3-1-2009

Evaluation of Performance of a Maximum Likelihood Estimator for Tracking Purposes in the Presence of Speckle Noise

Adrian Brett Monz

Evaluation of Performance of a Maximum Likelihood Estimator for Tracking Purposes in the Presence of Speckle Noise

THESIS

Brett Monz, Flight Lieutenant, RAAF

AFIT/GE/ENG/09-29

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/09-29

Evaluation of Performance of a Maximum Likelihood Estimator for Tracking Purposes in the Presence of Speckle Noise

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Brett Monz, BEng HonsI.

Flight Lieutenant, RAAF

26 March 2009

AFIT/GE/ENG/09-29

Evaluation of Performance of a Maximum Likelihood Estimator for Tracking Purposes in the Presence of Speckle Noise

Brett Monz, BEng HonsI.
Flight Lieutenant, RAAF

Approved:

/signed/	18 Feb 2009
Maj J. Schmidt, PhD (Chairman)	date
/signed/	18 Feb 2009
Dr. Stephen C. Cain (Member)	date
/signed/	18 Feb 2009
Dr. M. Fickus (Member)	date

Abstract

In many cases, optical tracking systems do not have cooperative beacons available. This is particularly true for the case involving tracking a laser illuminated target such as a missile seeker head, where the object of interest is an extended source. Furthermore the extended source is often observed in the presence of noise such as shot and speckle noise as well as atmospheric turbulence which further degrades the signal. This research effort presents the evaluation of an existing algorithm based on the maximum-likelihood technique for tilt estimation in the presence of extended sources and speckle noise, with particular application to the image motion tracking problem. Comparison is made between the performance of traditional centroiding algorithms and the existing projection-based correlation algorithm in simulation. The Maximum Likelihood Estimator using projection-based correlation is shown to offer improved performance in the motion tracking problem.

Acknowledgements

First and foremost, this is for my dad who sadly passed away before this project was completed. I also owe a large debt of gratitude to my wife for her tolerance, support and most of all her continuous challenging of my self imposed limitations. I also thank my committee members for their willingness to answer my seemingly endless and sometimes obvious questions.

Brett Monz

Table of Contents

Page

Abstract iv

Acknowledgements v

Table of Contents vi

List of Figures viii

List of Tables x

List of Abbreviations xi

I. Introduction 1
- 1.1 Motivation 1
- 1.2 Goals 2
- 1.3 Thesis Outline 3

II. Theory and Review 4
- 2.1 Point and Extended Sources 4
- 2.2 Atmospheric Turbulence 10
- 2.3 Speckle Noise 15
  - 2.3.1 Speckle Phenomena 15
  - 2.3.2 Speckle Intensity Statistics 17
  - 2.3.3 Speckle Photon Count Statistics 22
- 2.4 Tilt Estimation 26
- 2.5 Description of Algorithms 28
  - 2.5.1 Centroid algorithm 28
  - 2.5.2 Projection algorithm 29

III. Simulation Structure and Models 35
- 3.1 Simulation Structure 35
  - 3.1.1 Simulation Description 36
- 3.2 Simulation Models 39
  - 3.2.1 Optical Transfer Function Model 39
  - 3.2.2 Extended Source Model 39
  - 3.2.3 Speckle Noise Model 39
  - 3.2.4 Atmospheric Turbulence Model 41
  - 3.2.5 Centroid Method 43
  - 3.2.6 Projection-based Method 44
- 3.3 Simulation Metrics 45

Page

IV. Simulation Results 47
4.1 Simulation using Extended Source No Speckle 47
4.2 Simulation using Extended Source with Speckle 47
4.3 Effect of Atmospheric Turbulence and Speckle 50

V. Conclusions and Further Work 51
5.1 Summary 51
5.2 Key Results 53
5.3 Recommendations 55

Appendix A. Matlab Code 56
A.1 Parent Code 56
A.2 Make Otf function 62
A.3 Centroiding Algorithm 63
A.4 Projection Method Algorithm 2 64
A.5 Projection Method Algorithm 3 65
A.6 Vector Linear Shift Algorithm 66
A.7 Atmospheric Turbulence Simulation 67

Bibliography 68

Index 70

List of Figures

Figure		Page
2.1.	Light rays from a near extended object.	5
2.2.	Light rays from a distant extended object.	5
2.3.	Defining a point source.	6
2.4.	Variation in irradiance from a point source on a detector at distance r	7
2.5.	Extended source considered as a collection of point sources when viewed(a) at normal to the source and (b) from an angle θ to normal.	8
2.6.	Lambertian source where the intensity reduces as the cosine of the angle between the normal and the direction of propagation.	9
2.7.	Typical Hartmann lenslet array.	14
2.8.	(a) Conventional image point source case. (b) Conventional image extended beacon case. [21]	15
2.9.	Typical Speckle Noise Pattern.	16
2.10.	Speckle formation in the image of a rough object. (redrawn from [9])	17
2.11.	Negative Exponential distribution with mean $\bar{I} = 5$.	18
2.12.	Typical Young's Experimental Setup (redrawn from [9])	19
2.13.	Poisson distribution with mean $\overline{K} = 5$	25
2.14.	Bose-Einstein distribution with mean $\overline{K} = 5$	25
2.15.	Wavefront Tilt Geometry - redrawn from [26]	27
2.16.	Projection transformation employed on a sensor	30
3.1.	Conceptual Diagram of Simulation for the Centroiding case	36
3.2.	Conceptual Diagram of Simulation for the Projection-based case.	37
3.3.	Extended source Model	40
3.4.	Image of extended source	40
3.5.	Image of speckle source	42

Figure Page

3.6. Sample phase screen created by the FFT method in this simulation. 44

4.1. Computed Tilt Error for the Extended Source Case. 48

4.2. Comparison of Speckle Noise Results. 49

4.3. Computed Tilt Error for the Extended Source Case. 50

5.1. Initial Speckle Model Image 53

5.2. Computed Mean Tilt Error for Speckle Case 54

List of Tables

Table		Page
2.1.	Names and definitions of Various measures of Coherence used in reference to Figure 2.12 (reproduced from [9])	20
3.1.	Simulation Parameter Space	36

List of Abbreviations

Abbreviation		Page
MLE	Maximum Likelihood Estimation/Estimator	1
IR	Infrared	1
AO	Adaptive Optics	2
OI	Object of Interest	2
PSF	Point Spread Function	6
PDF	Probability Density Function	17
RV	Random Variable	18
OPL	Optical Path Length	26
ML	Maximum Likelihood	30
OTF	Optical Transfer Function	38

Evaluation of Performance of a Maximum Likelihood Estimator for Tracking Purposes in the Presence of Speckle Noise

I. Introduction

This research aims to evaluate a tilt estimation algorithm for tracking purposes that performs well in the presence of speckle noise. In particular, it addresses the type of noise produced when coherent light is reflected off an optically rough surface, as distinct from the twinkling viewed by astronomers which is caused by atmospheric turbulence. Maximum Likelihood Estimation (MLE) algorithms have been shown to give good performance compared to traditional centroiding in the presence of noise. An MLE algorithm, used in the presence of speckle noise, is investigated in simulation. Its performance in terms of tilt error is compared to that of traditional centroiding.

1.1 Motivation

In the field of Infrared (IR) counter measures, two basic systems are employed: a detection and tracking system, and a jamming system. On airborne platforms this leads to several limitations. Firstly, space and weight are at a premium on most platforms especially those of the fighter type. Secondly, with the detection system also providing tracking information, its ability to detect and process new threats can be limited. Thirdly, a large jamming beam foot-print is often required, as the location in space of the threat to be jammed is not known to any useful degree of precision. This is due to the fact that the exhaust plume of the threat is usually the element tracked, not the threat sensor itself. These limitations can be partially overcome if the tracking part could be integrated with the jamming system. The physical size of the systems would most likely be smaller, particularly the detection element. Once detection is accomplished and the threat has been handed over to the jamming

system, the detection system can revert to searching for new threats. Finally, if some additional threat information is known, the spatial location of the threat sensor to be jammed can be computed to a reasonable degree of accuracy, allowing a smaller beam footprint and providing increased power density in the jamming beam.

Imaging wavefront sensors, such as the well known Shack-Hartmann, currently used in the adaptive optics (AO) field to provide wavefront subaperture tilt estimates, can be used to provide estimates of global tilt. This global tilt can be used to provide information on a source's position relative to the optical axis of the tracking sensor. When this information is fed to a tracking system, the sensor can be adjusted such that the source is kept on the optical axis. When this information is also provided to the illuminating source in the case of an IR jamming system, the efficiency of the jammer can be much improved. Most sensors employ some form of tilt estimation algorithm to derive an estimate of the global tilt parameter. These algorithms generally perform well in scenarios where the object of interest (OI) occupies a small portion of the overall imaged scene and is the single brightest feature in it, and there is relatively little to no noise of any type. When the first conditions is not met, ie. the OI occupies a significant portion of the scene, the OI is termed non co-operative. This is the case for real-world tracking scenarios involving coherent illumination, where there is also noise present from the reflected light, the scene background, and from within the sensor itself.

Few, if any, existing algorithms have been shown to perform well in the perviously mentioned scenarios. An algorithm that performs well in the presence of such limiting factor would provide an increased level of tracking accuracy in real-world situations.

1.2 Goals

The primary objective of this research is to evaluate the performance of a maximum-likelihood tilt estimator in the presence of noise, particularly speckle. The research focuses on the performance of existing tilt estimation algorithms in applica-

tion for which they were not originally intended. To satisfy this goal, the research investigates the performance of an existing algorithm in ideal scenarios as well as in non-ideal scenarios. The ideal scenarios are used to establish a baseline for performance of each algorithm. Non-ideal effects are introduced, and the change in performance of each algorithm is examined.

1.3 Thesis Outline

This thesis meets the goals through detailed analysis and simulation. Chapter II examines the theory behind the important concepts of extended sources and speckle noise, including the statistics of speckle, upon which this research is based. It also examines several of the existing techniques used to estimate tilt in optical wavefront sensing. Chapter III includes a detailed explanation of the structure of the simulations used and reasoning for any assumptions made. Chapter IV presents the results of the simulation with MATLAB® as well as offering reasoning for the results. Chapter V provides concluding remarks as well as possible future research opportunities in extending this topic.

II. Theory and Review

This work builds on previous research, and moreover, this section reviews previous work in the field of tilt estimation with extended sources. It begins with the basic theory of extended and point sources and then explores other important and relevant areas that are the foundations of this research.

2.1 Point and Extended Sources

While point sources seem at first glance to be the simplest beacons to use in simulation, this is in fact not entirely true. If one were to accurately model a point source, it would have to be represented as a Dirac delta function in space. It has infinite bandwidth, which is clearly not possible in simulation because of a discretely sampled grid and impossible in a real-world situation. Firstly we attempt to define the physical differences between point and extended sources. When trying to characterize point and extended sources, it is often helpful to start with the geometrical perspective. A true point source at a finite distance emits divergent light in all directions, while at very large distances light from a point source can be considered collimated [12]. An extended source produces overlapping pencils of rays called beams from each object point. Figures 2.1 and 2.2 illustrate the spreading effect of an extended source.

A point source can be thought of in a number of ways. One is to simply refer to it as resolved or unresolved, in that it either occupies more than one pixel in the image sensor or not. However, this may not capture the physics of any propagation involved. Another is an idealized source whose dimensions are very small compared to the viewing distance, i.e. the product of the lateral source dimensions should be very much smaller than the distance to the detector or observation screen squared and the included solid angle is very small. Hence we can define a point source as shown Figure 2.3.

Radiant energy emitted by a point source is referred to as isotropic, i.e. it radiates equally in all directions. The surface area through which the radiation passes is $4\pi r^2$ at a distance r from the source. If a detector of area A is placed a distance r

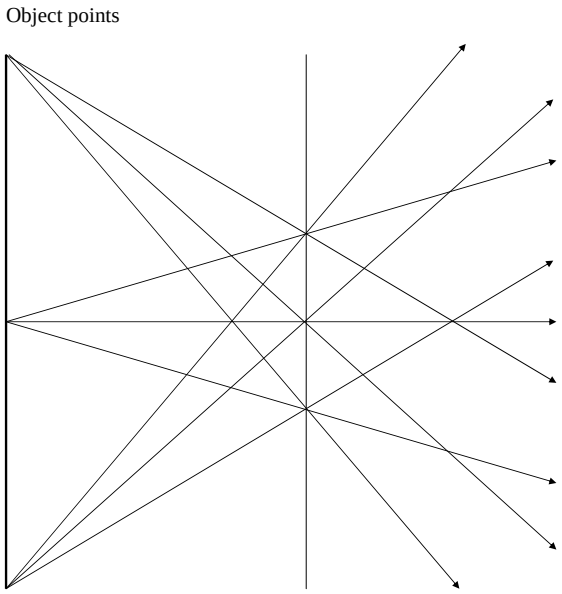


Figure 2.1: Light rays from a near extended object.

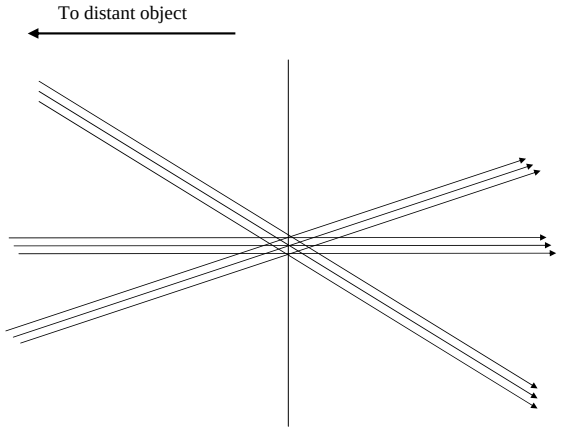


Figure 2.2: Light rays from a distant extended object.

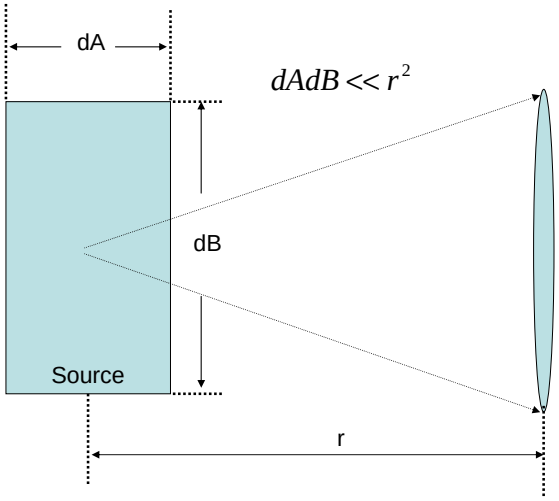


Figure 2.3: Defining a point source.

from the source, the detector subtends a solid angle $\Omega = A/r^2$. The radiant intensity can be expressed by

$$I_e = \frac{\phi_e}{\Omega}, \tag{2.1}$$

where ϕ_e is the radiant flux, defined as the measure of the total power of electromagnetic radiation (including infrared, ultraviolet, and visible light) landing on a particular surface. The measured intensity remains constant as the detector moves away from the source. However, we can rarely neglect diffraction effects and the measured intensity takes the form of $I_\theta = I_\theta(0)\cos^3\theta$ from Ref. [17], illustrated in Figure 2.4.

