

Computational Imaging through Atmospheric Turbulence

Stanley H. Chan
Nicholas Chmitt

Last edited: November 20, 2023

This book is based upon work supported in part by the Intelligence Advanced Research Projects Activity (IARPA) under Contract No. 2022-21102100004, and in part by the National Science Foundation under the grants CCSS-2030570 and IIS-2133032. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

Preface

The physics of imaging through atmospheric turbulence has been around for more than 80 years, and over this time it has generated a rich collection of texts and papers. We came to the field with an image processing background without having even read an optics textbook. Upon reading some of the imaging through turbulence literature, we very quickly realized the depth and breadth of the subject, but more strikingly the lack of an easy-to-read reference for people with a background like us.

The goal of this book is to provide a short introduction to the subject from the perspective of an image processing person. By image processing, we are thinking of scientists and engineers working on inverse problems in imaging systems with the goal of recovering signals from corrupted measurements. To this end, we are targeting readers who would like to know the physics of atmospheric turbulence so that they can improve their algorithms. Because of the specific perspective we take here and the targeted audience group, we shall not take a very rigorous physics-based approach. Unless the reader is already familiar with wave optics, the learning barrier will be so high that an average person would not be able to master the concepts quickly. Democratizing the ideas and educating the image processing community is an important mission of this book.

As we write this book, we aim in delivering the “big pictures” of the subject. Whenever needed, we streamline background materials including probability, optics, and optimization. Some sacrifices in the material are made to balance precision and clarity. Therefore, we do not regard this book as any substitution of the great optics books of our time. Whenever possible, we will connect the technical details back to our theme of computational imaging.

We would like to thank a lot of people who offered generous feedback to us: Jeremy Bos, Chris Dainty, Russell Hardie, Dan LeMaster,

Kevin Miller, Casey Pellizzari, Michael Roggemann, Mike Rucci, Jason Schmidt, and Mark Spencer. We also like to thank our fellow colleagues and students at Purdue: Charlie Bouman, Mark Bell, Mary Comer, and Amy Reibman, who examined several Ph.D. dissertations containing materials used in this text. Two members of our group are particularly instrumental to our turbulence project: Xingguang Zhang and Zhiyuan Mao. We thank the continuous support of IARPA and the Michigan State University team in the BRIAR program, especially Xiaoming Liu, Arun Ross, Anil Jain, Atlas Wang, Humphrey Shi, and their students.

We also wish to thank Mark de Jongh with Now Publisher in reaching out to us and helping to make this book possible. In addition, a big part of the text is presented and recorded at the 2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), and the 2022 IEEE International Conference on Image Processing. Readers interested in watching the video recordings can go to https://www.youtube.com/watch?v=g_VYOKToV_s&t=2s.

Nicholas Chmitt and Stanley H. Chan
West Lafayette, IN, United States
June 2023

Contents

1	Introduction	1
1.1	Scope of This Book	1
1.1.1	What is Atmospheric Turbulence?	1
1.1.2	Why Study This Problem?	6
1.2	Computational Imaging	10
1.2.1	Computational Camera	10
1.2.2	Computational Image Formation	16
1.2.3	Connecting with Imaging Through Turbulence .	18
1.3	Outline of This Book	20
2	Image Formation Model	24
2.1	Ray Optics or Wave Optics?	24
2.1.1	From Ray Optics to Wave Optics	24
2.1.2	This Book: Fourier Optics	25
2.1.3	Linear Space-Invariant Optical Systems	27
2.2	Waves	29
2.2.1	The Scalar Wave Equation	29
2.2.2	Helmholtz Equation	30
2.2.3	Types of Waves	33
2.2.4	Why Waves?	34
2.3	The Diffraction Problem	36
2.3.1	The Huygens-Fresnel Principle	36
2.3.2	Rayleigh-Sommerfeld Diffraction	38
2.3.3	Fresnel Diffraction	39
2.3.4	Fraunhofer Approximation	42
2.3.5	Diffraction Patterns	44
2.3.6	Diffraction Limit	45
2.4	Lenses	47

