py` file, the `Autocorrelation` class takes an image and computes the autocorrelation using the Wiener-Khinchin theorem. It can also be used to show the autocorrelation, either the full 2D one or the horizontal/vertical slices.

Speaking about the slices, the last class in the current file is `AutocorrelationSlices`, which takes a 2D autocorrelation array. It finds the 1D slices either at some specific indices or at the middle (the *more* relevant slices).

4.2.1 Preprocessing

Some preprocessing is done before computing the autocorrelation. In fact, two things *can* be applied to the speckle image:

1. Normalization by a gaussian filter
2. Removing small artifacts with a median filter

The first one, normalization, is simply dividing the speckle image by a gaussian filtered speckle image. This is good for removing non uniformity of the illumination. For example, see figure [5](#), where the left hand image is a uniform illumination speckle pattern, while the center image is the same speckle pattern, but filtered in a way that it looks like a non uniform speckle pattern. Finally, the right hand image is the non uniform speckle pattern that is normalized using the gaussian

filter. We can see that it is pretty good at recreating the original image. The only parameter is the standard deviation of the filter, which is 75 pixels by default and does a nice job. It could be hand picked to be better, but for this purpose, it was the default value.

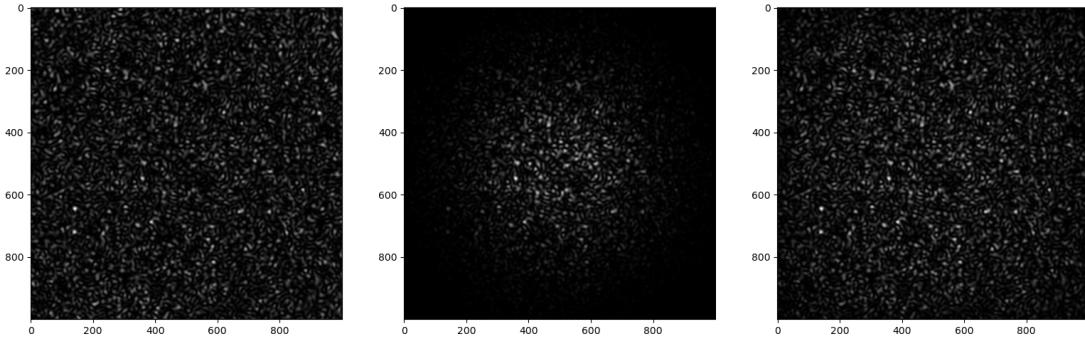


Figure 5: Simulation of speckles pattern. The image at the right is the uniform speckles, while the image next to it is the non uniform version. The image at the left is the corrected image using the gaussian filter normalization

The second one is really simple: we apply a median filter on the image.

It is important to state that those preprocessing steps are *optional*. If we give a standard deviation of 0, then no normalization is done. The same thing applies to the median filter. If we give a kernel size of 0, no filtering is done. Furthermore, it is important to understand that these preprocessing methods can alter the original speckle sizes, so the user must be careful when using them.

4.3 Finding the FWHM of the autocorrelation array

Finding the FWHM of the autocorrelation array is done in the `peakMeasurement.py` file. Two main ways are done to find the FWHM: with the mean or with a linear fit. Since the autocorrelation array is a discrete array, there is no analytic formula to use, so we must work with discrete data points. As for the FWHM, the autocorrelation array is normalized, because its minimum is (around) 0 and the maximum is (around) 1. That said, the FWHM is then the width of the central peak where its value is 0.5 (half of 1). Having no way to inverse it and find the roots where the data is 0.5, we must use our imagination.

4.3.1 The mean method

Like I said, we work with discrete data points, so we cannot find the inverse and the values of x where $y = 0.5$, we must use some workarounds. The figure 6 explains the reason behind the

mean method. First, we find some number of points where the value of the data is around 0.5. For example, we use a threshold of 20% and we search the data points in the range $0.5 \pm 0.5 \times 20\%$. Then, the indices where this is true are averaged, giving the *approximation* of the indices where the data is 0.5. Since the peak is symmetrical, we only take the left hand part of the peak (or right hand part) and do the difference with the centre of the peak. This gives us the half width at half maximum, HWHM. When we multiply this value by 2, we get the FWHM.