Extended sources require special treatment to properly measure corresponding flux and surface brightness. This arises from the complex scattering of incident light reaching the sensor focal plane array. The images of extended sources have more extended wings than those of point sources. Photons that would normally be scattered out of the point spread function (PSF) aperture used to measure a point source are instead captured when an extended source is present. Extended sources can be considered collection of point sources as suggested in Ref. [19]. For an extended source, the point source model needs to be expanded to account for the area of the source and the fact that the radiant intensity may vary across this area. If we consider the

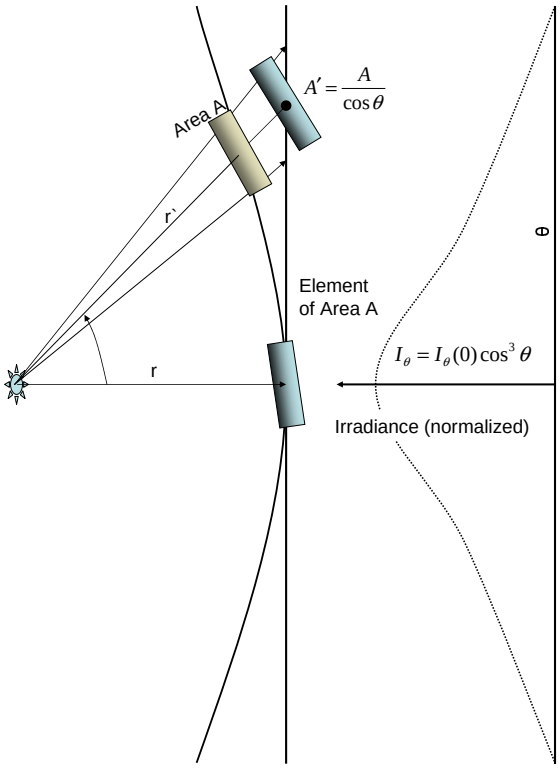


Figure 2.4: Variation in irradiance from a point source on a detector at distance r

extended source to comprise equally spaced point sources as shown in Figure 2.5(a), then the irradiance per unit area L_e can be expressed as

$$L_e = \frac{\phi_e}{A\Omega}, \tag{2.2}$$

where ϕ_e is the radiant flux of the source, A is the area of the source and Ω is the solid angle subtended by the detector at the source. If the same extended source is now viewed at an angle θ as shown in Figure 2.4, then the spacing between point sources is shortened in one direction and the area A is reduced by a factor $\cos\theta$ as in Figure 2.5(b). Thus A is replaced by $dA\cos\theta$ in Eq. (2.2) leads to

$$L_e(\theta) = \frac{L_e(0)}{\cos\theta}, \tag{2.3}$$

where $L_e(0)$ is the radiance at $\theta = 0$. Unfortunately with Eq. (2.3) we are about

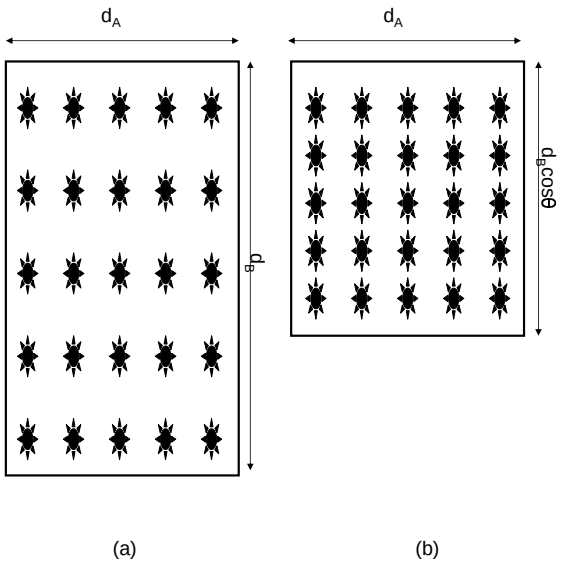


Figure 2.5: Extended source considered as a collection of point sources when viewed(a) at normal to the source and (b) from an angle θ to normal.

to violate a few physical laws in that it suggests that as θ decreases the radiance increases, eventually to infinity at $\theta = \pi/2$. One way to avoid this is to choose an extended source whose radiance is independent of viewing angle. This is possible if the radiant intensity falls off as $\cos\theta$ which compensates for the effect in Eq. (2.3), which now becomes

$$l_e(\theta) = \cos\theta \frac{\phi_e}{A\Omega}. \tag{2.4}$$

The extended source can be thought of as a lambertian source as shown in Figure 2.6 whose intensity varies as the cosine between the normal and the direction of propagation. It should be noted here that most practical sources do not strictly meet this requirement, though many are close enough to make the approximation valid. Now to calculate the irradiance at a distance r from the extended source, we note the irradiance produced by an element dA of the source is given by Eq. (2.2) which when substituted into Eq. (2.1) and after a bit of manipulation, gives the irradiance

$$E_e = \frac{L_e A}{r^2} = L_e \Omega. \tag{2.5}$$

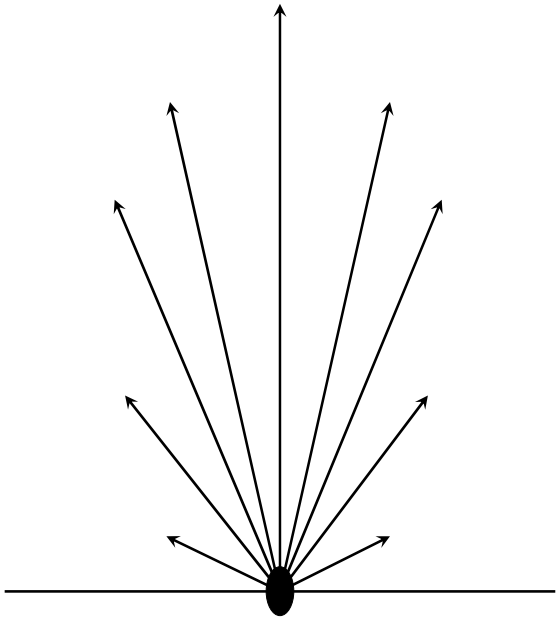


Figure 2.6: Lambertian source where the intensity reduces as the cosine of the angle between the normal and the direction of propagation.

The question then is, when is it no longer accurate to approximate a physical source by a point source? If we take the circular source irradiance as

$$E_p = \frac{\pi a^2 L}{r^2}, \tag{2.6}$$

where a is the source radius and r is the detector distance, the extended disk source irradiance is

$$E_e = \frac{\pi a^2 L}{a^2 + r^2}. \tag{2.7}$$

Now if we desire to have less than 1% error by approximating the disk as a point then

$$\frac{E_p - E_e}{E_e} \leqq 0.01 \Rightarrow a \leqq 0.01 \times r \tag{2.8}$$

This is the basis for the "5 times" rule of thumb, which is the observation distance should be at least five times the largest source dimension. The reader can compare this to the definition presented earlier in this section. The physical differences between

extended and point sources reduce wavefront sensing accuracy by various degrees depending on the sensor used.

2.2 Atmospheric Turbulence

Random variations in the temperature and pressure of the earth's atmosphere alter the the refractive index of the air, both spatially and temporally. Optical waves propagating through these variations are distorted. This distortion is known as turbulence. Turbulence affects all optical systems which propagate light through long atmospheric paths. There have been many theories and much work done on characterizing the effects of turbulence on optical propagation. A statistical analysis is necessary as it is not feasible to describe exactly the refractive index at all points in space and time. The most widely accepted theory, due to its consistency with observation, is that put forward by A.N. Kolmogorov [11]. His theory centers on randomly distributed pockets of air, called eddies, of varying sizes and temperatures causing the random variations in the refractive index of the atmosphere. From this, a refractive index profile of the section of atmosphere in question can be developed.

Kolmogorov suggested that turbulent flow, governed by the Navier-Stokes equations, could be described by the transfer of kinetic energy from large eddies into smaller eddies. The average size of these eddies being designated the outer scale L_0 for the large eddies and the inner scale l_0 for the smaller eddies. L_0 can range from the height above ground at lower altitudes up to hundreds of meters at higher altitudes. l_0 ranges from a few millimeters at low altitudes to centimeters higher up. The range of sizes between inner and outer scales is known as the inertial subrange, and Kolmogorov assumed that eddies within this range are statistically homogeneous and isotropic. It is, however, more accurate to say that properties such as refractive index and wind velocity have stationary increments. Kolmogorov determined that the average speed v of turbulent eddies is related to their scale size r via

$$v \propto r^{1/3}. \tag{2.9}$$

Kolmogorov's analysis of potential temperature (linearly related to regular temperature T) led to an expression for the refractive index at a point in space $\mathbf{r}$ given by

$$n(\mathbf{r}) = \mu_n(\mathbf{r}) + n_1(\mathbf{r}), \tag{2.10}$$

where $\mu_n(\mathbf{r})$ is the slowly varying mean of the refractive index value, and $n_1(\mathbf{r})$ is the deviation of the index from its mean value. This creates a zero-mean random process $n_1(\mathbf{r})$. At optical wavelengths considered here, the refractive index of air can be approximated by

$$n(\mathbf{r}) \cong 1 + 7.99 \times 10^{-5} \frac{P(\mathbf{r})}{T(\mathbf{r})} \text{ for } \lambda = 0.5\mu m, \tag{2.11}$$

where λ is the optical wavelength, P is the pressure in millibars and T is temperature in Kelvin. Then assuming each eddy has relatively uniform pressure, the variation in the refractive index is given by

$$dn = 7.99 \times 10^{-5} \frac{d\theta}{T^2}. \tag{2.12}$$

Variation in refractive index is directly proportional to the variation in potential temperature and as such, the refractive index structure function follows a similar power law to Eq. (2.9):

$$D_n(\mathbf{r}) = C_n^2 r^{2/3} \text{ for } l_0 < r < L_0, \tag{2.13}$$

where C_n^2 is the refractive-index structure function parameter, in $\mathrm{m}^{-2/3}$.

It is often necessary to have a spectral description of the fluctuation in refractive index. The Kolmogorov power spectral density $\Phi_n(\kappa)$ can be computed from Eq. (2.13) and is given by

$$\Phi_n(\kappa) = 0.033 C_n^2 \kappa^{-11/3} \quad \text{for} \quad \frac{2\pi}{L_0} < \kappa < \frac{2\pi}{l_0}, \tag{2.14}$$

where $\kappa = 2\pi\left(f_x\hat{\mathbf{i}} + f_y\hat{\mathbf{j}}\right)$ is the angular spatial frequency in rad/m.

Through the application of Rytov theory (Maxwell's equations and perturbation theory), the field can be written as

$$U(\mathbf{r}) = U_0(\mathbf{r})\exp[\psi(\mathbf{r})], \tag{2.15}$$

where $U_0(\mathbf{r})$ is the vacuum solution ($n_1 = 0$) of Maxwell's equations and $\psi(\mathbf{r})$ is the complex phase perturbation. Successive perturbations of the form

$$\psi(\mathbf{r}) = \psi_1(\mathbf{r}) + \psi_2(\mathbf{r}) + \ldots \tag{2.16}$$

can be used to compute statistical moments of ψ which in turn are used to compute statistical moments of the field. For example, turbulent phase screens such as those used in simulation in Section 3.2.4 are realizations of $\psi(\mathbf{r})$.

Rytov theory yields many parameters that can be used to characterize optical impact of turbulence. Two of these are the coherence diameter r_0 and the isoplanatic angle θ_0. both of these parameters are computed from the integrated moments of the structure parameter C_n^2. C_n^2 is the measure of the local turbulence strength at a point in space, and is a function the propagation distance Δz. Functions of C_n^2, known as *structure functions*, describe the local turbulence along a particular optical path. The coherence diameter r_0, also known as the Fried parameter, is given approximately by

$$D_\psi(\mathbf{r}) = 6.88\left(\frac{\mathbf{r}}{r_0}\right)^{5/3}. \tag{2.17}$$

Values of r_0 are typically 5-10 cm at visible wavelengths. The isoplanatic angle θ_0 is defined as the angle between two point sources for which the mean square phase differs by 1 rad^2. It may also be considered as the largest field angle over which the optical path through the turbulence does not vary significantly from the on-axis optical path

through the turbulence. Values for θ_0 at visible wavelengths are typically $5-10\mu$rad, looking directly overhead.

If the turbulence encountered by the beam is strong, then the beam from an extended source wanders over time. If a coherent source such as a laser is used to illuminate a rough target, the beam also exhibits random effects which include strong intensity and phase variations. These variations create challenging scenarios for wavefront sensing. Light rays arriving at a sensor aperture, such as a Shack-Hartmann wavefront sensor, from different points of an extended source have traveled significantly different atmospheric paths. Light corrupted by aberrations obtained from these different paths arrives superimposed at the aperture where it is used in the sensing system. Considering the light is already superimposed and has most probably arrived from many different directions, often separated by more than θ_0, the isoplanatic angle, given by ,

$$\theta_0 = \left(1.09k^2C_n^2L^{8/3}\right)^{5/3}, \tag{2.18}$$

for a constant C_n^2 path, or from points separated by distances exceeding r_0, the coherence diameter, given by

$$r_{0,sw} = \left[0.423k^2\int_0^{\Delta z} C_n^2(z)(\frac{z}{\Delta z})^{5/3}dz\right]^{-3/5}, \tag{2.19}$$

conventional wave-front sensing processes do not estimate the the same wave-front error that would be due to a point source. In many cases the error in estimation is significant and severely degrades the sensor's ability to correctly estimate the wave-front. Nominally, when an extended source is present, the intensity pattern in the pupil of the observation system should be a lower-contrast, blurred version of the intensity pattern expected of a point source.

Often a Hartmann-type sensor is used for wavefront sensing. The presence of an extended source in this case also creates some interesting effects. If the complex

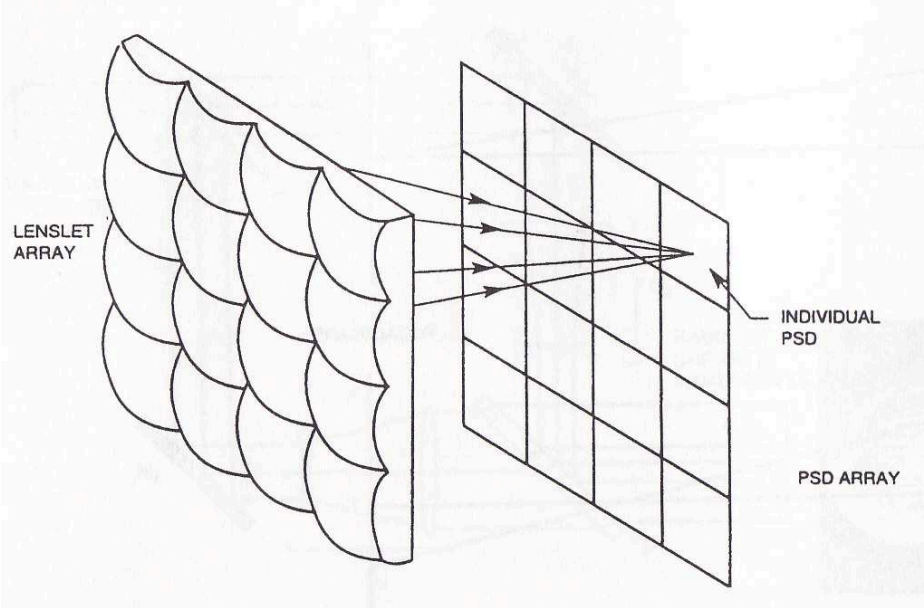


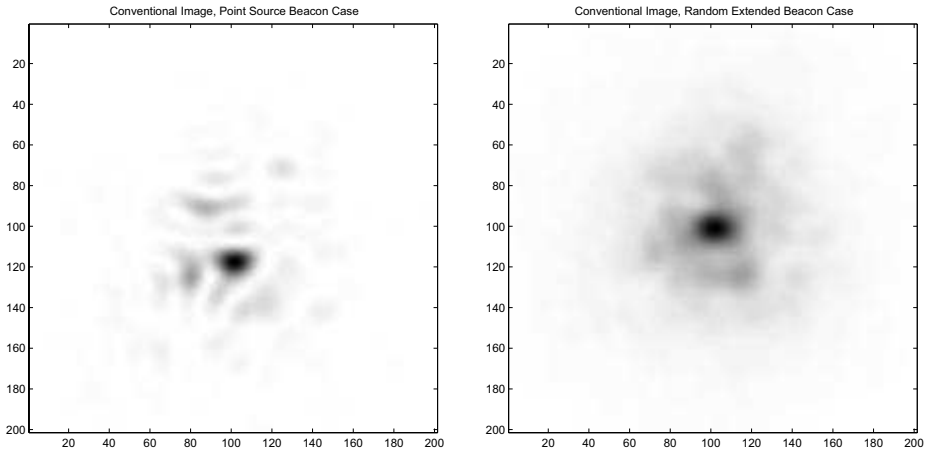
Figure 2.7: Typical Hartmann lenslet array.

transmittance of a Hartmann lenslet array such as the one shown in Figure 2.7 is given by

$$t_H(x_P) = \sum_S \exp[-j\frac{k}{2f_l}(x_P - x_s)^2]\text{rect}\left(\frac{x_P - x_s}{d}\right), \tag{2.20}$$

where S is the number of sub-apertures, x_s is the center of the s^{th} sub-aperture, f_l is the focal length, d is the side dimension of the lenslet and k is the wave number. When a field due to an extended source falls on a Hartmann array, due to the nature of the light from the extended source, some field segments may overlap onto adjacent sub-apertures. The resultant intensity is derived in Ref. [21] and is shown here as

$$I_H(x_H,t) = \int_{x_P}\int_{x'_P}\int_{x_T} dx_P dx'_P dx_T h_{NF}(x_P,x_H)h^*_{NF}(x'_P,x_H)t_H(x_P)t^*_H(x'_P) \times |U_T(x_T,t)|^2 h_A(x_T,x_P,t)h^*_A(x_T,x'_P,t), \tag{2.21}$$



(a) (b)

Figure 2.8: (a) Conventional image point source case. (b) Conventional image extended beacon case. [21]

where $h_{NF}(x_P, x_H)$ is the impulse response for the propagation from the lenslet array to the sensor detector plane, x_H is a coordinate in the Hartman detector plane and $h_A^*(x_T, x'_P, t)$ is the impulse response of the atmosphere [21]. The integrals over x_P and x'_P are propagations which move the field back from the lenslet array to the detector plane and their product becomes the detector intensity. The integral over x_T would, under isoplanatic conditions, be a convolution which describes the propagation of the field from the beacon to the aperture. However when considering an extended beacon and possibly anisoplanatic conditions associated with it, the Hartmann sensor has a significantly different result than for just a point source. This is illustrated in Figure 2.8 with images from Ref. [21] where image (a) clearly has a different centroid location from image (b).