CONTENTS

2.4.1	Phase Distortion by a Lens	48
2.4.2	Impulse Response of a Lens	50
2.4.3	The Amplitude Spread Function	52
2.5	Image Formation	53
2.5.1	Coherent and Incoherent Imaging	53
2.5.2	Imaging as a Space-Invariant System	57
2.5.3	Amplitude Transfer Function	59
2.5.4	Optical Transfer Function	61
2.6	Space-Variant Systems	63
2.6.1	Two Models for Convolution	64
2.6.2	Which Model is Correct for Imaging?	66
2.7	Summary	67
3	Modeling Turbulence	69
3.1	Overview and History	70
3.1.1	The Image Formation Model	70
3.1.2	Characteristics of Turbulence-Distorted Images .	72
3.1.3	Historical Developments	73
3.2	The Atmosphere's Index of Refraction	75
3.2.1	Kolmogorov Power Spectral Density	75
3.2.2	Structure Constant C_n^2	77
3.2.3	The Structure Function $\mathcal{D}_n(\mathbf{r})$	78
3.3	Structure Function of the Phase	80
3.3.1	A Single Turbulence Layer	80
3.3.2	Multiple Phase Screens	84
3.3.3	Wave Structure Function	85
3.4	Important Applications of the Model	87
3.4.1	Fried Parameter and Isoplanatic Angle	87
3.4.2	Instantaneous OTF	90
3.4.3	Long Exposure OTF	93
3.4.4	Short Exposure OTF	95
3.4.5	Probability of Getting a Lucky Observation . .	98
3.5	Split-Step Simulation	100
3.5.1	A History of Split-Step Simulation	101
3.5.2	Split-Step Simulation: Building Blocks	101
3.5.3	Generating Phase Screens	102
3.5.4	Numerical Wave Propagation	105
3.5.5	Image Formation	106
3.5.6	Pseudo-Code for Split-Step	107
3.6	Summary	108

CONTENTS

4 Propagation-Free Modeling and Simulation	110
4.1 Motivation	110
4.1.1 What's Wrong with Split-Step?	110
4.1.2 Diagnosing the Problem and a Solution	111
4.2 A Survey of Propagation-Free Methods	112
4.2.1 Simplistic Models	112
4.2.2 Direct-to-Image Methods	113
4.2.3 Ray Tracing Methods	116
4.2.4 Other Methods	117
4.2.5 Phase-Based Simulation	117
4.3 Representing Turbulent Phase Realizations	120
4.3.1 Phase as a Basis	120
4.3.2 Zernike Basis	121
4.3.3 Zernike Coefficients	123
4.3.4 Intermodal Correlation	125
4.3.5 What's Missing?	127
4.4 The Zernike Space	128
4.4.1 Overview	128
4.4.2 Mirroring	129
4.4.3 Spatial Correlation of the Tilts	132
4.4.4 High-Order Correlations and a General Solution	137
4.4.5 Image Formation and the Zernike Space	138
4.5 Zernike-Based Simulation	139
4.5.1 Numerical Challenges	140
4.5.2 Fourier Sampling the Approximate Zernike Space	141
4.5.3 Approximating Spatially Varying Blur	142
4.5.4 The Phase-to-Space Transform	145
4.5.5 Summary	147
4.6 Additional Topics	150
4.6.1 Wavelength	150
4.6.2 Elements of Realistic Simulation	151
4.6.3 Real-Time Turbulence Generation	153
4.7 Summary	153
5 Image Restoration	155
5.1 Understanding the Forward Model	157
5.1.1 Turbulence Operator \mathcal{H}_θ	157
5.1.2 Decoupling Tilt and Blur	159
5.1.3 Tilt-then-Blur or Blur-then-Tilt?	162
5.1.4 Geometric Analysis	164