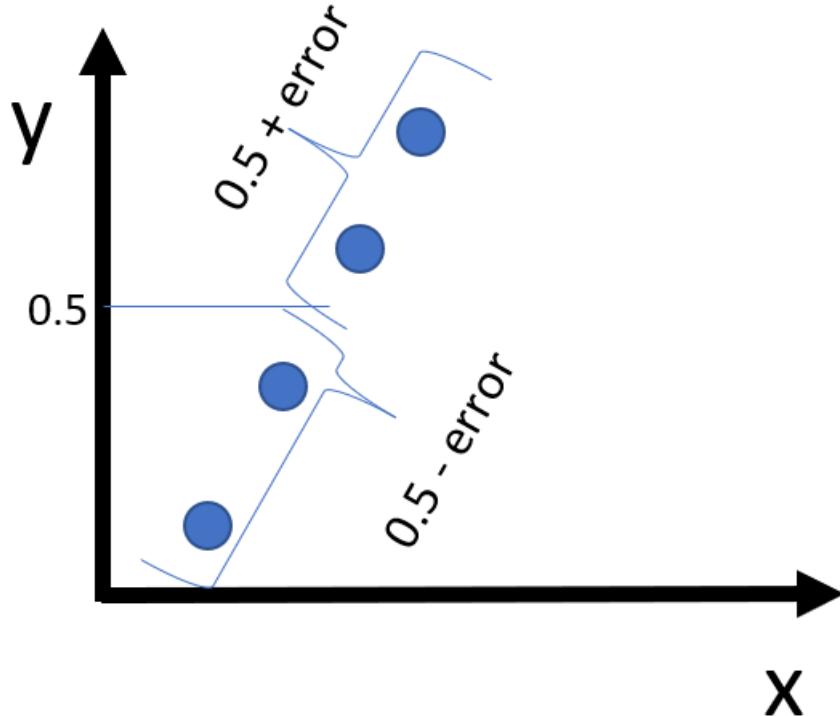


Figure 6: Example of the mean method. We look around the neighbourhood of $y = 0.5$ and find the indices where the data is in the neighbourhood. Then, the mean of the indices is done (the x values).

4.3.2 The linear fit method

This method is very similar to the mean approximation method, but instead of the mean, we do a linear fit on the neighbourhood data and estimate the value where the data is 0.5.

These 2 methods are done in the `peakMeasurement.py` file, with different classes for the different methods. The classes are not very relevant to the users, only the way the methods work is important.

4.4 Giving information to the user

4.4.1 Individual pieces of information

The user can get individual pieces of information in the `speckleCaracterization.py` file. There are methods to crop the image to a certain region of interest, to compute the FWHM on a specific axis or even to compute the local contrast of the speckles. A more experienced user may be interested in this file and the `SpeckleCaracerization` class, but the best is to use the all in one class.

4.4.2 All the info at the same time

The all in one class is in the `speckleStatsReport.py` file, under the class `SpeckleStatsReport`. There are lots of methods available, some can be used to see individual aspects of the speckle image, like the local contrast or the FWHM according to a specific axis. It can also print a full report of the different statistics computed, as well as show it as a graphical report.

4.5 Example of the code usage

This example presents how to access the statistical information of a single speckle image. All of this is done in the `speckleStatsReport.py` file, under the line `if __name__ == '__main__':`:

```
<...>

if __name__ == '__main__':
    # Insert here the path to the speckle image
    path = <...>

    # Here, we create a SpeckleStatsReport object.
    # We left every parameter by default (pre-processing ones)
    # and specified that the mean method should have a threshold of 20%
    ssr = SpeckleStatsReport(path, averageRange=20 / 100)

    # Here, we print the text report in the console
    print(ssr.textReport())

    # Here, we show the graphical display. The False should ALWAYS be there
    ssr.fullGraphicsReportDisplay(None, False)
```

The console output gives us:

```
===== Statistical properties of the speckle image =====
Mean intensity : 0.06619860976934433
Intensity std deviation : 0.06615938246250153
Maximum intensity : 1.0
Minimum intensity : 6.353977966000457e-08
Global contrast : 0.9994074300505783
Contrast modulation : 0.9999998729204488
This is a fully developed speckle pattern
(based on the intensity histogram, its maximum is at 0, assuming exponential distribution)
===== Statistical properties of the speckles =====
Vertical diam. : 4.0 pixels
Horizontal diam. : 4.0 pixels
```

Vertical FWHM finding method : Error/neighbors average method ($\pm 20.0\%$).
 For more info, see the method's 'fullMethodInfo'.** Info only on the HWHM finding **
 Horizontal FWHM finding method : Error/neighbors average method ($\pm 20.0\%$).
 For more info, see the method's 'fullMethodInfo'.** Info only on the HWHM finding **
 Local contrast mean : 0.8060457910406968
 Local contrast std deviation : 0.1901639101681258
 Local contrast median : 0.7852397705902351
 Local contrast min : 0.1887520004187369
 Local contrast max : 2.4287441894782287

The graphical report looks like:

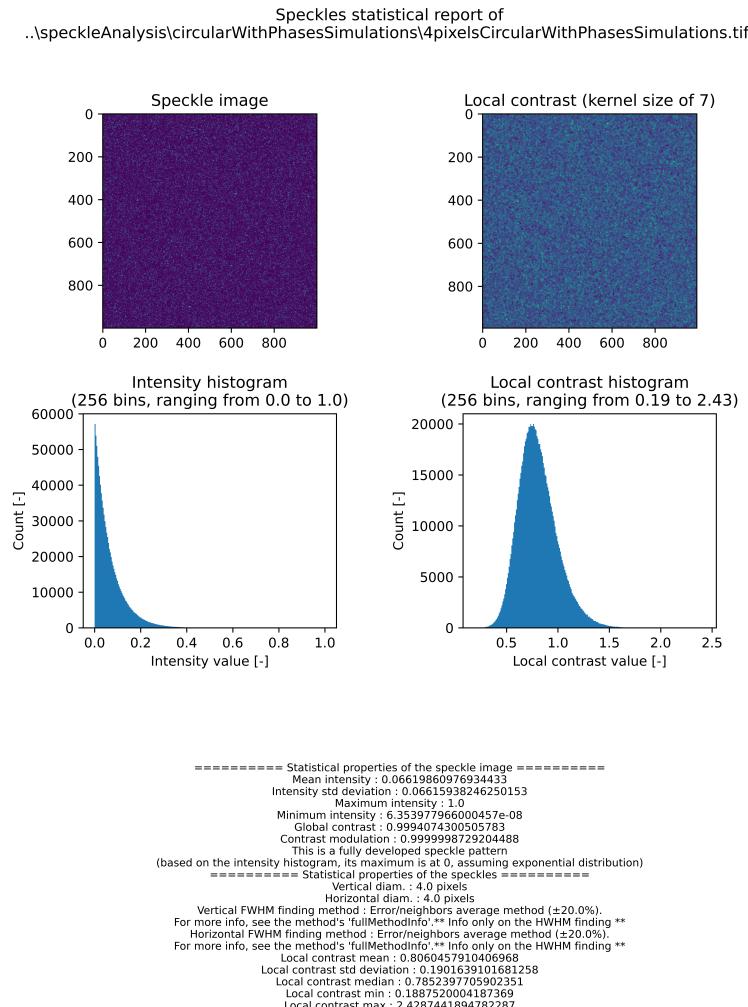


Figure 7: Graphical report of the speckles' statistical analysis

4.6 GUI implementation: what to know and how to use

The GUI approach is not mandatory to the user. If one wishes to go through the code and run directly the files, it is possible, although it can become annoying. This is why we implemented a small GUI, really not fancy, where we can select a speckle image and apply the preprocessing and access the stats from different panels in the GUI's window. The GUI is run from the `speckleStatsGUI.py`. Simply running the file will open the main window where we can play with a specific speckle image. The figure 8 shows what the window looks like when no picture is loaded, while figure 9 shows that the continue button is accessible and the preprocessing + processing can be done. Finally, figure 10 shows how the window looks like when all is done. We can access different panels and extract the information either from the text fields at the bottom or as a figure, like it was done with the *no GUI* version.