2.3 Speckle Noise

2.3.1 Speckle Phenomena. When complicated objects are illuminated by highly coherent light of the type produced by a laser, an important type of image defect is seen. In particular, whenever the object is rough on the scale of an optical wavelength, the image has a grainy appearance. The contrast in the image is very pronounced, with large numbers of bright and dark spots. These spots apparently

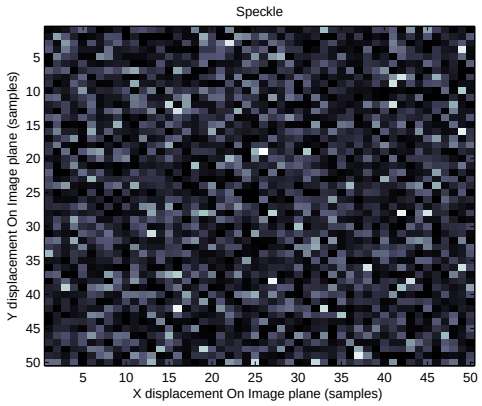


Figure 2.9: Typical Speckle Noise Pattern.

have relationship to the scattering properties of the surface of the scattering object. This image defect is called speckle. The effect is not limited to reflections from objects being also seen in images of transparent objects illuminated by coherent light through a diffuser. A typical speckle pattern is shown in Figure 2.9. Detailed analysis of speckle began in the 1960's. However, studies had actually been carried out far earlier (although not known as speckle at the time), by Verdet (1865) and Lord Rayleigh (1880) in their work on Fraunhofer rings. Von Laue [14–16] derived many of the basic speckle properties in the study of light scattered from a large number of particles.

The vast majority of surfaces are extremely rough on the scale of an optical wavelength [9] and under illumination by monochromatic light, the wave reflected form such a surface consists of contributions from many scattering points with random phases. The image formed at a given point in an observation plane consists of a multitude of amplitude spread functions, each arising from a separate point on the rough scattering surface as shown in Figure 2.10. As a result, the contributing spread functions have widely varying phase and when added together, produce a highly complex interference pattern.

The same arguments apply to transmission objects illuminated via a diffuser. The diffuser causes the exiting wavefront to have a highly complex and corrugated

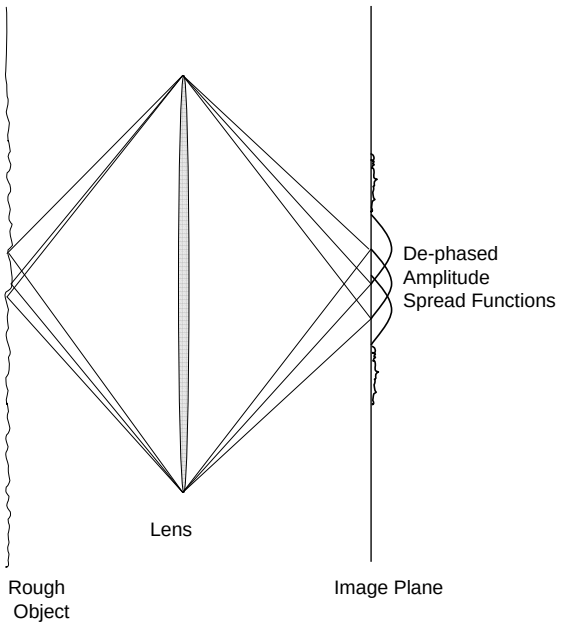


Figure 2.10: Speckle formation in the image of a rough object. (redrawn from [9])

structure. In the image of this type of object, we can again see the large intensity fluctuations caused by overlapping, dephased amplitude spread functions [9].

2.3.2 Speckle Intensity Statistics. Our knowledge of the exact wavelength-scale, structure of the wavefront leaving the surface is often extremely limited, so it is beneficial to think in terms of the statistical properties of speckle patterns. The statistics are defined over an ensemble of objects, all with the same properties but differing in the detail. If a detector is placed in the image plane at a precisely known position, the measured intensity cannot be predicted exactly. We therefore attempt to predict the intensity by using the statistical properties of the intensity over an ensemble of rough surfaces. A most important statistical property of speckle is the probability density function (PDF) of the observed intensity I at a point in the image. Effectively we are asking, how likely is it that we will see a spot of given intensity at that point? It is analogous to the well-known random walk [23, 24]. If the phases of the individual scattered contributions are uniformly distributed over $(-\pi, \pi)$, i.e. the object is rough on the scale of a wavelength, then the field associated with any single linear polarization component of the the image is a circular complex Gaussian random

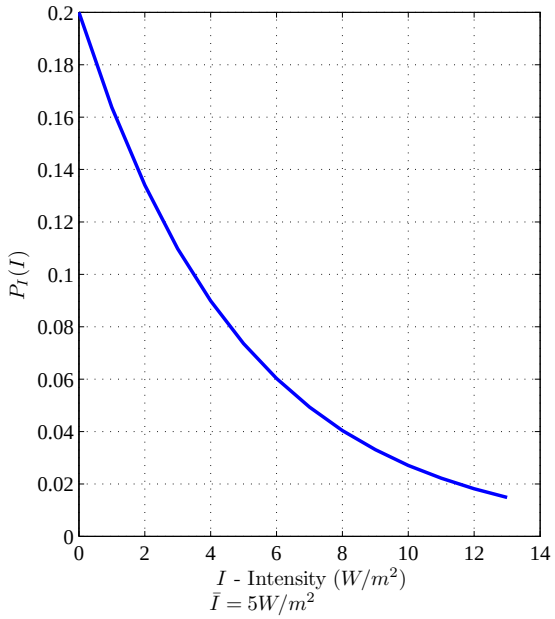


Figure 2.11: Negative Exponential distribution with mean $\bar{I} = 5$.

variable (RV), and its intensity statistics have a negative exponential distribution,

$$P_I(I) = \begin{cases} \frac{1}{\bar{I}} \exp\left(\frac{-I}{\bar{I}}\right) & \text{for } I > 0, \\ 0 & \text{elsewhere} \end{cases}$$

where $\bar{I}$ is the mean intensity associated with that polarized component. Figure 2.11 shows an example of such a negative exponential distribution. If the scattered wave is partially polarized, it can be shown that the density function for I is the difference of two negative exponential functions [9]. The negative exponential distribution nature of the speckle intensity implies the fluctuations around the mean are very pronounced. If we define the contrast of a speckle pattern as the ratio of standard deviation to mean, which for the polarized case, $C = \sigma_I/\bar{I} = 1$. It is because of this very high contrast that speckle is extremely disturbing to the human eye.

It is important to mention that the distribution of mean intensity $\bar{I}(x, y)$ in the image of a coherently illuminated object is identical to the image intensity that would be observed if the object were illuminated with spatially incoherent light of

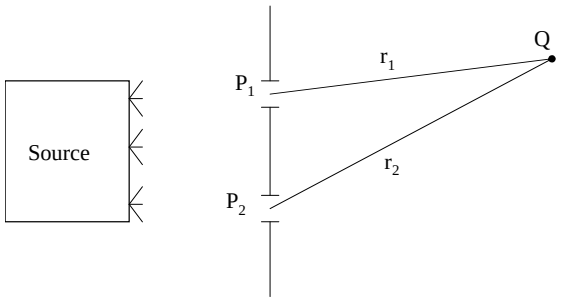


Figure 2.12: Typical Young's Experimental Setup (redrawn from [9])

the same spectral power density [9]. Incoherent illumination may be regarded as a rapid time sequence of spatially coherent wavefronts. Thus the time-integrated image intensity observed from spatially incoherent illumination is identical to the ensemble average intensity $\bar{I}(x, y)$ (assuming identical bandwidth). Therefore, methods used for incoherent intensity distribution may also be used for predicting the mean speckle intensity distribution with coherent illumination of a rough object.

When optically rough objects are illuminated by monochromatic light, the reflected wave cannot be treated as ergodic [9] as the time and ensemble averages are not equal. This can be demonstrated by considering two different Young's experiments of the type shown in Figure 2.12.

First, let light reflected from a rough surface fall on a mask containing two pinholes. The fringe formed is observed on a distant screen. As the light is monochromatic, it is also spatially coherent [9] and the fringe has visibility ν given by

$$\nu = \frac{2\sqrt{I_1 I_2}}{I_1 + I_2},$$

where I_1 and I_2 are the intensities of light falling on pinholes one and two, respectively. Since the distribution does not change with time and the coherence of the light has not been reduced by the amplitude and phase distribution imparted onto the wave by a rough surface, the modulus of the complex coherence factor $|\mu_{12}|$, as defined in Table 2.1, must be one (time averaged definition of coherence). In the second example, objects with different surface profiles are successively placed in the

Table 2.1: Names and definitions of Various measures of Coherence used in reference to Figure 2.12 (reproduced from [9])

Symbol	**Definition**	**Name**	**Temporal or Spatial Coherence**
$\Gamma_{11}(\tau)$	$\langle u(P_1, t = \tau u^*(P_1, t)\rangle$ [Note:$\Gamma_{11}(0) = I(P_1)$]	Self coherence function	Temporal
$\gamma_{11}(\tau)$	$\frac{\Gamma_{11}(\tau)}{\Gamma_{11}(0)}$	Complex degree of (self) Coherence	Temporal
$\Gamma_{12}(\tau)$	$\langle u(P_1, t = \tau u^*(P_2, t)\rangle$	Mutual coherence function	Spatial and temporal
$\gamma_{12}(\tau)$	$\frac{\Gamma_{12}(\tau)}{[\Gamma_{11}(0)\Gamma_{22}(0)]^{1/2}}$	Complex degree of coherence	Spatial and temporal
J_{12}	$\langle u(P_1, t = \tau u^*(P_1, t)\rangle = \Gamma_{12}(0)$	Mutual intensity	Spatial quasimonochromatic
μ_{12}	$\frac{J_{12}}{[J_{11}J_{12}]^{1/2}} = \gamma_{12}(0)$	Complex coherence factor	Spatial quasimonochromatic

illuminating beam, and all of the generated fringes are time integrated. Any one of these component fringes has a visibility corresponding to $|\mu_{12}| = 1$, but the superposition of the successive fringes does not because the phase from fringe to fringe is not constant as each amounts to a new realization. Therefore the ensemble averaged fringe produces a $|\mu_{12}|$ that is not equal to 1.

The wave equation governing the propagation of light remains the same for time- or ensemble-averaged properties of light. As such the laws governing the propagation of coherence functions for time- and ensemble-averaged qualities are identical. Therefore the mutual intensity, as defined in Table 2.1, of light reflected from a rough surface and observed very close to that surface is the same as that observed from an incoherent source. Over an ensemble of ideally rough surfaces there is little relationship between the phases of scattered light from two closely spaced ($< \lambda$) elements on the surface represented by a delta function mutual intensity,

$$\bar{J}(\xi_1, \eta_1; \xi_2, \eta_2) = \kappa \bar{I}(\xi_1, \eta_1)\delta(\xi_1 - \xi_2, \eta_1 - \eta_2),$$

where $\bar{I}$ is the ensemble average intensity distribution and κ is a constant. The mutual intensity observed at a distance z from the source can be computed using the Van Cittert-Zernike theorem and is given by

$$\bar{J}(x_1, y_1; x_2, y_2) = \frac{\kappa \mathrm{e}^{-j\psi}}{(\bar{\lambda}z)^2} \iint_{-\infty}^{+\infty} \bar{I}(\xi, \eta) \exp\left\{ j\frac{2\pi}{\bar{\lambda}z}[(\Delta x\xi + \Delta y\eta)] \right\} d\xi d\eta,$$

where $\psi = \frac{\pi}{\lambda z}[(x_2^2 + y_2^2) - (x_1^2 + y_1^2)]$, (ξ, η) is a point in the source plane, (x, y) is a point in the observation region, $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$, $\bar{\lambda}$ is the mean wavelength of the source and $\bar{I}(\xi, \eta)$ is the ensemble averaged intensity distribution across the scattering spot on the rough surface. Given the geometry as in Figure 2.10,

the mutual intensity in the image is given by

$$\bar{J}(u_1, v_1; u_2, v_2) = \frac{\kappa}{(\bar{\lambda} z_2)^2} \exp\left\{-j\frac{\pi}{\bar{\lambda} z_2}\left[(u_2^2 + v_2^2) - (u_1^2 + v_1^2)\right]\right\} \times \iint_{-\infty}^{+\infty} |P(x,y)|^2 \exp\left\{j\frac{2\pi}{\bar{\lambda} z_2}(\Delta u x + \Delta v y)\right\} dxdy,$$

where P is the complex pupil function of the lens, κ is a constant and $\Delta u = u_2 - u_1, \Delta v = v_2 - v_1$.

2.3.3 Speckle Photon Count Statistics. While intensity distribution statistics are both helpful and valuable aids in describing speckle, photon count statistics are at the heart of detection and motion estimation for speckle images. We consider a single-mode laser whose light falls on a detector and we wish to determine the distribution of the number of events in any τ-second interval. Assuming constant incident intensity, the integrated intensity on a pixel with area A is given by

$$W = I_0 A \tau,$$

with probability density of the form

$$P_W(W) = \delta(W - I_0 A \tau).$$

Substituting into Mandel's formula [18], in which the unconditional probability of observing K photo events can be expressed as

$$\begin{aligned} P(K) &= \int_0^\infty P\left(K|W\right) p_W\left(W\right) dW \\ &= \int_0^\infty \frac{(\alpha W)^K}{K!} \exp^{-\alpha W} p_W\left(W\right) dW, \end{aligned} \tag{2.22}$$

where α is a proportionality constant equal to the ratio of quantum efficiency to energy $\eta/h\bar{\nu}$ and $p_W\left(W\right)$ is the probability density function of the integrated intensity.

Performing the integration yields

$$P_K(K) = \frac{(\alpha I_0 A\tau)^K}{K!} \mathrm{e}^{-\alpha I_0 A\tau}. \tag{2.23}$$

Since the mean number of photon events $\overline{K} = \alpha I_0 A\tau$, we can rewrite Eq. (2.23) as

$$P_K(K) = \frac{(\overline{K})^K}{K!} \mathrm{e}^{-\overline{K}}.$$

From this, the mean in terms of the variance is $\sigma_K^2 = \overline{K}$. With care, this ideal model can be closely approximated in practice. There are two cases of photon count statistics for polarized thermal radiation:

1. Counting time shorter than coherence time, and
2. Counting for arbitrary time.

we begin with the first case, it gives a good approximation for the tracking scenario. Since the counting time is extremely short, the incident intensity can be assumed to be constant over the entire counting interval. Therefore, the integrated intensity is equal to the product of the intensity, the counting time, and the detector area:

$$W = I(t)A\tau.$$

The value of the intensity within that interval is random and obeys negative exponential statistics as discussed in Section 2.3.2. It follows that the same should be true of the integrated intensity, such that

$$P_W(W) = \frac{1}{\overline{W}} \exp\left(\frac{-W}{\overline{W}}\right), W \geqq 0.$$

We can now find the photon count statistics by substituting into Mandel's equation and integrating to give

$$P(K) = \frac{1}{1+\alpha\overline{W}} \left(\frac{\alpha\overline{W}}{1+\alpha\overline{W}}\right)^K.$$

Substituting in $\overline{K} = \alpha W$ gives

$$P(K) = \frac{1}{1+\overline{K}} \left(\frac{\overline{K}}{1+\overline{K}} \right)^K . \tag{2.24}$$

The distribution represented by Eq. (2.24) is the Bose-Einstein distribution which has variance equal to $\sigma_K^2 = \overline{K} + (\overline{K})^2$. Note that the first term, $\overline{K}$, represents the Poisson nature of the integration of light and matter and the second, $(\overline{K})^2$, represents the fluctuations of integrated intensity which are significant if $\overline{K} \gg 1$. If we examine the signal-to-noise ratio given by

$$\frac{S}{N} = \sqrt{\frac{\overline{K}}{1+\overline{K}}},$$

it can be seen that the S/N approaches 1 as $\overline{K} \to \infty$. This indicates that the count variation, just as with the intensity variation, is substantial. Probability masses associated with Poisson and Bose-Einstein distributions are shown in Figures 2.13 and 2.14, respectively. Comparison of these two figures shows that when the mean number of counts is greater than 1, the spread of the Bose-Einstein distribution is greater than that of a Poisson distribution and consequently fluctuations in photon count for Bose-Einstein is greater than for Poisson. Additionally, when the number of counts $\overline{K}$ is $\ll 1$, the differences between the two distributions is small, and it can be shown [9] that only one and zero events have significant probability so that the two distributions become asymptotically the same.