CONTENTS

5.1.5	Impact to Real Images	166
5.1.6	From Forward Model to Inverse Problems	167
5.2	Lucky Imaging	170
5.2.1	Lucky Probability via Zernike	170
5.2.2	Optical Lucky Imaging	174
5.2.3	Digital Lucky Imaging: Sharpness Metric	176
5.2.4	Digital Lucky Imaging: Fusion Step	178
5.3	Image Registration	182
5.3.1	Registration for Non-Rigid Deformation	182
5.3.2	Reference Frame	184
5.3.3	Other Image Registration Methods	187
5.4	Image Deconvolution	187
5.4.1	Tilt-then-Blur in Inverse Problems	188
5.4.2	Why Image Deconvolution?	189
5.4.3	Alternating Minimization	190
5.4.4	Modeling the Blur	191
5.4.5	Beyond Deconvolution	194
5.5	Deep Learning Methods	195
5.5.1	The Importance of Data	195
5.5.2	Network Designs	200
5.5.3	Learning Uncertainty	204
5.5.4	Single-image Restoration by Re-degradation	205
5.5.5	Multi-frame End-to-End Transformer	207
5.5.6	Generative Methods	210
5.6	Summary	212

1. Introduction

1.1 Scope of This Book

1.1.1 What is Atmospheric Turbulence?

Our Earth's atmosphere is a beautiful creation. Not only does it provide necessities for our lives, but it also makes the way we *see* things interesting. The following phenomena should be familiar to many of you: Suppose that you are standing in a desert. There is a car located at some distance from your camera. You take a picture of the car, and you see that the image of the car is wavy and blurry. To give an idea of what the images would look like, we invite you to take a look at the examples in Figure 1.1. A natural question, of course, is whether it is possible to recover the original image from these corrupted measurements. The subject of capturing and recovering images associated with atmospheric turbulence is broadly known as **imaging through atmospheric turbulence**.



Figure 1.1: Examples of imaging through atmospheric turbulence. Notice that the captured images look wavy and blurry.

CHAPTER 1. INTRODUCTION

But exactly what is atmospheric turbulence? When we talk about “atmospheric turbulence”, we have the tendency to associate the word with roars of gusty winds and unstable air. The word “turbulence” carries a lot of assumptions that may vary from person to person. To begin to describe what turbulence is, we refer to the famous Vincent van Gogh painting, *The Starry Night* (1889). The painting shows a few scientific phenomena that may correlate well with what you think. Firstly, there is a swirling air flowing in the middle of the sky. This kind of unstable air flow with a directional movement creates a subjective feeling of “turbulent”. This painting does not convey a sense of chaos, but instead a calm, peaceful night. The same is true for turbulence; turbulence is not necessarily something that is aggressive and forceful, but instead may be calm and slow-moving. Furthermore, if we look at the sky, the stars do not appear as points but as big blobs. The dispersion of the light from a point to a blob is due to a **point spread function**. In turbulence, we call it the **blur** effect. The blur effect reduces the spatial resolution of the object.



Figure 1.2: A photograph of Vincent van Gogh’s *The Starry Night* (1889), which currently hangs in the Museum of Modern Art in New York.

While the above Van Gogh painting definitely captures some essence of turbulence that correlates with how we interpret the word, there is an

1.1. SCOPE OF THIS BOOK

interesting binary definition of the word “turbulent”. In fluid dynamics, flow is either laminar or turbulent. There is nothing in between. Laminar flow occurs when all elements of fluid move in the same direction smoothly, everything else is turbulent. The steam pouring out of an engine exhibits turbulence, similar to cream as it spreads throughout a cup of coffee. So, if you agree with this definition, then turbulence is everywhere in our lives. In fact, right under your nose, the mixing molecules are experiencing turbulence!

So, when will we encounter atmospheric turbulence in images? If air is flowing under your nose, why don't you see the effects of turbulence when reading this book? The answer has to do with how light propagates across different layers of medium with a changing **index of refraction**. Figure 1.3 shows a common experiment we all can do at home. Suppose we put a pencil in a cup of water, the pencil will appear as bent. The bending of the pencil is caused by the difference in the index of refraction of air and water. Since water molecules are more densely packed, it reduces the speed of light. In optics, we say that the water introduces a phase delay. If we replace water with another transparent liquid, we will change the index of refraction and so the bending will appear differently.