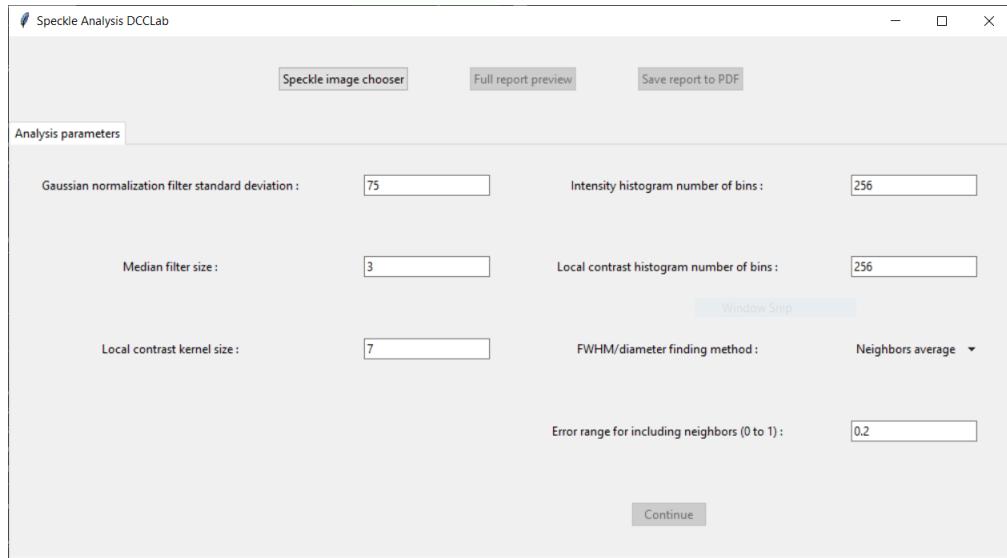


Figure 8: GUI when no image is selected

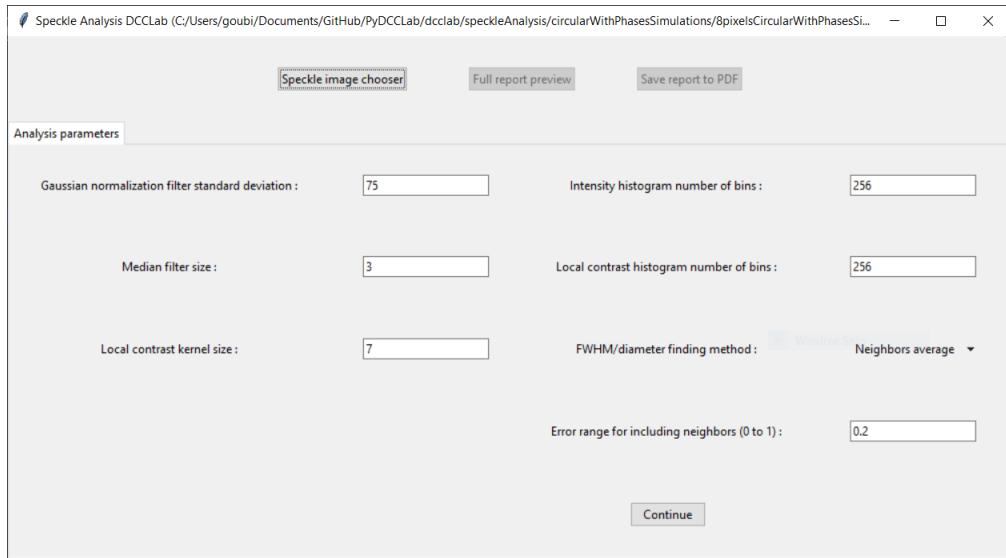


Figure 9: GUI when an image is selected

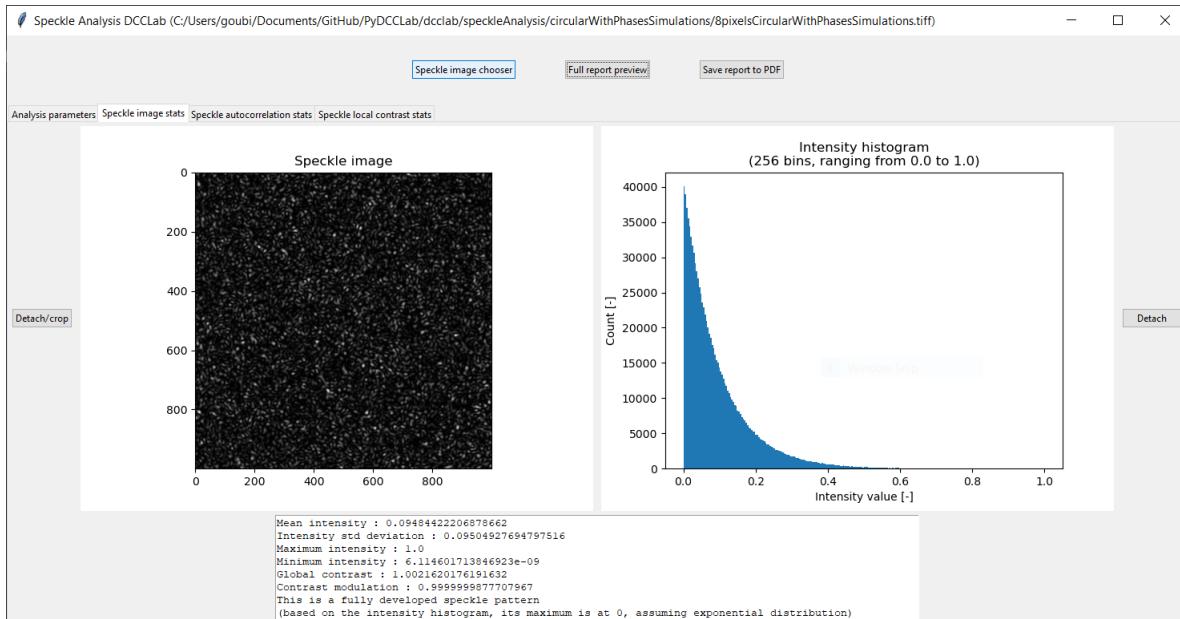


Figure 10: GUI when the stats are computed

The GUI is straight forward to use and even has text boxes explaining what the different parameters do. It has different panel as shown in figure 10 and every image is detachable to be able to zoom or save individually. Finally, we can have the full report like we did earlier (report shown at figure 7).

5 Conclusion

This document went over the mathematical basis of the speckle phenomenon, as well as presenting *DCCLab's Python* implementation of a speckle analysis algorithm. First, we explained with mathematics why speckles behave like they do and why they are so fascinating using only basic to moderate statistical analysis. Then, we saw how we can measure the size of the speckles in different geometries, this involved more complicated and tedious math, but still we arrived at the autocorrelation function. Finally, we presented our implementation of the autocorrelation function in *Python* using basic tools like the fast Fourier transform and simple filters.