Alternatively, since this research is giving consideration to shot noise as well as speckle noise, the negative binomial distribution case shown by Goodman in Ref. [9] is also examined. This case includes the effects of both the random arrival nature of photons and the negative exponential distribution of speckle noise. The difference in this case is that the counting interval τ, previously assumed to be much shorter than the coherence time of the incident light, is now an arbitrary interval which maybe longer than the coherence time. Assuming that the wave incident on the sensor has a coherence area that is much larger than the area of the sensor, the approximate

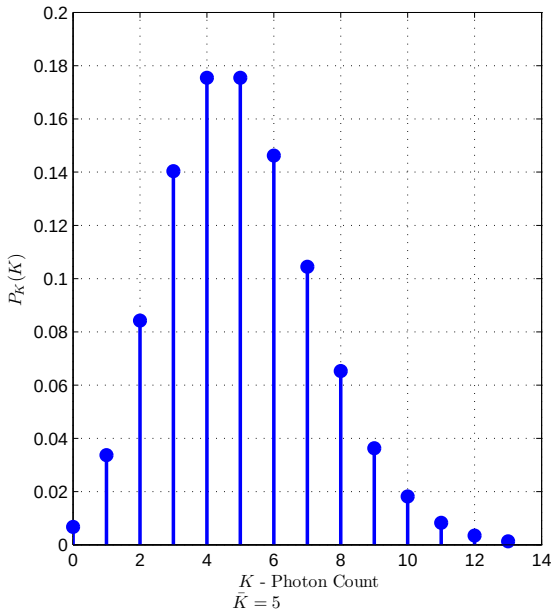


Figure 2.13: Poisson distribution with mean $\overline{K} = 5$

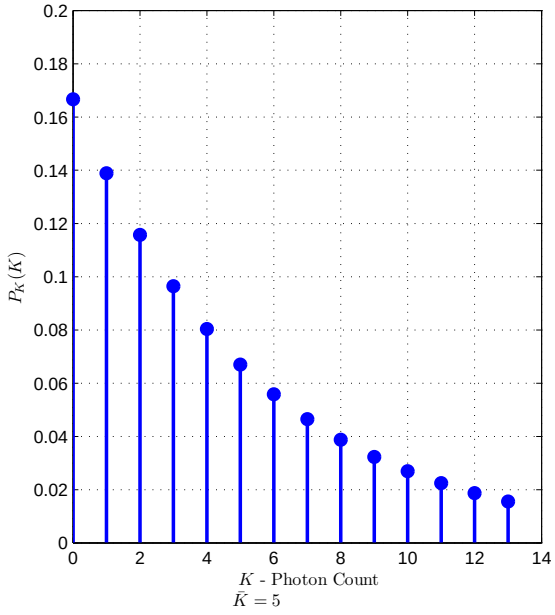


Figure 2.14: Bose-Einstein distribution with mean $\overline{K} = 5$

solution for the probability density $P_W(W)$ is a gamma probability density [9], given by

$$P_W(W) = \begin{cases} \left(\frac{\mathcal{M}}{\overline{W}}\right)^{\mathcal{M}} \frac{W^{\mathcal{M}-1}\exp\left(-\mathcal{M}\frac{W}{\overline{W}}\right)}{\Gamma(\mathcal{M})} & \text{for } W \geq 0, \\ 0 & \text{elsewhere} \end{cases}$$

where $\mathcal{M}$ is the number of degrees of freedom of the intensity included in the measurement interval (see Ref. [9] for more detail). Knowing the approximate form of the probability density function of the integrated intensity, the probability density function of the number of photo counts in the arbitrary time interval can be calculated. Again using Mandel's formula and integrating gives

$$P(K) = \frac{\Gamma(K+\mathcal{M})}{\Gamma(K+1)\Gamma(\mathcal{M})}\left[1+\frac{\mathcal{M}}{\overline{K}}\right]^{-K}\left[1+\frac{\overline{K}}{\mathcal{M}}\right]^{-\mathcal{M}}, \tag{2.25}$$

where $\overline{K} = \alpha\overline{W}$. This is known as the negative binomial distribution and is a good approximation [9] to the photo count distribution considered in this research.

2.4 Tilt Estimation

Wave fronts, also called phase fronts can be described as a line along which all points have the same phase, i.e. a surface of constant optical path length (OPL). It would be fantastic if we could directly measure this wavefront, however, at the visible and infrared frequencies concerned, the phase of the light does not interact with the medium through which it travels in a manner which we can observe. Just as our eyes respond to changes in intensity, most detectors also respond to the intensity of the incident light. We need a way of using this fact to derive the phase of the wavefront from the observed change in intensity. There are many types of aberrations resulting from changes imposed on the wavefront. We are only concerned here with tilt and the measurement of it for tracking purposes. Tilt can be described as the deviation of the incident wavefront from a reference.

Figure 2.15 shows how a beam with a tilted wavefront, passing through an aperture follows a set geometry. If the beam is focussed to a spot, and the beam

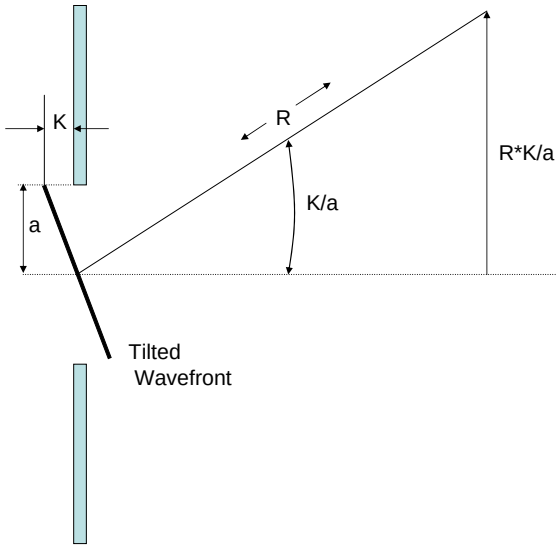


Figure 2.15: Wavefront Tilt Geometry - redrawn from [26]

has no tilt, then the spot would be focused on the optical axis. If the beam has tilt then the spot is focussed off axis. The spot position is shifted off axis by a distance proportional to the tilt angle. In actual fact, it is the weighted centroid of the focussed spot that is shifted. If a detector which responds to the position of the focussed spot is placed in the focal plane we have a way of measuring the tilt. Tilt sensors measure the OPL difference either directly in the form of an interferogram or indirectly through the differential wavefront as a function of pupil coordinates. Tilt sensors convert angular wavefront errors into intensity variations that can be sensed by a photodetector and converted to a wavefront tilt measurement by other means. The traditional tilt estimation and motion tracking problem is well-understood and adequately addressed when the object of interest is a point source or a source whose dimensions and distance from the observing aperture allow it to be safely treated as a point source. However, when the dimensions of the object of interest preclude it from being safely modeled as point source, as in many real-world scenarios, the problem is not so well-addressed. The uncooperative or extended source presents many challenges to the motion tracking and tilt estimation problem. The traditional centroiding method employed in most Shack-Hartmann type sensors relies on points of high contrast within the image to act as beacons in estimating the wavefront

tilt. When considering extended beacons or scenes which are large compared to the aperture diameter, there are often few, if any, high contrast points and the centroid algorithm often performs poorly. Additionally the centroid algorithm is sensitive to noise such as that due to the random arrival of photons to the sensor, shot noise. Cross correlation algorithms offer improved noise rejection properties as well as better performance over scenes where there are few prominent high contrast points, such as that found with some extended sources. They do have one serious drawback in that the computational burden due to the two dimensional cross correlations carried out is often heavy and renders the method slow and intensive.

The projection algorithm as derived in Ref. [5] utilizes two separate sensor arrays to sense the tilt in each of two dimensions. Optical tilt over the incoming wavefront is fed to each of the two sensors via a beam splitting mechanism and produces a shift in spot position on the array. The projection based algorithm relies on the data read out from the sensors in vector form and uses the cross-correlation technique to derive tilt in each dimension. This vector readout method is not new and has been used before by MIT Lincoln Labs in their SWAT system [1]. It does depart, however, in the method used to estimate optical tilt, the cross correlation of the vector read out.

2.5 Description of Algorithms

2.5.1 Centroid algorithm. The centroid algorithm used in this report is the traditional algorithm whereby the center of mass is calculated by dividing the sum across both x and y dimensions of the pixel values, weighted according to their displacement from the center, by the unweighted sum across both dimensions. The calculation is similar for both the x and y dimensions. The equation for the x dimension is

$$C_x = \frac{\sum_x \sum_y xi(x,y)}{\sum_x \sum_y i(x,y)} \tag{2.26}$$

where x is the position in the detector array from the origin and $i(x, y)$ is the signal at detector coordinates(x, y). As used in simulation described in Sect. 3.2, this tilt is in units of detector pixels. To enable comparison with other algorithms we convert it into rad/meter in the following manner.

If we take the PSF for no tilt to be

$$Psf(n) = \left| \sum_{k=1}^{N} A(k) \mathrm{e}^{\frac{-j2\pi nk}{N}} \right|^2 \tag{2.27}$$

where A is the amplitude of the k^{th} pixel in the array, N is the dimension of the array in pixels and n is the pixel index, then the PSF for the motion of one pixel is

$$Psf(n) = \left| \sum_{k=1}^{N} A(k) \mathrm{e}^{\frac{-j2\pi(n-1)k}{N}} \right|^2 \tag{2.28}$$

which can be separated in the exponential to give the slope of the tilt change to be $2\pi/N$ rad/aperture pixel. This is easily adapted to give the conversion from detector pixels to rad/m as $2\pi P/(ND)$ rad/m/detector pixel of motion, where D is the aperture diameter in meters and P is in aperture pixels.

The centroid algorithm is the optimal tilt estimator for Gaussian focal spots with Poisson noise [6]. However as can be seen from Eq. (2.26), the outlying pixels are weighted more than the ones closest to the center. The dimmest pixels are often found furthest from the center, which means that the least important and noisiest pixels are given a higher weighting in this algorithm which contributes to its lower performance in noisy extended scenes.

2.5.2 Projection algorithm. The key advantage to using the projection-based cross-correlation algorithm is that the two-dimensional image is reduced to a one-dimensional image, which preserves the tilt information present in one dimension of the original image. So that tilt information in both dimensions can be measured, two sensors are required. The operation of both is identical, and for the purposes of

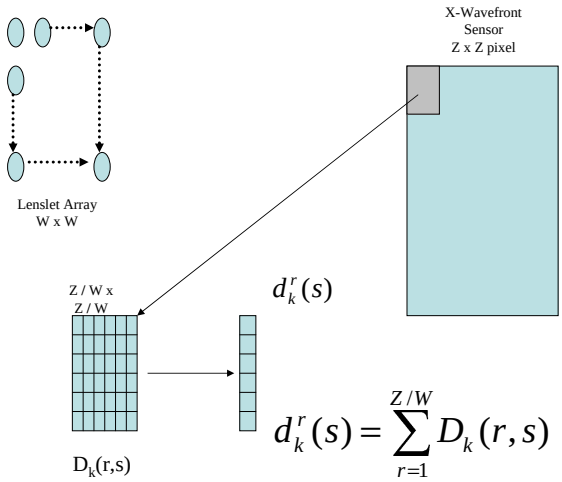


Figure 2.16: Projection transformation employed on a sensor

this report, only analysis for one sensor is shown. Referring to Figure 2.16, assuming a $W \times W$ lenslet array forming images onto a $Z \times Z$-pixel sensor array and considering just one image formed on the sensor, the data are read off in vector form such that each projection $d_k^r(s)$ is the summation of the image across the columns of the sub-array being considered as

$$d_k^r(s) = \sum_{r=1}^{Z/W} D_k(r,s) \tag{2.29}$$

where (r,s) are sensor array coordinates, $D_k(r,s)$ is image associated with the k^{th} observation, and Z and W are as previously defined.

In Ref. [6] it is shown that the maximum-likelihood (ML) estimator is a good slope estimator for wavefronts in the presence of noise. This combined with the benefits of using cross-correlation make it beneficial to use a Bayesian estimator for the tilt parameter. Here the estimator is derived by forming the likelihood function for the tilt conditional on the data and then maximizing it with respect to the tilt [27]. The likelihood function can be cast as the conditional *a posteriori* density or the conditional probability of the tilt parameter, given the measured projection and the tilt parameter from the previous image frame. It is assumed that an estimate of the true projection is known and is deterministic, and the associated tilt parameter β in

this case for vertical shift is random. The estimate is given by

$$i^r(s)w^r(s) = \sum_{r=1}^{Z/W} i(r,s)w(r,s) \tag{2.30}$$

where $i(r,s)$ is a reference frame, which may be the first frame captured, and $w^r(s)$, $w(r,s)$ are window functions of value 1 in the area of interest and zero otherwise.

In order to capture the discrete and non-negative nature of the photon counting process, a Poisson model is used for the measured projection $d_k^r(s)$ corresponding to the k^{th} frame captured. The model having a point-wise mean of $i^r(s-\beta_k)w^r(s)$ for a given shift in the vertical of β_k. As previously stated, the likelihood function requires knowledge of the distribution of the tilt parameter from frame to frame. This can be illustrated using Bayes' rule, in that

$$f_{\beta_k|d_k^r,\beta_{k-1}}(b|d,b') = \frac{f_{d_k^r|\beta_k}(d|b)f_{\beta_k|\beta_{k-1}}(b|b')}{f_{d_k^r}(d)}, \tag{2.31}$$

where $f_{\beta_k|d_k^r,\beta_{k-1}}(b|d,b')$ is the probability of the tilt for frame k conditioned on the measured projection d_k^r and the tilt from the previous frame β_{k-1} , $f_{\beta_k|\beta_{k-1}}(b|b')$ is the probability of the random tilt being equal to $\beta_k = b$, conditioned on the tilt parameter of the previous frame being $\beta_{k-1} = b'$, $f_{d_k^r|\beta_k}(d|b)$ is the probability that the projection vector random process is equal to a specific realization of that process conditioned on $\beta_k = b$ and $f_{d_k^r}(d)$ is the unconditional probability that the projection vector random process is equal to a specific realization.

To estimate the tilt parameter, Eq. (2.31) is maximized with respect to its logarithm $L(b)$ giving the maximum likelihood function

$$L(b) = \ln[f_{d_k^r|\beta_k}(d,b)] + \ln[f_{\beta_k|\beta_{k-1}}(b|b')] - \ln[f_{d_k^r}(d)], \tag{2.32}$$

which can be expressed as a function of terms that only depends on the shift b, by dropping the third term which is independent of b to give

$$L(b) = \ln[f_{d_k^r|\beta_k}(d, b)] + \ln[f_{\beta_k|\beta_{k-1}}(b|b')]. \tag{2.33}$$

In order to maximize this expression in terms of the tilt parameter b, we must have some knowledge of the probability of the current tilt parameter conditioned on the tilt from the previous frame. If this knowledge is available it should be used, however in many cases it is not and it is common practice [6] to choose a uniform density which does not vary with b and the second term of Eq. (2.33) may be dropped. In this research a uniform window centered on the previous tilt estimate is used.