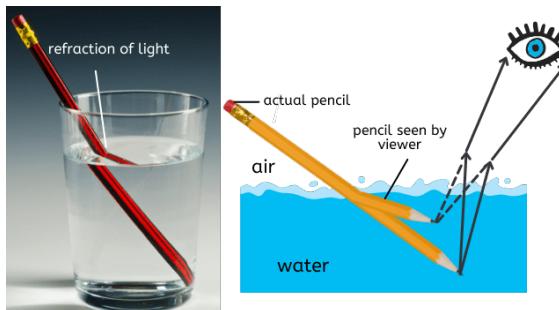


Figure 1.3: Physics of refraction. <https://smartclass4kids.com/refraction-of-light/>

This same phenomenon happens in the air but in a more nuanced fashion. We have linked density and air previously, taking this a step further, we can imagine that as the air undergoes turbulent motion some areas will have collisions while others will have particles moving away from each other. The collisions will have more particles per unit area (thus higher density) than the other areas. This will cause

CHAPTER 1. INTRODUCTION

microscopic fluctuations in the density of the medium, and thus the refractive index will also have these fluctuations. The temperature, humidity, wind velocity, altitude, etc. will all affect this behavior.

As light propagates through this medium, it will experience phase delays just as in the case of the pencil in water. For the pencil, there is a significant shifting of its observed location, but we can still easily recognize it as a pencil. In the atmosphere, the light will propagate through the medium and become dented up along the way to the observer, and the subsequent phase delay will have structured randomness. This can be imagined to some degree as looking down at the pencil in the glass of water while you're shaking the glass slightly. This will not produce such a sharp shift but instead smaller random shifts, and therefore it may be difficult to recognize what we are looking at until the water becomes still once again. This random phase-induced pixel shifting in the turbulence literature is known as **tilt**. For the atmosphere, this motion will be constant and inescapable, but there will be times when it is less aggressive than others.

The visibility of the turbulent effect depends on many factors. However, the two most important ones are distance and heat.

- **Distance.** The atmospheric turbulence effect would not be visible if the distance between the object and the camera is close. The reason is that the propagation of light requires a path. If the index of refraction is changing slightly in a local segment, then we need to accumulate many of these segments in order to create the turbulence effects. If the object is placed very close to the camera, there is simply not enough distance to accumulate the turbulence effects.
- **Heat.** The index of refraction is mainly determined by the density of air. If the temperature is high, air tends to be less densely located. But more critically is the instability of the temperature. Because the atmosphere exists in the presence of wind, external heat sources, and so on, the mixing of warm and cold air is nearly ever-present. This mixing occurs in a turbulent fashion and will contribute to the effects that we see in images.

It is easy to imagine that besides distance and heat, any environmental factor could contribute to the amount of turbulence:

- **Altitude.** The density of air is different at different altitudes. When it is near the ground, the air is denser than that in the

1.1. SCOPE OF THIS BOOK

stratosphere. So, if we take a ground-to-sky picture (e.g., for astrophotography), we will experience atmospheric turbulence. Most of the distortions would occur near the ground, and less so high up in the sky. This is mostly caused by the heat stored on the ground and sea.

- **Humidity.** Humidity affects the temperature and amount of heat. A locally more humid region will have a different heating and cooling mechanism than a dry region. Therefore, the turbulence effect will also be different.
- **Time of day.** Turbulence is stronger during the day and weaker during the night. This is again due to instability in temperature.
- **Wind velocity.** The wind will act as a mechanism to influence the mixing of hot and cool air, which may help to speed up the turbulent mixing. The wind will most notably influence how fast a turbulent image changes in time.

Because of the locality of the weather conditions (temperature, humidity, etc.), the optical path from the object to the camera could vary greatly between the two locations. For example, imagine that you want to take a picture of Jupiter. If you stand at one site and the air above you is rapidly flowing, you will see a severely distorted image. However, if we go to another site where the air is stable, the image will have a little atmospheric effect. As we can see from the illustration in Figure 1.4, the amount of turbulence along the optical path between the camera and the object will determine the quality of the image.