The last pages of this document are annexes where we present more mathematical concepts that can be useful to the reader, but that couldn't really fit in the main core of the previous sections, including more about statistics and speckles.

References

- ¹J. Goodman, *Speckle phenomena in optics: theory and applications* (Roberts & Company, 2007).
- ²Wikipedia, the free encyclopedia, ed., *Rayleigh distribution*, 2020.
- ³Z. Hajjarian and S. K. Nadkarni, “Tutorial on laser speckle rheology: technology, applications, and opportunities”, *Journal of Biomedical Optics* **25**, 1–19 (2020).
- ⁴G. Bradski, “The OpenCV Library”, Dr. Dobb’s Journal of Software Tools (2000).
- ⁵S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, “Scikit-image: image processing in Python”, *PeerJ* **2**, e453 (2014).
- ⁶D. Lecompte, A. Smits, S. Bossuyt, H. Sol, J. Vantomme, D. Van Hemelrijck, and A. Habraken, “Quality assessment of speckle patterns for digital image correlation”, *Optics and Lasers in Engineering* **44**, 1132–1145 (2006).

Annexes

A The Delta function

The Delta function, also known as the Delta distribution or the Dirac Delta, is represented by the symbol δ and is widely used in physics. It is a distribution which is defined by:

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

It also satisfies the following identities¹⁰:

1. $\int_{-\infty}^{\infty} f(x)\delta(x - x_0) = f(x_0)$, which can be generalized in n dimensions
2. $\delta(g(x)) = \sum_{i=1}^n \frac{\delta(x-x_i)}{|g'(x_i)|}$ where x_i are defined by $g(x_i) = 0$ and n is the number of different values that x_i can take. Also, $g'(x_i) \equiv \left. \frac{dg(x)}{dx} \right|_{x=x_i} \neq 0$

B Some moderately advanced notions of statistics

Statistics are really important in natural science. As we have seen throughout this document, speckle patterns follow some very common probability distributions under the right assumptions and conditions. But some more advanced knowledge may be required in order to fully understand the simplicity and the beauty of the speckle statistics.

B.I Central Limit Theorem

The central limit theorem, *CLT*, is an important theorem in statistics. It stipulates that sums of probability densities can be approximated by a Gaussian distribution, when the number of densities is large. An important aspect of the summed distributions is that they must be statistically independent. A more formal mathematical statement would be:

Let X_1, X_2, \dots, X_n be n independent distributions taken from a population¹¹ and let σ_p, μ_p be respectively the standard deviation and the mean of X_p , with $p \in \{1, 2, \dots, n\}$. We can define a new variable, Y defined by $Y = \sum_{p=1}^n X_p$. Then, under some general conditions, such as $n \rightarrow \infty$ ¹², Y can be defined by a normal distribution of mean $n\mu$ and variance $n\sigma^2$.

A general case of the *CLT* is when the variable is reduced and centered. This means that its mean is 0 and its variance is 1. This can be written by:

$$Z = \frac{Y - \sum_{p=1}^n \mu_p}{\sqrt{\sum_{p=1}^n \sigma_p^2}}$$

where Z follows a normal distribution of zero mean and unity variance. This case is more general, because it works for a sum of different distributions. For example, it would work if we sum 10^3 exponential distributions with 10^5 Weibull distributions and $10^4 + 2$ uniform distributions.

¹⁰This list is not exhaustive. I only present the important identities that are used in this guide.

¹¹This case requires that they follow the same distribution. For example, they must all be exponential distributions.

¹²It is worth noting that the theorem still holds when n is not so big. The approximation is still good with sums of a few hundreds or a few dozens.

B.II Change of variable in statistics

This is a relatively advanced topic. It can be counter intuitive to a non experienced reader. The first method is the easier one, because it is simple and doesn't require hard maths.

1. If the change of variable is bijective... Let X be a random variable whose possible values are in the interval $[0, \infty)$. Its probability density is $p_X(x)$. If we want instead the probability density $p_Y(y)$ with $Y = X^2$, we need to proceed to a change of variable. Since the range of X is $[0, \infty)$, Y is strictly increasing and hence is bijective¹³. With some *not-so-relevant-for-this-purpose* maths, it can be found that the probability density $p_Y(y)$ is simply:

$$p_Y(y) = p_X(f^{-1}(y)) \left| \frac{df^{-1}(y)}{dy} \right| = p_X(\sqrt{y}) \left| \frac{1}{2} y^{-\frac{1}{2}} \right|$$

We can also generalize this method to n dimensions with:

$$p_{\mathbf{Y}}(\mathbf{y}) = p_{\mathbf{X}}(f^{-1}(\mathbf{y})) \left| \det \left[\frac{\partial f^{-1}(\mathbf{y})}{\partial \mathbf{y}} \right] \right| \quad (19)$$