The projection vector d_k^r represents an ensemble of independent Poisson random variables associated with individual pixel measurements. Given the initial assumption of statistical independence between measurements, the pdf of a collection of samples of d_k^r given $\beta_k = b$ can be expressed as a product of the marginal densities over all pixels in d_k^r, such that

$$f_{d_k^r|\beta_k}(d, b) = \mathsf{P}(d_k^r = d|\beta_k = b) = \prod_{s=1}^{Z/W} \frac{w^r(s)i^r(s-b)^{d(s)}e^{-i^r(s-b)w^r(s)}}{d(s)!}, \tag{2.34}$$

giving the log-likelihood function to be

$$L(b) = \sum_{s=1}^{Z/W} d(s) \ln[w^r(s)i^r(s-b)] - w^r(s)i^r(s-b) \tag{2.35}$$

In this research the widowing function $w^r(s)$ is chosen to be smaller than the size of the projection vector by a number of pixels equal to $2b_{\max}$, where $b_{\max}$ is the expected maximum absolute value of the tilt parameter. Then, because the window

function is defined as

$$w^r(s) = \begin{cases} 0 & \text{for } 1 \leq s \leq b_{\max}, \\ 0 & \text{for } Z/W - b_{\max} \leq s \leq Z/W, \\ 1 & \text{elsewhere,} \end{cases}$$

the natural logarithm is equal to negative infinity for values of s which make the window function equal to 0. To avoid this, the limits of integration are chosen to include only those points where the window function is non-zero, making the log-likelihood function

$$L(b) = \sum_{s=b_{\max}}^{Z/W-b_{\max}} d(s) \ln[i^r(s-b)] - i^r(s-b). \tag{2.36}$$

Equation (2.36) can be maximized with respect to tilt parameter b using an iterative approach to computing the value of the function locally around the current estimate for b and updating the estimate in the direction of increasing L in steps of Δb. The value of Δb becomes the resolution of the tilt estimation algorithm in units of array pixels. A linear interpolator is chosen to produce sub-pixel resolution for tilt estimates. The linear interpolator has the form

$$i^r(s-b) = [(1-b_f)i^r(s-b_i) + (b_f)i^r(s-(b_i-1))], \tag{2.37}$$

where b_i and b_f are the integer and fractional parts of b respectively.

In general, the wavefront sensing and tilt estimation problem has been addressed well in the case of point sources and to some extent extended beacons [20, 21]. The projection-based cross correlation algorithm has been suggested as being effective in viewing stellar and laser beacons but little has been done to evaluate its performance in the specific case of speckle noise. Research has been done on the effects of speckle on various aspects of motion estimation and imaging Ref. [2, 7, 25, 28]. This research

aims to examine that particular case where the projection algorithm is applied to the speckle noise case.

III. Simulation Structure and Models

This chapter conveys the approach and methodology used to address the research goals. First, the simulation structure is presented and discussed. Then, each conceptual model used in the simulation is explained. Lastly, the metrics developed to enable meaningful conclusions to be drawn from the results are presented and discussed.

3.1 Simulation Structure

This section describes how the simulation is built and provides a visual representation of the physical simulation structure to assist the reader in conceptualizing the simulation goals. The simulation goals are to compare and contrast the performance of projection-based cross-correlation and traditional centroiding tilt estimation algorithms in the presence of speckle noise. Figures 3.1 and 3.2 show the conceptual structure of the simulation for each of the centroiding and projection-based cases. Simply, the simulation involves a source propagated to an optical element which focusses the light onto a wavefront sensor. In both cases, the source is situated on the optical axis, a distance Z away from the focussing element, a thin lens. It is assumed that $Z \gg 2D^2/\lambda$ where D is the diameter of the pupil in meters and λ is the wavelength of light in question. This geometry was chosen to simplify the propagation problem to one of simply performing a Fourier transform to propagate the source to the lens. It was not a primary goal of this research to compare the results with and without turbulence. However, real-world scenarios would always have some degree of atmospheric turbulence present, and for this reason it is given consideration here. The turbulence in this simulation is applied at the lens surface. The turbulence applied is assumed to simulate path turbulence between the source and the lens. For simulation purposes the wavefront sensor is located at the focal plane of the lens. Again, this allowed a Fourier transform to be used to propagate from the lens to the sensor. In the case of the projection-based algorithm shown in Figure 3.2, a beam-splitter is situated between the lens and each wavefront sensor. The reasoning for splitting

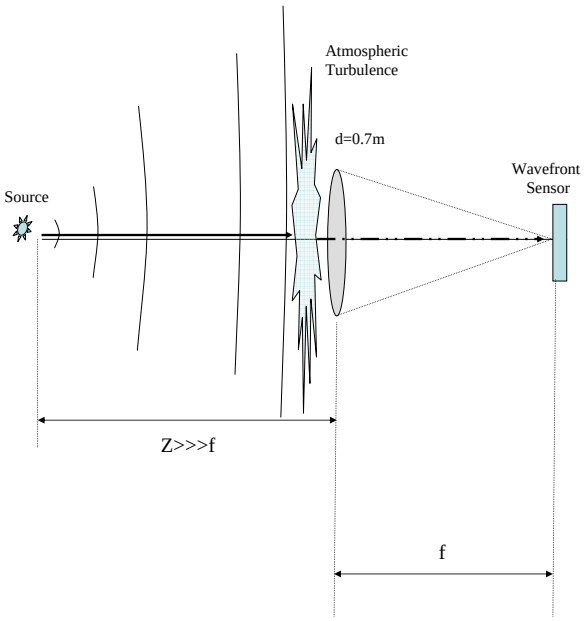


Figure 3.1: Conceptual Diagram of Simulation for the Centroiding case

Table 3.1: Simulation Parameter Space

Parameter	**Symbol**	**Units**	**Value**
Propagation grid	N	pixels	128
Aperture Dimension	P	pixels	64
Aperture Diameter	D	m	0.07
Fried Seeing Parameter	r_0	m	0.07
Intensity DOF	$\mathcal{M}$	unitless	1
Window function	w	pixels	24

the beam into separate X and Y plane components is described in Section 2.5.2. In this simulation only one sensor is simulated as the performance of each is identical. Table 3.1 lists the parameters used throughout the simulation.

3.1.1 Simulation Description. This section describes the process of the simulation. Both the projection-based and centroiding algorithms are simulated in each of the following four scenarios:

1. Extended source without turbulence,
2. Speckle source without turbulence,

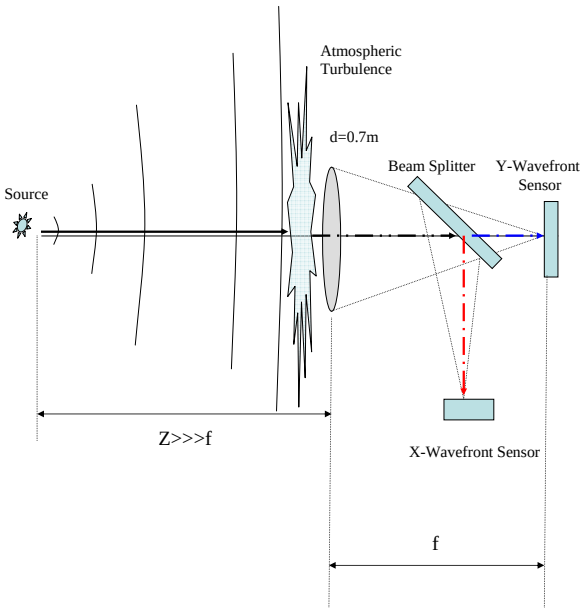


Figure 3.2: Conceptual Diagram of Simulation for the Projection-based case.

3. Extended source with turbulence, and

4. Speckle source with turbulence.

The analysis computes the rms tilt error for mean light levels of 100 to 1000 photons received, in steps of 100 photons. At each light level, 1000 realizations of the propagation are used to compute the rms tilt error. As the tilt variance can be very large, the rms error is computed to better model the tilt measured by the sensor due to pixel integration time. A separate phase screen, simulating atmospheric turbulence, is used for each realization. This increases the randomness of the simulation. Poisson-distributed random noise is added to each realization to simulate the random arrival nature of the photons, known as shot noise. A non-noisy image frame is used as a reference at each light level. For the projection-based model, the reference frame is computed from the average of 1000 realizations of the source to image plane. This corresponds the the idea that the sensor may be staring at the target for a period of time before tracking commences. The original source image is used for the centroiding algorithm, since the source is on the optical axis. Then the tilt measured by the

centroid on the source image is the *zero tilt* reference for the centroiding algorithm. Listing III.1 shows the MATLAB® code used to compute the reference image.

Listing III.1: Reference Image Code Section. (chapter3/ReferenceImageCode.m)

```
otf1 = otf_array(:,:,iloop1); % choose otf from tilt removed phase...
    screen
            if iloop1 == 1
                for iloop3 = 1:images
                    psf1 = (abs(ifft2((fft2(extended_source)).* ...
                       otf1)));
                    psf1_correct_kbar = psf1 * Kbar(iloop2); % set...
                        average rxd photon count Kbar
                    temp=ones(128,128)./(ones(128,128)+...
                       psf1_correct_kbar/2);
                    speck_image_correct_Kbar = icdf('nbin',rand...
                       (128,128),1,temp); % set average rxd photon...
                        count Kbar

                    speckle_ref_image = speckle_ref_image + (...
                       speck_image_correct_Kbar)/images;
                    proj_ref_image = proj_ref_image + (...
                       psf1_correct_kbar)/images;
                end % for iloop3
            end % if iloop1
```

In order to propagate the source through the system, firstly, the MATLAB® function *Make_otf.m*, described in Section 3.2.1 is used to compute the Optical Transfer Function (OTF) of the propagation. Then the inverse Fourier transform of the product of the source and the OTF is computed to give the PSF which is normalized for one photon. At this point the PSF is corrected for the desired mean light level (100 to 1000 photons). The speckle image is now created as described in Section 3.2.3. Shot noise is then applied using the MATLAB® function *poissrnd*. The wavefront tilt estimates using both centroid and projection-based methods are then calculated using the functions *centroid.m* and *projection_methodX.m*, which are described in Sections 3.2.5 and 3.2.6, respectively. The tilt error, defined here as the difference between the known or reference tilt and the computed tilt of the new image, is then computed for each scenario. The projection-based algorithm internally computes the tilt in rad/m. However, the centroiding algorithm does not. Therefore, the measured tilt in the centroiding case must be converted into units of rad/m before the error from both

methods is stored and used to compute the rms tilt error as described in Section 3.3. The rms tilt error for each light level is then stored for display in plot form.

3.2 Simulation Models

3.2.1 Optical Transfer Function Model. The system OTF is computed by the function *Make_otf2.m* which is shown in Listing A.2. *Make_otf2.m* works by creating an aperture function and filling it with the phase at the aperture (provided by the user). In this simulation, the phase is the turbulence-induced wavefront error produced by the phase screen generation code as described in Section 3.2.4 or in the case where turbulence is not considered, a zero phase screen. The function then performs an autocorrelation of the pupil using the *fft2* function and then normalizes the result to yield the OTF.

3.2.2 Extended Source Model. An often-used definition of an extended source is one where the source is considered extended when it can be resolved in the image plane. A considerably more in-depth description is provided in Roggemann and Welsh [21], and further references to extended sources are made in Refs. [12, 17, 19, 20, 22]. Although, none of those cited works gives clear guidance on when a source is extended and when it is not. For the purposes of this research, all that is required is for the source to be sufficiently large to adequately display the effects of speckle noise and to be resolved in the image. For this reason, the extended source was chose to be a 4×4 pixel source in a 128×128 propagation grid which, when imaged through the system, was resolved and provided an acceptable level of speckle distortion. Figures 3.3 and 3.4 show the extended source model and the image produced by the simulation. It clearly meets the objective of being resolved.

3.2.3 Speckle Noise Model. Various statistical models for use in simulating speckle noise are described in Section 2.3.3. The negative binomial model was chosen for this simulation because of its relative ease of implementation in MATLAB® . No direct function exists for generating negative binomial distributed random variables in

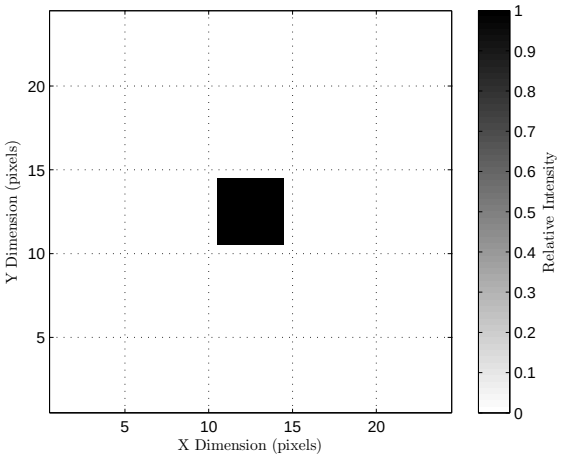


Figure 3.3: Extended source Model

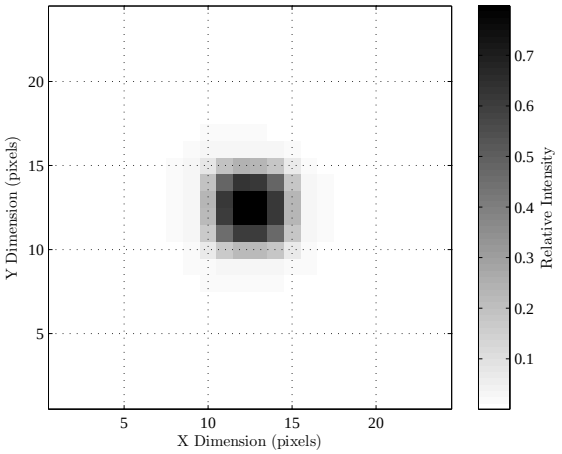


Figure 3.4: Image of extended source

MATLAB® . However, there is an Inverse Cumulative Distribution Function (*ICDF*), which, when given an appropriate probability distribution and other parameters, generates a random variable of the desired distribution and mean. The ICDF returns an array of values of the inverse cumulative distribution for a specified probability distribution, given an input data set and mean. In this simulation, the desired distribution is Negative Binomial, the data set is a random realization of 128×128 pixel image with pixel intensity from 0 to 1 and the mean is a 128×128 array in which each value is set according to

$$B = \frac{\mathcal{M}}{\mathcal{M} + \overline{K}}, \tag{3.1}$$

where B is the mean and $\mathcal{M}$ and $\overline{K}$ are as previously defined.

This method of generating random variables has been validated through numerical analysis [13]. A 128×128 pixel image (later windowed to 24×24 as described in 3.2.6) was computed by propagating the source using a series of Fourier transforms and applying an OTF generated from the OTF model from Section 3.2.1 to compute the PSF. The PSF is then corrected for the desired mean light level. A temporary array is created with the correct mean photon count and supplied to the *icdf* function to produce the speckle image with correct statistical distribution and mean light level. Listing III.2, taken from the parent file *Thesis_simulation_code_V*11, is used to implement this model. Figure 3.5 shows an example of the image of a speckle source created with the previously discussed method.

Listing III.2: Speckle Model Code section. (chapter3/Specklemodel.m)

```
psf1 = (abs(ifft2((fft2(extended_source)).* otf1)));

psf1_correct_kbar = psf1 * Kbar(iloop2); % set average rxd...
    photon count Kbar
temp=ones(128,128)./(ones(128,128)+psf1_correct_kbar/2);
speck_image_correct_Kbar = icdf('nbin',rand(128,128),1,...
   temp); % set average rxd photon count Kbar
```

3.2.4 Atmospheric Turbulence Model. Atmospheric turbulence results in variations in the refractive index along optical path. This variation is a random process, and so a model of this turbulence should be a statistical average of this

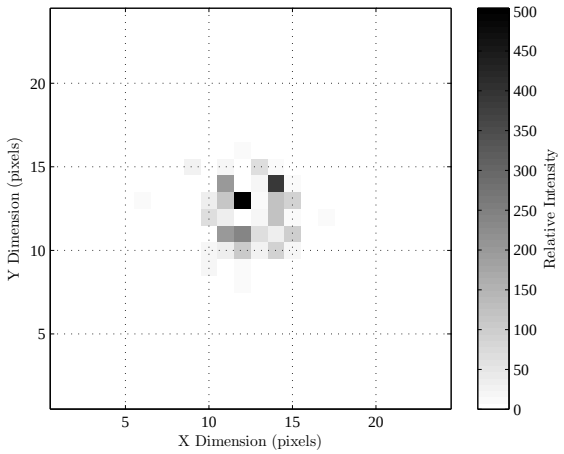


Figure 3.5: Image of speckle source

random variation. Phase screen creation involves generating single realizations of this random process. This can be achieved by transforming computer-generated random numbers into two dimensional arrays of phase values that have the same statistical distribution as the random turbulence-induced phase variations. In most cases the phase is written as a sum of basis functions. Two commonly used basis sets are Zernike polynomials and Fourier series. The Fourier series is the basis set used in this simulation. A detailed description of this method is found in Ref. [11]. Briefly, this method involves writing the optical phase $\phi(x, y)$ as a Fourier series:

$$\phi(x, y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} c_{n,m} \exp[i2\pi(f_{x_n} x + f_{y_m} y)], \tag{3.2}$$

where f_{x_n} and f_{y_m} are the x and y spatial frequencies and $c_{n,m}$ are the Fourier coefficients. Treating the phase as a two dimensional signal and making use of Parseval's theorem the Fourier series coefficients become

$$\langle |c_{n,m}|^2 \rangle = \Phi(f_{x_n}, f_{y_m}) \Delta f_{x_n} \Delta f_{y_m}, \tag{3.3}$$

where $\Phi(f_{x_n}, f_{y_m})$ is the power spectral density of the turbulence-induced phase delay and Δf_{x_n} and Δf_{y_m} are the corresponding sample spacings of the spatial frequencies. The expectation has been taken because the phase is a random process. Given

the phase variations are independent from one another in a statistical sense, the central-limit theorem can be used to determine that the coefficients are Gaussian in distribution, which obey circular Gaussian statistics of zero mean and variance given in Eq. (3.3). Knowing this, random Gaussian numbers generated via MATLAB® can be multiplied by the square root of the variance in Eq. (3.3). In this simulation the Fourier-series coefficients are generated on a uniformly sampled grid and the FFT method is used to synthesize phase screens with a good degree of computational efficiency. Listing A.7 gives the MATLAB® code used to implement this method and generate the phase screens. However as noted in Ref. [11] this method can result in poor simulation of low-order modes such as tilt. In this research it is desired to only examine the tilt induced by speckle noise, and not introduce anymore tilt than that of the speckle. Therefore this is not an issue. Furthermore, any residual tilt is removed from the phase screen by generating an array with the projected tilt across it and subtracting this from the generated phase screen. Listing III.3 from Listing A.2 shows MATLAB® code used to implement the tilt removal.