One of the most difficult things in imaging through atmospheric turbulence is the *randomness* of the phenomenon. A turbulence-distorted image is a *realization* of a random process. Therefore, if you take a picture now and another picture one second later, the two pictures will appear different even if the object is stationary. Various mathematical models can tell us the structure of the randomness, e.g., the covariance matrix or the autocorrelation function, but we will never know the exact realization. Therefore, unlike the bending of a pencil in water which we can predict exactly what will happen, turbulence can vary significantly even if the turbulence parameters remain the same. Such randomness does not only make the parameter estimation task extremely difficult, but it also makes image recovery difficult. Moreover, data collection is difficult too because there is no way we can capture the turbulence field. The best we can capture would be the distorted images. The lack

CHAPTER 1. INTRODUCTION

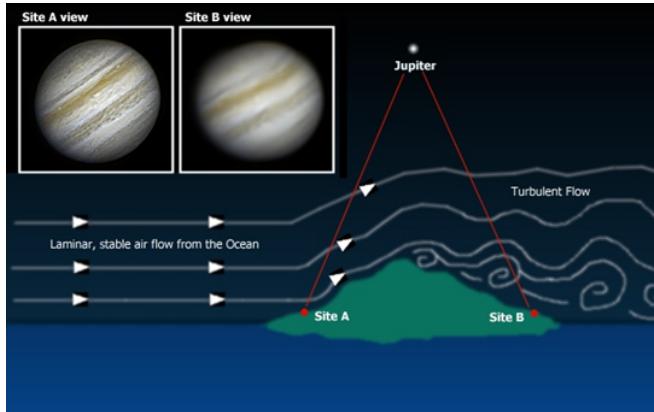


Figure 1.4: Examples of imaging through atmospheric turbulence. Notice that the changing atmosphere makes the captured images look wavy and blurry. Image source: <https://www.findlight.net/blog/deformable-mirrors-in-astronomy/>

of ground truth, model, precise measurements, etc. makes the subject significantly more challenging than many mainstream image restoration tasks such as deblurring.

1.1.2 Why Study This Problem?

If imaging is so difficult, why are we still interested in the subject? For years, imaging through turbulence is a priority for space-related missions. Some defense-related industries are interested in the subject for ground-to-ground applications. More recently, there is an increasing interest in long-range biometric applications. In what follows, we highlight a few reported applications in the literature.

Astronomy. One of the earliest motivations for developing imaging through turbulence is astrophotography. As one can imagine, this has connections with space missions including observing planets, the moon, stars, or galaxies. At least for the past century, physicists and engineers have spent a tremendous amount of effort trying to understand the birth of the universe, the behavior of black holes [1], and seeking a habitable planet other than Earth. However, seeing through a ground telescope is fundamentally limited by the atmosphere. Even if we can identify and track the target using mechanical instruments,

1.1. SCOPE OF THIS BOOK

the images are still severely distorted by atmospheric turbulence. Figure 1.5 shows a raw Saturn image we captured in West Lafayette in July 2022. The telescope used was a fairly low-end one that we could purchase from the internet. The captured images are processed carefully using many advanced deep-learning algorithms. As we can observe, the image resolution has been significantly improved. From there, physicists can perform various analyses to answer their scientific questions.

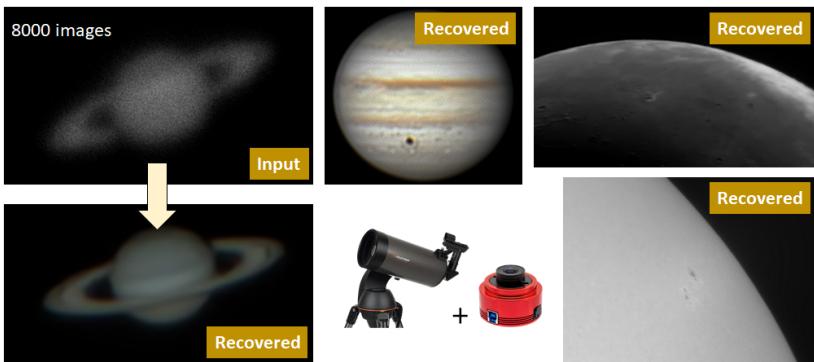


Figure 1.5: Imaging through atmospheric turbulence has an important usage in astronomy. Shown in this figure are some of the real captures we did in July 2022, in West Lafayette, Indiana. By developing image reconstruction algorithms, we can recover various planetary objects.