2. If the change of variable is not bijective... Let's take the previous example of $Y = X^2$, but instead of the interval $[0, \infty)$, let's consider the interval $(-\infty, \infty)$. With this range, the function is not bijective and we cannot use the first method. We must use something more complicated. The association $X \rightarrow Y$ must be made with¹⁴:

$$p_Y(y) = \int_{R_X} p_X(x) \delta(y - f(x)) dx \quad (20)$$

Knowing property 1 of the Delta distribution, we could solve this integral without too much trouble, but x_0 must not be a function of x , unlike equation 20. Thus, we must use property 2 of the Delta distribution. Let $g(x) = y - f(x)$, we then find the zeros x_i of $g(x)$ and $g'(x_i)$. We can then solve the following integral:

$$p_Y(y) = \int_{R_X} p_X(x) \delta(g(x)) dx = \sum_{i=1}^n \frac{1}{|g'(x_i)|} \int_{R_X} p_X(x) \delta(x - x_i) dx = \sum_{i=1}^n \frac{p_X(x_i)}{|g'(x_i)|} \quad (21)$$

Equation 21 may not seems very complicated, but the thought behind it is quite complex. It is not very hard to make some mistakes when using it, so be careful!

We now have seen how to apply a change of variable in statistics. We will use this to find some distributions.

¹³The fact that the function is strictly increasing is sufficient but not necessary for the function to be bijective. For example, $f(x) = ax + b$ is bijective and $f(x) = -ax + b$ is also bijective. A function can be bijective if it is injective and surjective. An injective function is a function for which $f(a) = f(b)$ happens if and only if $a = b$. An example of a non injective function would be $f(x) = x^2$ over all the real values, because $f(x) = f(-x)$ and $x \neq -x$ (except for 0). A surjective function is a function for which there exists a value a for every value b such that $f(a) = b$. An example of a non surjective function would be $f(x) = \sqrt{x}$ over all the real values, because $f(a), a \in (-\infty, 0)$ doesn't return a real value.

¹⁴ R_X simply means "the set of values that X can take". It is the support of X .

B.III Sum of probability distributions

The sum of probability densities is something that requires some knowledge about Fourier transforms and convolutions. When we are dealing with a lot of distributions, we can invoke the *CLT*, but when we sum two or three distributions, we must proceed carefully. Let X and Y be two different random variables and let $Z = X + Y$. In the discrete case, the distribution of Z is:

$$p_Z(z) = \sum_{k=-\infty}^{\infty} p_X(k)p_Y(z-k)$$

which is the discrete convolution of $p_X(x)$ with $p_Y(y)$. This equation holds only if X and Y are independent. If X and Y are continuous and independent, we have:

$$h(z) = (f * g)(z) = \int_{-\infty}^{\infty} f(z-t)g(t)dt = \int_{-\infty}^{\infty} f(z)g(z-t)dt \quad (22)$$

C Random phasors

Throughout this document, the notion of phasor was important to the understanding of the speckle phenomenon. This subsection aims to mathematically describe the concept of random phasors and their statistics.

C.I What is a phasor

A phasor is complex number representing a sinusoidal function. It is most of the time written in the form $Ae^{i\theta}$ where A is the amplitude of the function and θ is the phase of the function. Phasor notation is widely used in optics and electromagnetic studies. Instead of writing the electric field in terms of cosine and sine, which is annoying most of the time, we can simply write it in a simple, yet complete form.

C.II Sum of random phasors

For this subsection, it is important to understand that a sum of phasors is a phasor, just like a sum of vectors is a vector. The definition of a sum of phasors is:

$$\mathbf{A} = Ae^{j\theta} = \frac{1}{\sqrt{N}} \sum_{n=1}^N \mathbf{a}_n = \frac{1}{\sqrt{N}} \sum_{n=1}^N a_n e^{i\theta_n} \quad (23)$$

where A and θ are respectively the amplitude and the phase of the resultant phasor, \mathbf{a}_n is the complex amplitude, while a_n and θ_n are respectively the amplitude and the phase of every individual phasor composing the sum. \mathbf{A} is known as the complex amplitude of the final phasor. The factor $N^{-\frac{1}{2}}$ is there to make sure the second moment of the sum is finite[1]. The sum can also be written in terms of imaginary and real parts:

$$\mathcal{R} = \text{Re}(\mathbf{A}) = \frac{1}{\sqrt{N}} \sum_{n=1}^N a_n \cos \theta_n \quad (24)$$

$$\mathcal{I} = \text{Im}(\mathbf{A}) = \frac{1}{\sqrt{N}} \sum_{n=1}^N a_n \sin \theta_n \quad (25)$$

where $\sum_{n=1}^N a_n (\cos \theta_n + i \sin \theta_n) = \sum_{n=1}^N a_n e^{i\theta_n}$. To get the joint probability density given at equation 1, we must consider some approximations or make some assumptions:

1. The amplitudes a_n and a_m are statistically independent for $n \neq m$. The same assumption must be made for the phases θ_n and θ_m . In other words, knowing a_n gives no information about a_m (same thing for the phases).
2. The amplitudes a_n and phases ϕ_n are statistically independent of each other. In other words, knowing a_n (or θ_n) gives no information about θ_n (or a_n).
3. The phases θ_n are uniformly distributed between $-\pi$ and π . This means that each value in the interval $[-\pi, \pi]$ has the same probability.