Listing III.3: Tilt Removal Code section. (chapter3/TiltRemovalCode.m)

```
%% tilt removal section
tx  = 2*pi*(-64:63)/128;
tx_matrix = ones(128,1)*tx;
wvs_x = sum(sum(tx_matrix.*phase.*aperture));
wvs_x = wvs_x/sum(sum(aperture.*tx_matrix.^2));
new_screen = phase - wvs_x*tx_matrix;
```

Figure 3.6 shows an example phase screen generated by this model.

3.2.5 Centroid Method. The centroiding algorithm described in Section 2.5.1 is implemented in MATLAB® code. The centroiding equation, Eq. (2.26) is almost coded line for line in MATLAB® . The supplied image is windowed in the same dimensions as the projection-based method uses to enable valid comparison of results. Listing A.3 shows the MATLAB® code used. The code outputs the tilt in units of pixels so this must be converted to rad/m, as described in Section 3.3, for comparison with the projection result. As in the projection-based method, only one dimension is simulated here.

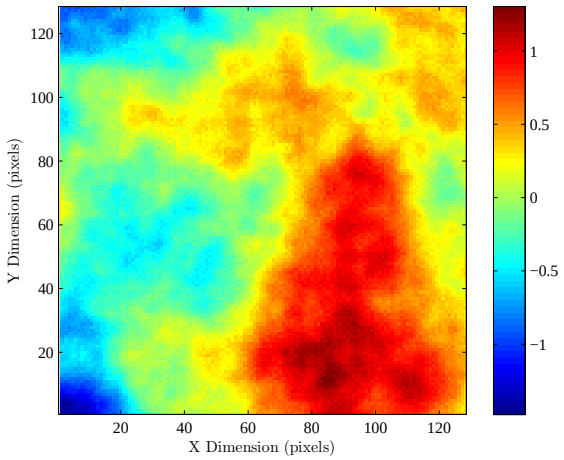


Figure 3.6: Sample phase screen created by the FFT method in this simulation.

3.2.6 Projection-based Method. There are two functions in this simulation that implement the projection algorithm as described in Section 2.5.2. They are functionally identical, differing only in the manner in which the linear interpolation step is implemented, the reasons for this are explained later in this section. The algorithm is implemented in two steps. Step one is the formation of the vector $d_k^r(s)$ as in Eq. (2.29), by summing across the sensor in the dimension desired, x or y, which would correspond to the sensor being read out in vector mode. Lines 9 to 11 of Listings A.4 and A.5 show how this is implemented. This process is carried out on both the image frame of interest and the reference image frame. Step two is the vector cross-correlation, which involves the formation of the likelihood function and the linear interpolation as described in Section 2.5.2 and Eq. (2.33), (2.34) and (2.35). The windowing of the log-likelihood function as described in Section 2.5.2 and shown in Eq. (2.36) is implemented by line 8 and the loop beginning at line 14 of listings A.4 and A.5. Note the end points and step size of the loop are identical to those of the vector initialized in line 8. These lines of code set the window size to ± 5 pixels from image center, which corresponds to the optical axis. The window size is modeled on that chosen in the proposed implementation of this algorithm in the Dunn Solar Telescope in Ref. [5]. In that case, an E2V CCD60 array with pixels of $24\mu m$ pitch, and a sub array size of 24×24 pixels was used, yielding a field of view

of 1/3 of an arc second. The difference in the two functions came from experience gained during the simulation evaluation. The difference lies in the method used to derive sub-pixel accuracy in the linear interpolation. Both methods achieve the same accuracy using different approaches. *ProjectionMethod2* separates the interpolation into an integer and fractional component and interpolates from there. This achieved good results with the speckle case, and poorer results with the non-speckle case. Through research it was found that inherent sub-pixel accuracy of the *makeshift_vec* function achieved better results with the non-speckle case and thus two versions are used.

3.3 Simulation Metrics

In order to make proper comparison of results obtained, the rms tilt error due to noise in units of radians per meter was calculated for all simulations. As previously mentioned the projection-based method inherently computes the tilt error in units of rad/m. By examining the centroiding equation, Eq. (2.26), it can be seen that the tilt computed by the centroiding method is in units of detector pixels. To enable comparison with other algorithms we convert it into rad/m in the following manner. If we take the point spread function (PSF) for no tilt to be as shown in (2.27) then the PSF for motion of one pixel is

$$PSF(n-1) = \left| \sum_{k=1}^{N} A(k) \mathrm{e}^{\frac{-j2\pi(n-1)k}{N}} \right|^2, \tag{3.4}$$

which can be separated in the exponential to give the slope of the tilt change to be $2\pi/N$ rad/aperture pixel. This is easily adapted to give the conversion from detector pixels to rad/m as $2\pi P/(ND)$ rad/m/Detector pixel of motion, where D is the aperture diameter in meters and P is in aperture pixels.

The rms tilt error σ_n is computed by

$$\sigma_n = \sqrt{\frac{\sum_{n=1}^{N_i} \Delta_n^2}{N_i}}, \tag{3.5}$$

where N_i is the number of realizations made at each mean light level and Δ is the computed tilt error for a given realization.

IV. Simulation Results

This chapter presents the results of the simulation described in Chapter III and presents some discussion in the context of the original research goals.

4.1 Simulation using Extended Source No Speckle

The research done by Cain in Ref. [5] and Cain, Hyatt, and Armstrong in Ref. [4] established a baseline premise for the results of this research, in that the projection-based cross-correlation algorithm exhibited improved performance when viewing stellar beacons as well as extended objects and scenes. The first goal of this research was to establish the performance of the projection-based algorithm when considering extended objects. Figure 4.3 shows the computed tilt error for both the projection-based and traditional centroiding algorithms in plot (a) the case with no turbulence considered and plot (b) with turbulence considered. In Figure 4.3(a) it can be seen that, as expected, the projection algorithm performs better than the centroiding algorithm at all mean light levels simulated. Although the difference is not significant, it is measurable. In Figure 4.3(b) it can be seen that, again the projection method exhibits better performance, in this case by an increased margin. Whilst the computed error has increased for both methods, the increase in error produced by the centroid method was much larger and tends to stabilize as the light level increases while the project method continues to improve.

4.2 Simulation using Extended Source with Speckle

Figure 4.2 shows the computed tilt error for both the projection-based and traditional centroiding algorithms in the presence of speckle noise for plot (a) the case with no turbulence considered and plot (b) turbulence considered. Again, the projection method performs better than the centroid, although with a much less observable difference. This indicates that the speckle noise is the dominant effect, and that the turbulence may even exacerbate the effect of speckle.

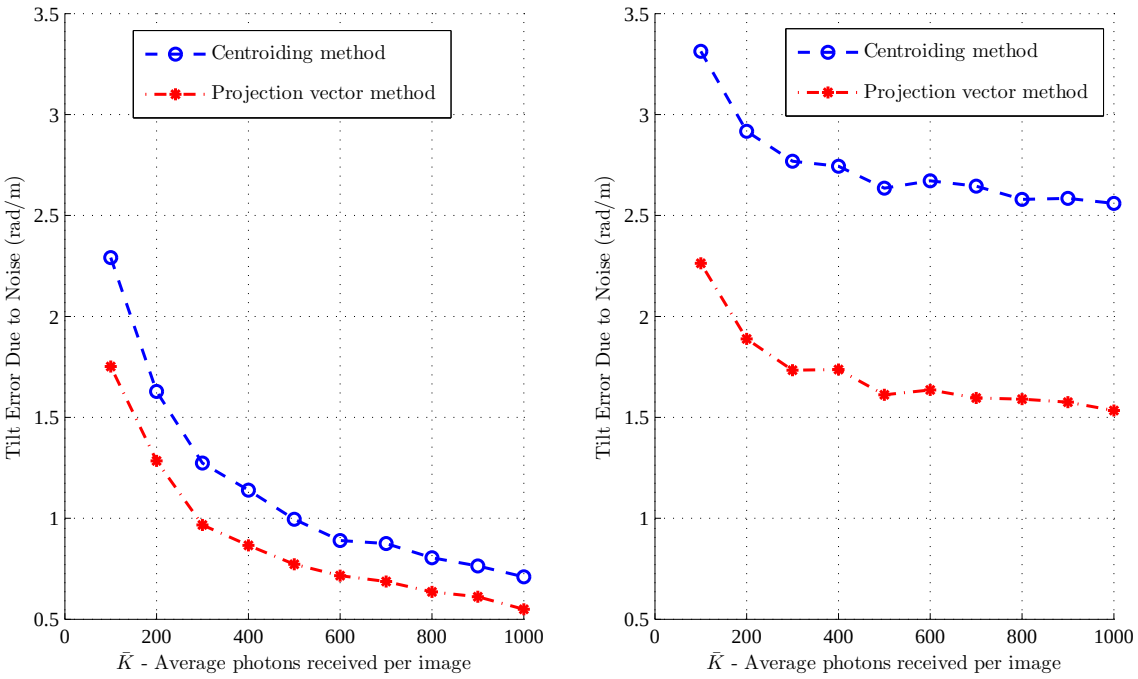
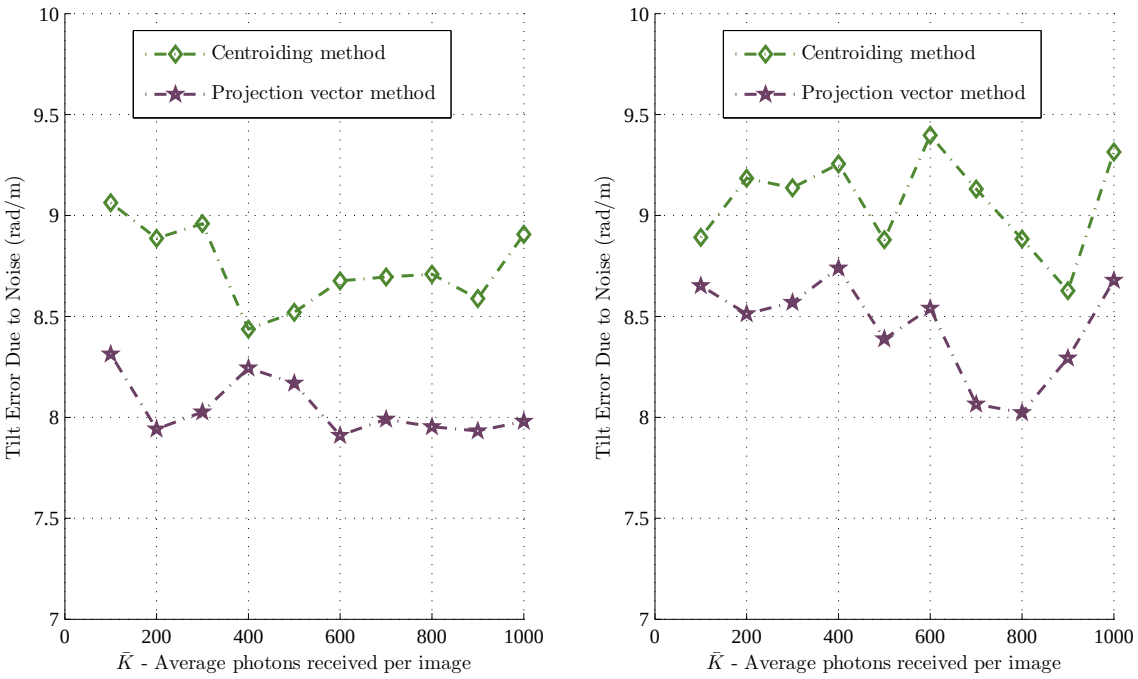


Figure 4.1: (a) Tilt error for extended source with no turbulence.
(b) Tilt error for extended source with turbulence.



(a) (b)

Figure 4.2: (a) Tilt error for speckle with no turbulence.
(b) Tilt error for speckle with turbulence.

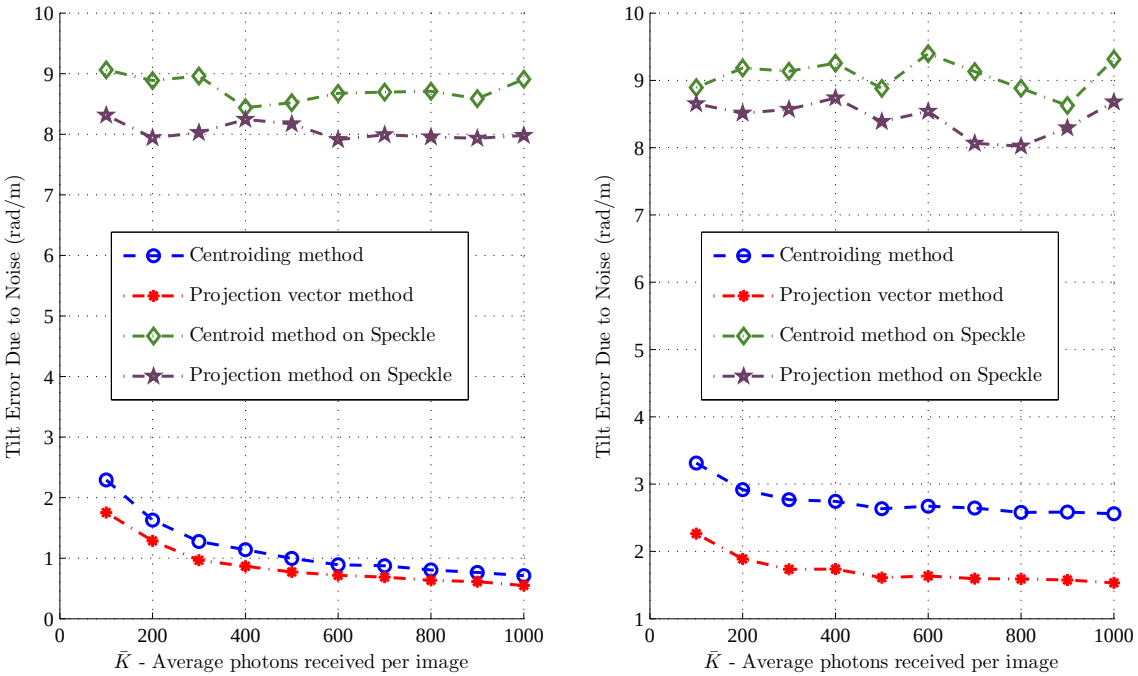


Figure 4.3: (a) No turbulence considered tilt error.
(b) Turbulence considered tilt error.

4.3 Effect of Atmospheric Turbulence and Speckle

When considering all results, it can be seen that once speckle is introduced the turbulence effect is minimal by comparison. The projection-based algorithm consistently performs better than the traditional centroiding algorithm in all cases. It is interesting to note however, that the increasing light levels did not appear to favor either method when speckle was considered.

V. Conclusions and Further Work

This chapter presents a summary of the research and key results from Chapter IV. Further, it addresses the main challenges faced during this research and how well the initial goals were satisfied. Recommendations of further work that could begin with and extend on these results is also discussed.

5.1 Summary

The objective of this research was to examine the performance of a projection-based cross correlation algorithm as developed by Cain in Ref. [5] in the presence of speckle noise via simulation. Three areas of research made up the bulk of this work. They were:

1. Extended Beacon modeling,
2. Speckle noise modeling, and
3. Implementation of MLE and the projection-based algorithm.

It was necessary to have all three areas functioning correctly to produce results from which valid conclusions could be drawn. As discussed in Chapter III, consideration was also given to effect of atmospheric turbulence on the performance of the algorithm.