Defense Missions. The development of imaging through turbulence is never separable from defense missions. Unmanned aerial vehicles (UAVs) constantly carry out remote sensing tasks. However, due to the atmosphere, the images being captured would be severely distorted as shown in Figure 1.6. If we want to perform any downstream computer vision tasks such as identifying an object, understanding the action, or tracking the object, we need methods to compensate for the turbulence effect.

Some readers at this point may ask: both the current example and the previous example assume a conventional camera. Why not use a more sophisticated adaptive optics system for the astronomical application, and perhaps use a camera with a different wavelength or even use radar / LiDAR for the monitoring application? The answer is typically associated with the constraint of the cameras we have. This could include the cost, size, weight, and power. Conventional cameras are easily accessible. They are also significantly less expensive than

CHAPTER 1. INTRODUCTION



Figure 1.6: A monitoring application of techniques developed for imaging through turbulence. Shown in this figure is a reconstruction result for an airborne image.

specialized cameras. As such, it is easier to deploy conventional cameras at a large volume. For some types of missions, the added functionality of specialized cameras may not justify the cost.

Biometrics. Another increasingly popular application of imaging through atmospheric turbulence is human and object recognition. The idea is to place long-range cameras in strategically chosen locations for monitoring purposes. With the distance being long and ground-to-ground, the turbulence effect becomes unavoidable. Figure 1.7 shows a cartoon illustration of the problem. In the same figure, we also show a realistic image recovery of a long-distance object. In this example, we use optical character recognition as the task. Without appropriately processing the image, it would be extremely difficult to recognize the pattern in the image.



Figure 1.7: A monitoring application of techniques developed for imaging through turbulence. Shown in this figure is a reconstruction result for an airborne image.

1.1. SCOPE OF THIS BOOK

In computer vision and image processing, imaging through atmospheric turbulence is often an upstream task. It means that we are interested in *modeling* the physical phenomenon of the turbulence, and *processing* of the images using algorithms. Some prior work develops *hardware* to acquire images in a different way, such as the speckle imaging or adaptive optics. Although any of these can be the terminal task, they can also be fed into other downstream tasks such as detection and tracking. A frequently asked question is whether it is necessary to recover the images before running any of these recognition algorithms. We do not think there is a definitive answer at this moment, because some people argue that no matter what restoration we do, the features we use to perform the restoration are likely the same as the features we use to perform recognition. So why not just do the recognition directly? Others argue that the prior information and signal structure being explored by restoration are different from recognition. For example, in restoration, we typically use a large spatial-temporal window to align the features with the help of some specially chosen loss function (loss functions in the sense of deep learning). Integrating these techniques in recognition would basically require us to perform the restoration.

Another important aspect of the problem is the end goal. An astrophysicist probably has a different expectation from law enforcement. For astronomy, we care about a *small field of view* because the stars are so far away. We want very high resolution and we can afford to buy big telescopes. For law enforcement, it is more likely that we will use standard cameras to see a scene with a *large* field of view. The difference between a small field of view image and a large field of view image is the turbulence type. If the field of view is small, the turbulence will most likely be correlated, sometimes perfectly correlated. This is an easier problem in the sense of deblurring because the blur will be spatially invariant. However, the object will usually be small and so we will have a weak signal. If the field of view is large, the turbulence will a lot harder to handle because one point in the image may not correlate strongly with another point in the image. Therefore, algorithms developed for astrophysics may not apply to a ground-to-ground problem. The simulation parameters will be different, and the restoration techniques will also be different.

As far as this book is concerned, we are mostly interested in ground-to-ground imaging. This means that we are more focused on computer vision and image processing instead of physics and astronomy. Under this context, it makes sense for us to aim for speed, scale,

CHAPTER 1. INTRODUCTION

and image appearance. If we want to train a neural network to reconstruct the image, we expect to have a real-time simulator that can be integrated into the restoration algorithm to predict the distortion. In physics applications, we will likely have less urgency for speed as we will be more interested in resolution and potentially new discoveries. Therefore, as you read this book, we hope that you can appreciate the signal processing aspect of the problem.