With that, we can easily find the mean and the variance of \mathcal{R} and \mathcal{I} :

$$\begin{aligned}\bar{\mathcal{R}} &= \frac{1}{\sqrt{N}} \sum_{n=1}^N \overline{a_n \cos \theta_n} = \frac{1}{\sqrt{N}} \sum_{n=1}^N \underbrace{\overline{a_n} \times \overline{\cos \theta_n}}_{\text{assumption 1}} = 0 = \bar{\mathcal{I}} \\ \text{Var}(\mathcal{R}) &= \overline{\mathcal{R}^2} - \bar{\mathcal{R}}^2 = \frac{1}{N} \sum_{n=1}^N \overline{a_n^2} \times \overline{\cos^2 \theta_n} = \frac{1}{N} \sum_{n=1}^N \frac{\overline{a_n^2}}{2} (\overline{\cos(2\theta_n)} + 1) = \frac{1}{2N} \sum_{n=1}^N \overline{a_n^2} \\ \text{Var}(\mathcal{I}) &= \overline{\mathcal{I}^2} - \bar{\mathcal{I}}^2 = \frac{1}{N} \sum_{n=1}^N \overline{a_n^2} \times \overline{\sin^2 \theta_n} = \frac{1}{N} \sum_{n=1}^N \frac{\overline{a_n^2}}{2} (1 - \overline{\cos(2\theta_n)}) = \frac{1}{2N} \sum_{n=1}^N \overline{a_n^2}\end{aligned}$$

We can see that the mean of the real and the imaginary parts is 0, while their variance is the same.

C.III Joint probability distribution of the real and imaginary parts of the phasors sum

We have seen earlier that the real and imaginary parts can be written as:

$$\begin{aligned}\mathcal{R} &= \text{Re}(\mathbf{A}) = \frac{1}{\sqrt{N}} \sum_{n=1}^N a_n \cos \theta_n \\ \mathcal{I} &= \text{Im}(\mathbf{A}) = \frac{1}{\sqrt{N}} \sum_{n=1}^N a_n \sin \theta_n\end{aligned}$$

We have also seen that a large number of distributions can be approximated as a Gaussian distribution. Let's use this fact and with a large number of phasors we can say that the distributions of \mathcal{R} and \mathcal{I} are both Gaussian. This allows us to write the following joint probability distribution:

$$p_{\mathcal{R}, \mathcal{I}}(\mathcal{R}, \mathcal{I}) = \frac{1}{2\pi\sigma^2} \exp \left\{ -\frac{\mathcal{R}^2 + \mathcal{I}^2}{2\sigma^2} \right\} \quad (26)$$

where $\sigma^2 = \sigma_{\mathcal{R}}^2 = \sigma_{\mathcal{I}}^2 = \frac{1}{2N} \sum_{n=1}^N \overline{a_n^2}$. This distribution is known as the Rayleigh distribution[2].

C.IV Joint probability distribution of the amplitude and the phase of the phasors sum

As written before, the sum of phasors is a phasor. In this case, what is the probability distribution of the amplitude and the phase of the resultant phasor? Well, it is given at equation 1. We can find this equation by manipulating equation 26 and using changes of variable. We use the first method of change of variable and obtain:

$$p_{\mathcal{R}, \mathcal{I}} = \frac{1}{2\pi\sigma^2} \exp \left\{ -\frac{\mathcal{R}^2 + \mathcal{I}^2}{2\sigma^2} \right\} \rightarrow p_{A, \theta} = \frac{A}{2\pi\sigma^2} \exp \left\{ -\frac{A^2}{2\sigma^2} \right\}$$

where $A = \sqrt{\mathcal{R}^2 + \mathcal{I}^2}$ and $\theta = \arctan\left(\frac{\mathcal{I}}{\mathcal{R}}\right)$. One can also easily find the marginal probability densities of A and θ by using this formula:

$$p_X(x) = \int_{R_Y} p_{XY}(x,y)dy$$

$$p_Y(y) = \int_{R_X} p_{XY}(x,y)dx$$

C.V Probability density of the speckle pattern intensity

Like I said in a previous section, we measure the speckle intensity in the lab, not \mathcal{R} , \mathcal{I} , A or θ . It is also widely known that the intensity of an electromagnetic wave written in phasor form is $I = |\mathbf{A}\mathbf{A}^*|$ where $*$ is the complex conjugate. Using this definition of the intensity and equation 23, we find that the intensity of the speckle pattern is given by $I = Ae^{j\theta} \times Ae^{-j\theta} = A^2$. We can then apply a change of variable $I = A^2 \implies A = \sqrt{I}$ and use method number 2 of the change of variable:

$$p_I(I) = \int_0^\infty p_A(A)\delta(I - A^2)dA \quad (27)$$

Posing $g(A) = I - A^2$:

$$g(A_i) = 0 \implies 0 = I - A^2 \implies A = \sqrt{I}$$

There are then 2 zeros: $A_1 = +\sqrt{I}$ and $A_2 = -\sqrt{I}$. But, since a negative amplitude doesn't make any sens, we let go of the second zero¹⁵. We can now use the final form of equation 21 and find:

$$p_I(I) = \frac{1}{|g'(A_1)|} \frac{\sqrt{I}}{\sigma^2} \exp\left\{-\frac{I}{2\sigma^2}\right\} = \frac{1}{2\sqrt{I}} \frac{\sqrt{I}}{\sigma^2} \exp\left\{-\frac{I}{2\sigma^2}\right\} = \frac{1}{2\sigma^2} \exp\left\{-\frac{I}{2\sigma^2}\right\}$$

One could easily find that $2\sigma^2 = \bar{I}$ and write:

$$p_I(I) = \frac{1}{\bar{I}} \exp\left\{-\frac{I}{\bar{I}}\right\}$$

which is exactly equation 4.

¹⁵One could argue that since the negative root is physically impossible, the square root function is (in this case) bijective and we could use method 1.

D Sum of speckle patterns

As we have seen with equation 4, the intensity of the speckle pattern follows an exponential law. But what happens when the speckle pattern is composed of speckle patterns? Well, we must use the sum of random variables. Let's first of all consider the case where two fully developed speckle patterns are summed (or acquired). Let $I_t = I_1 + I_2$, where I_1 and I_2 are the intensity of the first pattern and second pattern respectively. We know that the final intensity distribution of I_t is calculated with equation 22 and in the case of exponential distributions, the convolution is not so bad:

$$\begin{aligned} p_{I_t}(I_t) &= \int_{-\infty}^{\infty} p_{I_1}(I_1)p_{I_2}(I_t - I_1)dI_1 = \frac{1}{\bar{I}_1}\frac{1}{\bar{I}_2}\int_0^{I_t} \exp\left\{-\frac{I_1}{\bar{I}_1}\right\} \exp\left\{-\frac{I_t - I_1}{\bar{I}_2}\right\} dI_1 \\ &= \frac{1}{\bar{I}_1\bar{I}_2} \exp\left\{-\frac{I_t}{\bar{I}_2}\right\} \int_0^{I_t} \exp\left\{-\frac{I_1}{\bar{I}_1}\right\} \exp\left\{\frac{I_1}{\bar{I}_2}\right\} dI_1 = \frac{1}{\bar{I}_1\bar{I}_2} \exp\left\{-\frac{I_t}{\bar{I}_2}\right\} \int_0^{I_t} \exp\left\{I_1\left(\frac{1}{\bar{I}_2} - \frac{1}{\bar{I}_1}\right)\right\} dI_1 \\ &= \frac{1}{\left(\frac{1}{\bar{I}_2} - \frac{1}{\bar{I}_1}\right)\bar{I}_1\bar{I}_2} \exp\left\{-\frac{I_t}{\bar{I}_2}\right\} \left[\exp\left\{I_t\left(\frac{1}{\bar{I}_2} - \frac{1}{\bar{I}_1}\right)\right\} - 1\right] = \frac{1}{\left(\frac{1}{\bar{I}_2} - \frac{1}{\bar{I}_1}\right)\bar{I}_1\bar{I}_2} \left[\exp\left\{-\frac{I_t}{\bar{I}_1}\right\} - \exp\left\{-\frac{I_t}{\bar{I}_2}\right\}\right] \\ &= \frac{1}{\bar{I}_1 - \bar{I}_2} \left[\exp\left\{-\frac{I_t}{\bar{I}_1}\right\} - \exp\left\{-\frac{I_t}{\bar{I}_2}\right\}\right] \end{aligned}$$

There is a special case where $\bar{I}_1 = \bar{I}_2 = \bar{I}$:

$$p_{I_t}(I_t) = \frac{1}{\bar{I}^2} \exp\left\{-\frac{I_t}{\bar{I}}\right\} \int_0^{I_t} \exp\left\{I_1\left(\frac{1}{\bar{I}} - \frac{1}{\bar{I}}\right)\right\} dI = \frac{1}{\bar{I}^2} \exp\left\{-\frac{I_t}{\bar{I}}\right\} \int_0^{I_t} dI = \frac{I_t}{\bar{I}^2} \exp\left\{-\frac{I_t}{\bar{I}}\right\}$$

The case where there are N speckle patterns is not presented here, but one would find equation 5 after doing the big maths.