The research began with an examination of the performance of the projection algorithm with extended beacons. This required the development of a suitable model for use in simulation. Much work [22,28,29], has been done on the theory of extended versus point sources, but little has been done on mathematically modeling them. This presented a challenge in this research. It was eventually decided that any source that could be resolved in the final image would be sufficient to model an extended source for the purposes of this research. The performance of the projection algorithm was examined in simulation against the performance of the traditional centroiding algorithm and was shown to offer measurable improved performance over the centroiding

method in both the case where turbulence is not considered and when it was considered. In particular the performance improvement margin was significant in the case where turbulence was considered. This could be attributed to the ability of the projection based algorithm to reject the low spatial frequency changes in the image introduced by turbulence. Also noteworthy in the case where turbulence is considered, is the reducing error trend as the mean light level increases for the projection method. This contrasts with result for the centroiding method, which is trending to level out.

Considerable work had been done previously on both characterizing the statistical nature of the speckle phenomena [2, 8–10], and on the use of MLE for motion estimation [3, 7, 25]. Little work had been done on combining these experimentally in simulation and examining the results. Several statistical models for speckle phenomena have been put forward and in some cases [9, 10] validated by mathematical evaluation. These models all had limiting cases and applications and remained unclear as to whether there was an overall model that, while not characterizing all aspects of speckle noise completely, would be acceptable for widespread use. Two models in particular were evaluated for their ability to model speckle in the context of this research, that is, the case of tracking. The negative exponential distribution for photon count, as discussed in Section 2.3 with later addition of Poisson distributed shot noise, was initially explored. However this method produced images, Figure 5.1, after propagation that were no longer acceptable as speckle images because they did not display the known properties of speckle noise. The negative binomial distribution as detailed in Section 2.3 and Ref. [9], which includes both speckle effect and shot noise was found to generate images with the correct speckle features as shown in Figure 3.5. Using this negative binomial distribution reduced the computational load in computing the final image to present to the projection algorithm. It was shown that the projection algorithm again consistently performed better than the traditional centroiding algorithm in both the case of with turbulence and without turbulence. It is noteworthy again that the addition of turbulence had little effect on the overall

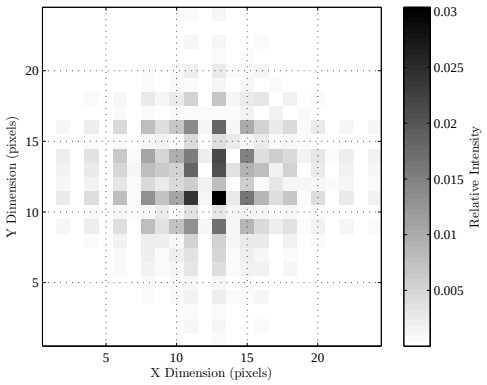


Figure 5.1: Initial Speckle Model Image

performance of either algorithm when speckle was present. Turbulence did not seem to affect the average performance of either algorithm measurably, however it did produce a pronounced increase in the variation in computed error. Figure 5.2 shows the results obtained with the overall mean error for each case. This would indicate that speckle is the dominant effect, which is not surprising as the statistical pdf used for the MLE was Poisson and does not account for the Bose-Einstein nature of speckle noise. The reason for the increase in error at a mean light level of > 800 photons is unclear and may in fact be a statistical anomaly which would be resolved by more realizations at each light level with the added burden of increase computation time or may be averaged out with an increase in the maximum mean light level simulated to greater than 1000 photons. Additional research may be carried out to determine if there are limits (high or low mean photon counts), outside of which the existing algorithm no longer provides a good approximation. Overall the projection-based cross-correlation algorithm has been shown to have increased performance over the traditional centroiding algorithm in the presence of speckle noise.

5.2 Key Results

This section summarizes the key conclusions from the simulation results in Chapter IV.

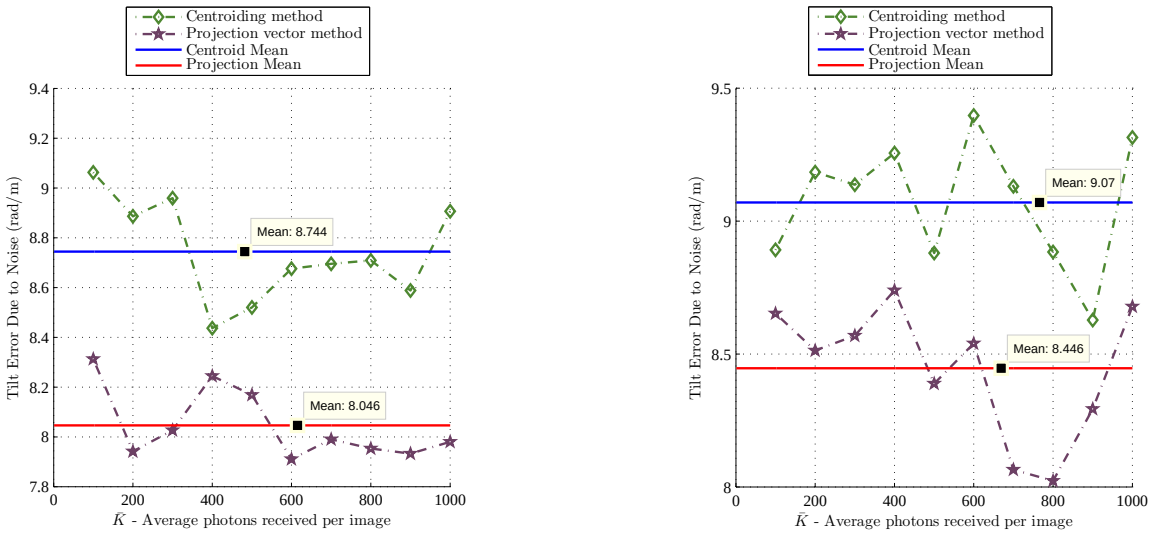


Figure 5.2: (a) Computed mean tilt error no turbulence.
(b) Computed mean tilt error with turbulence.

- The projection-based cross-correlation algorithm showed improved performance in tilt estimation over the traditional centroiding method when considering extended beacons even when atmospheric turbulence is considered and the correct (Bose-Einstein) pdf was not used.
- The projection-based cross-correlation algorithm showed improved performance in tilt estimation over the traditional centroiding method in the presence of speckle noise. Different models for speckle noise, while statistically correct, may not accurately model the phenomena in the case being considered.
- While the projection algorithm did outperform the centroid algorithm in all cases, the difference in performance was not always significant. With this in mind, and taking into account the computational burden of the projection-based algorithm, in some cases depending on the conditions the centroiding algorithm may be all that is required.

5.3 Recommendations

This section provides recommendations for future work based on this research. *Develop Bose-Einstein MLE Algorithm.* Initially a goal of this research, the MLE algorithm could be developed using the Bose-Einstein statistical distribution for photon count in speckle images. While this research has shown that the projection-based method has improved performance over the traditional centroiding method using the Poisson distribution, it is not the true distribution and the algorithm may show greater improvement if the correct statistical model where used in the MLE.

Continuation of Speckle Model. More work could be done characterizing the speckle noise model so it could be propagated in the same manner as the extended source. Then the algorithms could be further tested and the models merely swapped in and out of the simulation. The speckle model could also be verified against some lab tests with hardware.

Real-World Data. The existing simulation could be run again with some real-world image data and the results compared to the existing simulation data. Real-world data could be gathered from surfaces of varying degrees of optical roughness and the performance of the algorithm examined for each surface.

Appendix A. Matlab Code

This appendix contains all the MATLAB® code used in the thesis.

A.1 Parent Code

Listing A.1: The parent file from which all other functions are called. (appendix1/ParentCodeV11.m)

```
%% Thesis_simulation_code_V11.m
% Written by FLTLT Brett Monz
% 2008
%...
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%...

% Version Development notes:
% V1 - Extended source modelling
% V2 - Added centroiding algorithm and plot or noise vaiance
% V3 - corrected error in D aperture diameter now 0.07m
%    - added SRI eqn 5.38 Roggemann
% V4 - changed beta =1, s = 0.5 changed code to only claculate ...
    psf1 once
% V4A- removed taking real part of fft of extended source
% V5 - added projection vector method code
% V6 - added eqn 5:30 from Roggemann
% V6aug21 - corrected code such that projection now performs ...
    better than
% centroid with extended source
% V8 - Thesis version
% Changed name to Thesis_simulation_code
% V1 - implemented centroid code as function call
% V1a  - implemented projection vector method as function call
% V2 - implementing speckle image processing with centroiding  NO
%      turbulence consideration implemented for speckle
% V3 - implementing speckle image processing for projection  NO
%      turbulence consideration implemented for speckle
% V3A- Initial Image plots formatted for paper/thesis
% V4 - Turbulence Included, now uses speckle_gen2.m function and ...
    save_plots
%      variable to allow sving to .eps or not
% V4A- test to change dimension of extended source
% V5 - added set colour values to specify plot colours for graphs
% V6 -  zero tilt phase screens
% V7 - individual phase screen for each realisation
% V8 - use average of 100 images for reference for projection ...
    method, also
% requires projection_method2.m
% V10 - changed speckle simulation in non turbulent case
% V11 - changed speckle simulation in turbulent case
%...
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%...
```

```
clear all;
close all; clc
check_image = true; % set if image display req
%check_image = false; % set if image display not req
%turbulence_included = true; % set to true if turbulence ...
    considered and will load phase screen and use correct plot text
turbulence_included = false; % set to false if turbulence not ...
    considered and
%will use correct plot text and not load phase screen
%save_plots = true; %uncomment to save plots to .eps files
save_plots = false; % uncomment to not save plots to .eps files
darkgreen = [78 136 49]./255; % set colour dark green
maroon = [153 0 102]./255; % set colour maroon
%% initialising
if turbulence_included
load Simdata_thesis_1000_tilt_rem %Simdata_thesis%Simdata
end %if turbulence included load individual otfs peviously created

N = 128; %number of aperture pixels
D = .07; %aperture diameter in m
P = 64; %pixels in aperture
ro = 0.07; % fried seeing parameter
ua = .75; % atmospheric visibility factor
beta = .5; % angular size of beacon relative to seeing limited ...
    angle
s = 0.5 ;% normalised shear
speckle_source_dim = 4; %dimension of source for speckle image
images = 1000; % number of images per light level
Kbar = (100:100:1000); % set array of kbar values
speckle_ref_image = zeros(128);
proj_ref_image = zeros(128);
phase_slope_centroid_speckle = zeros(1,length(Kbar)); %create ...
    array for phase slope values
phase_slope_centroid_noise = zeros(1,length(Kbar)); %create array ...
    for phase slope values

sigma_n = zeros(1,length(Kbar)); %create array for tilt error ...
    values

extended_source = zeros(128,128); %
extended_source(64:67,64:67) = ones(4,4);%
Likelihood = zeros(1,10);
sigma_n_projection = zeros(1,length(Kbar));
sigma_n_centroid_speckle = zeros(1,length(Kbar));
sigma_n_projection_speckle = zeros(1,length(Kbar));
phase_slope_projection = zeros(1,100);
phase_slope_projection_speckle = zeros(1,100);


otf_no_turb = Make_otf(32,0,128,1,zeros(128,128)); %make otf ...
    without atmospheric turbulence screen
```

```
if turbulence_included
    otf1 = otf_array(:,:,1); % choose otf from tilt removed phase ...
       screen
else
    otf1 = otf_no_turb;
end %if turbulence included load phase screen data

% if check_image debug flag set display images
if check_image
    psf1 = (abs(ifft2((fft2(extended_source)).* otf1)));
    psf1_correct_kbar = psf1 * Kbar(5); % set average rxd photon ...
       count Kbar
    temp=ones(128,128)./(ones(128,128)+psf1_correct_kbar/2);
    speck_image_correct_Kbar = icdf('nbin',rand(128,128),1,temp); ...
       % set average rxd photon count Kbar

%  speck_image = speckle_gen2(speckle_source_dim,...
   turbulence_included,otf1); %generate speckle image
%  %psf1a = fftshift(abs(ifft2(otf1)));
%  psf1 = (abs(ifft2((fft2(extended_source)).* otf1)));

set(0, 'defaulttextinterpreter', 'latex');
f =figure();
ax2 = axes('Units', 'Inches', 'OuterPosition', [0 1 5 5]);
imagesc(psf1(54:77,54:77));colorbar
 colormap('gray');
 cmap = colormap('gray');
 colormap(flipud(cmap)); % use inverse colormap for printing
 grid
 xlabel([' X Dimension (pixels) ' ]);
ylabel(['Y Dimension (pixels)']);

 f=figure();
 ax3 = axes('Units', 'Inches', 'OuterPosition', [0 1 5 5]);
 imagesc(speck_image_correct_Kbar(54:77,54:77));colorbar
 colormap('gray');
 cmap = colormap('gray');
 colormap(flipud(cmap)); % use inverse colormap for printing
 grid
  xlabel([' X Dimension (pixels) ' ]);
ylabel(['Y Dimension (pixels)']);
end % if check_image

%% Make Images at "images" light levels

for iloop2 = 1:length(Kbar) % loop for light levels

    for iloop1 = 1:images % loop through images per light level
iloop1
pause(.1)
        if turbulence_included
```

```
129             otf1 = otf_array(:,:,iloop1); % choose otf from tilt ...
                    removed phase screen
                if iloop1 == 1
                    for iloop3 = 1:images
                        psf1 = (abs(ifft2((fft2(extended_source)).* ...
                            otf1)));
                        psf1_correct_kbar = psf1 * Kbar(iloop2); % set...
                            average rxd photon count Kbar
134                     temp=ones(128,128)./(ones(128,128)+...
                            psf1_correct_kbar/2);
                        speck_image_correct_Kbar = icdf('nbin',rand...
                            (128,128),1,temp); % set average rxd photon...
                            count Kbar

                        speckle_ref_image = speckle_ref_image + (...
                            speck_image_correct_Kbar)/images;
                        proj_ref_image = proj_ref_image + (...
                            psf1_correct_kbar)/images;
139                 end % for iloop3
                end % if iloop1

            else
                otf1 = otf_no_turb;
144             proj_ref_image=psf1*Kbar(iloop2);
                speckle_ref_image=psf1*Kbar(iloop2);
            end %if turbulence included load phase screen data
            %psf1a = fftshift(abs(ifft2(otf1)));
            psf1 = (abs(ifft2((fft2(extended_source)).* otf1)));
149
            psf1_correct_kbar = psf1 * Kbar(iloop2); % set average rxd...
                photon count Kbar
            temp=ones(128,128)./(ones(128,128)+psf1_correct_kbar/2);
            speck_image_correct_Kbar = icdf('nbin',rand(128,128),1,...
                temp); % set average rxd photon count Kbar
            psf_noise_less = psf1_correct_kbar;
154         Cx_no_noise = centroid(extended_source);% Cx_no_noise = ...
                centroid(psf1_correct_kbar);
            Cx_no_noise_speckle = centroid(extended_source);% ...
                Cx_no_noise_speckle = centroid(speck_image_correct_Kbar)...
                ;

            psf1_correct_kbar_noisy = poissrnd(psf1_correct_kbar); % ...
                make noisy
            temp=ones(128,128)./(ones(128,128)+psf1_correct_kbar);
159         speck_image_correct_Kbar_noisy= icdf('nbin',rand(128,128)...
                ,1,temp); ; % make noisy

            %calculate centroid noisy for extend source
            Cx_noise = centroid(psf1_correct_kbar_noisy);
            phase_slope_centroid_noise(iloop1)= (Cx_noise-Cx_no_noise)...
                *2*pi*P/(N*D); % in rads/m
164
```

```
            %calculate centroid noisy for speckle source
            Cx_no_noise_speckle = centroid(speck_image_correct_Kbar);
            Cx_noise_speckle = centroid(speck_image_correct_Kbar_noisy...
               );
169         phase_slope_centroid_speckle(iloop1)= (Cx_noise_speckle - ...
               Cx_no_noise)*2*pi*P/(N*D); % in rads/m


            % calculate phase slope using projection method
            phase_slope_projection(iloop1)= projection_method3(...
               psf1_correct_kbar_noisy,proj_ref_image,P,N,D);% in rads...
               /m
174         phase_slope_projection_speckle(iloop1)= projection_method2...
               (speck_image_correct_Kbar,speckle_ref_image,P,N,D);% in...
                rads/m

        end %iloop1
        sigma_n(iloop2) = sqrt(sum((phase_slope_centroid_noise).^2)./...
           images);
        sigma_n_projection(iloop2) = sqrt(sum((phase_slope_projection)...
           .^2)./images);
179     sigma_n_centroid_speckle(iloop2) = sqrt(sum((...
           phase_slope_centroid_speckle).^2)./images);
        sigma_n_projection_speckle(iloop2) = sqrt(sum((...
           phase_slope_projection_speckle).^2)./images);

    iloop2

184 end % iloop2

    %% plot figure
    set(0, 'defaulttextinterpreter', 'latex');
    f =figure();
189 ax1 = axes('Units', 'Inches', 'OuterPosition', [0 1 5 5]);
    plot(ax1,Kbar,sigma_n,'b--o');
    hold on; grid on;

    plot(ax1,Kbar,sigma_n_projection,'r-.*');
194 plot(ax1,Kbar,sigma_n_centroid_speckle,'-.d','color',darkgreen);
    plot(ax1,Kbar,sigma_n_projection_speckle,'-.p','color',maroon);
    xlabel([' $\bar{K} $  - Average photons received per image ' ]);
    ylabel(['Tilt Error Due to Noise (rad/m)']);

199 l2 = legend(['Centroiding method'],['Projection vector method'],['...
       Centroid method on Speckle'],['Projection method on Speckle']);
    set(l2, 'location', 'northeast','Fontsize',10 );

    if save_plots
        if turbulence_included
204         print('-depsc', 'Research_tilt_error_turbulence');
        else
```

```
            print('-depsc', 'Research_tilt_error_no_turbulence');
        end %if  turbulence_included)
    end % if save_plots
209
    %save('Thesis_sim_data_no_turbulence_1000_4x4_ext_source_testing1...
        ', 'Kbar', 'sigma_n','sigma_n_projection','...
        sigma_n_centroid_speckle','sigma_n_projection_speckle')
```

A.2 Make Otf function

Listing A.2: This function computes the Optical transfer Function from the source to the lens
(appendix1/Makeotf2.m)

```
function [otf,aperture] = Make_otf2(r1,r2,si,scale,phase)
% [otf,apeture] = make_otf(r1,r2,si,scale,phase);

mi = floor(si/2);
mi=mi+1;
aperture = zeros(si,si);
for i = 1:si

    for j = 1:si

        dist = sqrt((i-mi)^2+(j-mi)^2);
        if(dist<=r1)

            if(dist>=r2)
                aperture(i,j) = 1;

            end
        end

    end

end
%% tilt removal section
tx  = 2*pi*(-64:63)/128;
tx_matrix = ones(128,1)*tx;
wvs_x = sum(sum(tx_matrix.*phase.*aperture));
wvs_x = wvs_x/sum(sum(aperture.*tx_matrix.^2));
new_screen = phase - wvs_x*tx_matrix;
%%
pupil = aperture.*cos(new_screen) + sqrt(-1)*aperture.*sin(...
   new_screen);
psf = real(fft2(pupil).*conj(fft2(pupil)));

psf = scale*psf/sum(sum(psf));

otf = fft2(psf);
```

A.3 Centroiding Algorithm

Listing A.3: This function computes the centroid of the image sent and returns the wavefront tilt.
(appendix1/centroid.m)

```
function[cx]=centroid(image)
% returns centroid (cx) in x dimension only of (image)
% image should have dim 128x128  (not checked)
[posx,posy] = meshgrid(-12:1:11); % create position array each ...
    element value is its delta from center in pixels
cx=sum(sum(posx.*image(54:77,54:77)))./sum(sum(image(54:77,54:77))...
    );
```

A.4 Projection Method Algorithm 2

Listing A.4: This function computes the wavefront tilt using a MLE algorithm with fractional and integer pixel resoultion.
(appendix1/projectionmethod2.m)

```
function[phase_slope]=projection_method2(image,ref_image,P,N,D)
% returns phase slope of (image)
% N number of aperture pixels
% D aperture diameter in m
% P pixels in aperture
% image should have dim 128x128 (not checked)
% requires makeshift_vec.m to reside in same directory
bvec=-5:.01:5; % vector of pixel locations for linear ...
   interpolation
psf11=(image(54:77,54:77)/2);
Px = sum(psf11);
Po = sum(ref_image(54:77,54:77)/2);
bindex = 1;
% do linear interpolation
for bloop  = -5:.01:5
    bi = floor(bloop);
    bf = bloop-bi;
    Pso = makeshift_vec(Po,bi);
    Ps1 = makeshift_vec(Po,bi+1);
    Pos = (1-bf)*Pso + bf*Ps1;

    % do likelihood
    Likelihood(bindex) = sum(Px.*log(Pos)- Pos);
    bindex = bindex+1 ;   % update bindex
end%bloop
maxLikelihood = max(Likelihood);
dummyval = find(Likelihood == maxLikelihood);
phase_slope= bvec(dummyval)*2*pi*P/(N*D) ;% in rads/m
```

A.5 Projection Method Algorithm 3

Listing A.5: This function computes the wavefront tilt using a MLE algorithm . (appendix1/projectionmethod3.m)

```
function[phase_slope]=projection_method3(image,ref_image,P,N,D)
% returns phase slope of (image)
% N number of aperture pixels
% D aperture diameter in m
% P pixels in aperture
% image should have dim 128x128 (not checked)
% requires makeshift_vec.m to reside in same directory
bvec=-5:.01:5; % vector of pixel locations for linear ...
    interpolation
psf11=(image(54:77,54:77)/2);
Px = sum(psf11);
Po = sum(ref_image(54:77,54:77)/2);
bindex = 1;
% do linear interpolation
for bloop  = -5:.01:5
    % bi = floor(bloop);
    % bf = bloop-bi;
    % Pso = makeshift_vec(Po,bi);
    % Ps1 = makeshift_vec(Po,bi+1);
     Pos = makeshift_vec(Po,bloop);

     % do likelihood
     Likelihood(bindex) = sum(Px.*log(Pos)- Pos);
     bindex = bindex+1 ;   % update bindex
end%bloop
maxLikelihood = max(Likelihood);
dummyval = find(Likelihood == maxLikelihood);
phase_slope= bvec(dummyval)*2*pi*P/(N*D) ;% in rads/m
```

A.6 Vector Linear Shift Algorithm

Listing A.6: This function implements a linear shift of points in a supplied vector. (appendix1/makeshiftvec.m)

```
function [img2]= makeshift_vec(img1,dx)
% function [img2]= makeshift(img1,dx,dy)
% dy and dx are the shifts in the vertical and horizontal ...
    directions respectively
% img1 and img2 are the two images from a sequence of video
% delta is the denominator of the fraction of a pixel to which the...
    estimation is to be done
% ex 1/10 pixel estimation means delta =10

sz=size(img1);
sz=max(max(sz));
center = [floor(sz/(2))+1];
                linx = -center+1:1:center-2;
                linx = -2*pi*linx/sz;
      linx = fftshift(linx);


px = cos(linx*dx)+sqrt(-1)*sin(linx*dx);

img2 = real(ifft(fft(img1).*(px)));
```

A.7 Atmospheric Turbulence Simulation

Listing A.7: This code produces tilt removed phase screen realizations. (appendix1/PhaseScreen.m)

```
function [screen,deltrho]=PHASE_SCREEN(D,ro,lam1,si,K)
% [screen]=PHASE_SCREEN(D,ro,si,K);
%D is the linear dimension in meters corresponding to a square ...
   with si pixels on a
%side. Example:  You have a wfs aperture of 10 cm and are putting ...
   it
% in an array 32 pixels on a size.  The array is twice the size of...
    the aperture.
% then D=.1 and si = 32.  ro is Fried's seeing parameter in meters...
   .
% K is the number of independent phase screens you need to ...
   generate.
%on output make sure deltrho is not larger than about 4 ...
   millimeters

si1=1024;
si2=1024;
deltrho=D/si;
Lo=deltrho*si1/2
ac=zeros(si1,si2);
mi1=si1/2+1;
mi2=si2/2+1;
for i = 1:si1
    i ;
    for j = 1:si2
        rho = deltrho*sqrt((i-mi1)^2+(j-mi2)^2);
        ac(i,j)=besselk((5/6),(2*pi*(rho)/Lo))*((rho)^(5/6))*(Lo...
           /(2*pi))^(5/6);
        ac(i,j) =ac(i,j)/((2^(5/6))*gamma(11/6));
        if(rho==0)
            ac(i,j)=0;
        end
    end
end
sz=1024;
Cn2dz = ((.185)^(5/3))*(lam1^2)/(ro^(5/3))*10^(-12);
k = 2*pi/(lam1*10^(-6));
ac1=ac*.033*(4*pi^2)*k*k*Cn2dz;
ac1=fftshift(ac1);
ac1(1,1)=ac1(1,2);
ft1 = sqrt(abs(fft2(ac1)));
for k=1:K
    phase = randn(sz,sz);
    phase1 = real(ifft2(fft2(phase).*ft1));
    screen(:,:,k)=phase1(1:si,1:si);
    k
    pause(.1)
end;return
```

Bibliography

1. Barclay, H. T., P. H. Malyak, W. H. McGonagle, R. K. Reich, G. S. Rowe, and J. C. Twichell. "The SWAT wavefront sensor". *Lincoln Laboratory Journal*, 5:115–130, 1992.

2. Baribeau, Réjean and Marc Rioux. "Influence of speckle on laser range finders". *Appl. Opt.*, 30(20):2873–2878, 1991. URL `http://ao.osa.org/abstract.cfm?URI=ao-30-20-2873`.

3. Barrett, Harrison H., Christopher Dainty, and David Lara. "Maximum-likelihood methods in wavefront sensing: stochastic models and likelihood functions". *J. Opt. Soc. Am. A*, 24(2):391–414, 2007. URL `http://josaa.osa.org/abstract.cfm?URI=josaa-24-2-391`.

4. Cain, S.C., M.M. Hyatt, and E.A. Armstrong. "Projection Based Image Registration in the presence of fixed pattern noise". volume 10 of *IEEE Transactions on Image Processing*, 1860–1872. Dec. 2001.

5. Cain, Stephen. "Design of an image projection correlating wavefront sensor for adaptive optics". *Optical Engineering*, 43(7):1670–1681, 2004. URL `http://link.aip.org/link/?JOE/43/1670/1`.

6. van Dam, Marcos A. and Richard G. Lane. "Wave-front slope estimation". *J. Opt. Soc. Am. A*, 17(7):1319–1324, 2000. URL `http://josaa.osa.org/abstract.cfm?URI=josaa-17-7-1319`.

7. Duncan, Donald and Sean Kirkpatrick. "Performance analysis of a maximum-likelihood speckle motion estimator". *Opt. Express*, 10(18):927–941, 2002. URL `http://www.opticsexpress.org/abstract.cfm?URI=oe-10-18-927`.

8. Goodman, J. W. "Some fundamental properties of speckle". *J. Opt. Soc. Am.*, 66(11):1145–1150, 1976. URL `http://www.opticsinfobase.org/abstract.cfm?URI=josa-66-11-1145`.

9. Goodman, Joseph W. *StatisticalOptics*. John Wiley and Sons, 2000.

10. Goodman, Joseph W. *Speckle Phenomena in Optics - Theory and Applications*. Roberts and Company, 2006.

11. Jason Schmidt. "OENG 645 Wave Optics Book Notes". Class Text, 2008.

12. Katz, Milton. *Introduction to Geometrical objects*. World Scientific, 2002.

13. Knuth, Donald. *The Art of Computer Programming*. Addison-Wesley, 1969.

14. von Laue, M. *Sitzungsber Akad Wiss*, 1914.

15. von Laue, M. *Mitt Physik Ges*, 1916.

16. von Laue, M. *Verhandl Deut Ohys Ges*, 1917.

17. M. Freeman, W.N Charman, Christopher Hull. *Optics*. Elseivier Health Sciences, 2004.

18. Mandell, L. *Proc. Phys. Soc.*, 1959.

19. O'Shea, Donald C. *Elements of Modern Optical Design*. Wiley, 1985. ISBN 9780471077961.

20. Poyneer, Lisa A. "Scene-Based Shack-Hartmann Wave-Front Sensing: Analysis and Simulation". *Appl. Opt.*, 42(29):5807–5815, 2003. URL `http://ao.osa.org/abstract.cfm?URI=ao-42-29-5807`.

21. Roggemann, Michael C. "Fundamental considerations for wavefront sensing with extended random beacons". Michael T. Valley and Mikhail A. Vorontsov (editors), *Proc. SPIE Vol. 5552, Atmospheric Tracking Imaging and Compensation, Michael T. Valley; Ed.*, volume 5552, 189–199. SPIE, 2004. URL `http://link.aip.org/link/?PSI/5552/189/1`.

22. Sidick, Erkin, Joseph J. Green, Rhonda M. Morgan, Catherine M. Ohara, and David C. Redding. "Adaptive cross-correlation algorithm for extended scene Shack-Hartmann wavefront sensing". *Opt. Lett.*, 33(3):213–215, 2008. URL `http://ol.osa.org/abstract.cfm?URI=ol-33-3-213`.

23. Strutt, J.W. *Proceedings London Mathematical Society*, 1871.

24. Strutt, J.W. *Phil Mag.*, 1919.

25. Tsou H. and Racho C. and Yan T. Y. "Maximum-Likelihood-Based Extended-Image Spatial Acquisition and Tracking Technique". AIAA Space Technology Conference and Exposition Alburquerque, New Mexico, USA, 1999.

26. Tyson, Robert K. *Introduction to Adaptive Optics*. SPIE Press, 2000.

27. Van Trees, H. L. *Detection, Estimation and Modulation Theory*. Wiley and Sons, 1968.

28. Védrenne, Nicolas, Vincent Michau, Clélia Robert, and Jean-Marc Conan. "Shack-Hartmann wavefront estimation with extended sources: anisoplanatism influence". *J. Opt. Soc. Am. A*, 24(9):2980–2993, 2007. URL `http://josaa.osa.org/abstract.cfm?URI=josaa-24-9-2980`.

29. Wyant, J. C. "White Light Extended Source Shearing Interferometer". *Appl. Opt.*, 13(1):200–202, 1974. URL `http://ao.osa.org/abstract.cfm?URI=ao-13-1-200`.

Index

The index is conceptual and does not designate every occurrence of a keyword. Page numbers in bold represent concept definition or introduction.

AO, 2

Code Listings
- appendix1/Makeotf2.m, 62
- appendix1/ParentCodeV11.m, 56
- appendix1/PhaseScreen.m, 67
- appendix1/centroid.m, 63
- appendix1/makeshiftvec.m, 66
- appendix1/projectionmethod2.m, 64
- appendix1/projectionmethod3.m, 65
- chapter3/ReferenceImageCode.m, 38
- chapter3/Specklemodel.m, 41
- chapter3/TiltRemovalCode.m, 43

infra red, *see* IR
IR, 1

maximum likelihood estimator, *see* MLE
ML, 30
MLE, 1

OI, 2
OPL, 26
optical path length, *see* OPL
OTF, 38

PDF, 17
Point Spread Function, *see* PSF
PSF, 6, 38

RV, 18

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
26–03–2009	Master's Thesis	Sept 2007 — Mar 2009

4. TITLE AND SUBTITLE

Evaluation of Performance
of a Maximum Likelihood Estimator
for Tracking Purposes in the Presence of Speckle Noise

5a. CONTRACT NUMBER

5b. GRANT NUMBER

F2KBAC8308G001

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Adrian Brett Monz, FLTLT, RAAF

5d. PROJECT NUMBER

ENG JON 145

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/09-29

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory(Darryl J. Sanchez, PhD)
3550 Aberdeen Avenue SE
Kirtland Air Force Base, NM 87117-5776
((505) 846-7509 (DSN: 246-7509), email: darryl.sanchez@kirtland.af.mil)

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RDS

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In many cases, optical tracking systems do not have cooperative beacons available. This is particularly true for the case involving tracking a laser illuminated target such as a missile seeker head, where the object of interest is most definitely an extended source. Furthermore the extended source is often observed in the presence of noise such as shot and speckle noise as well as atmospheric turbulence which further degrades the signal. This research effort presents the evaluation of an existing algorithm based on the maximum-likelihood technique for tilt estimation in the presence of extended sources and speckle noise, with particular application to the image motion tracking problem. Comparison is made between the performance of traditional centroiding algorithms and the existing projection-based correlation algorithm in simulation. The MLE using projection-based correlation is shown to offer improved performance in the motion tracking problem.

15. SUBJECT TERMS

speckle,tracking,likelihood,extended source

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			MAJ Jason D. Schmidt, PhD
U	U	U	UU	83	19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636, ext 7224; Jason.schmidt@afit.edu