1.2 Computational Imaging

One may look at the history of imaging through atmospheric turbulence, which has been studied in its modern form for 80 years since Kolmogorov, and wonder: What new angles can we take? With almost sure probability, every aspect of the problem would have some prior work, from hardware to algorithms. In parallel, signal processing and computer vision have experienced a paradigm shift in recent years with the advent of deep learning methodologies. The broader community has dedicated a considerable amount of time to these methods which are constantly advancing and replacing older methodologies. Of particular interest to us for this book is deep learning for **image reconstruction**.

The angle we take in this book is what we called **computational imaging** through turbulence, contrasting the classical imaging through turbulence. The word “computational” highlights the unique conceptual difference that we are *not* solving the inverse problem in a post-processing manner. Instead, we emphasize the significance of having a **co-design** of the computational forward model and the computational algorithm. This co-design philosophy is particularly important when we start to build neural networks to recover images. We want the network to acknowledge the physics of turbulence. Conversely, we want the forward image formation model to be compatible with the neural network, e.g., being differentiable, scalable, and fast.

1.2.1 Computational Camera

The spirit of computational imaging is to co-design the camera’s optics and the image processing algorithm. Perhaps the most common view of computational imaging is to co-design the acquisition unit (typically a camera) and the reconstruction algorithm. For the co-design to make sense, the camera should have a configurable component. For example, in coded aperture [2], the configurable component is the spatial coding

1.2. COMPUTATIONAL IMAGING

of the mask. In lensless imaging [3], the configurable component is the diffuser. The configuration component can also be a light source as in the case of structured illumination where the configurable component is the illumination pattern. Magnetic resonance and computed tomography imaging also share a similar spirit to these methods as the acquisition occurs in a different domain. Other examples of these co-design concepts include light-field imaging [4], holography, plenoptic camera, and ghost imaging or on the image sensor level, cameras such as the event camera, 1-bit quanta image sensors, time-of-flight detectors, and focal plane arrays. We refer to this co-design of hardware and algorithms as **computational camera** models. These models are at the intersection of computer vision, optics, and signal processing.

In a conventional camera, we can think of the camera as being a passive device. If we use $\mathbf{x} \in \mathbb{R}^d$ to denote the ground truth image in the object plane, and $\mathbf{y} \in \mathbb{R}^d$ to denote the observed image in the image plane, the camera can be mathematically described as the mapping \mathcal{G} from \mathbf{x} to \mathbf{y} :

$$\underbrace{\mathbf{y}}_{\text{observed image}} = \underbrace{\mathcal{G}}_{\text{camera}} (\underbrace{\mathbf{x}}_{\text{true image}}). \quad (1.1)$$

In a conventional (non-computational) camera, the camera model \mathcal{G} is often fixed and known.

A post-processing image reconstruction algorithm for the conventional camera is to find the best estimate $\hat{\mathbf{x}}$ by minimizing a certain loss function:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \text{ Loss}(\mathcal{G}(\mathbf{x}), \mathbf{y}). \quad (1.2)$$

For example, if we prefer the sum square loss, then we have

$$\text{Loss}(\mathcal{G}(\mathbf{x}), \mathbf{y}) = \|\mathcal{G}(\mathbf{x}) - \mathbf{y}\|^2. \quad (1.3)$$

Other loss functions may be used or a **regularization function** for \mathbf{x} to improve the optimization search space.

The key observation of the above equations is that in a conventional capture-then-reconstruct camera setting, the camera is *pre-defined*. While we use the model \mathcal{G} during reconstruction, the design of \mathcal{G} is completely separated from the reconstruction. We almost always want \mathcal{G} to produce the ideal image and for good reasons.

In a computational camera, the above isolation of the camera and the algorithm is replaced by a co-design philosophy. The first thing we do is to *parameterize* the camera with a finite set of tunable knobs θ .

CHAPTER 1. INTRODUCTION

This θ can contain any attributes of the camera. For example, we can input the exposure time, exposure pattern [5], coded aperture pattern, etc. By changing the parameter θ , we effectively change how the image is acquired.

The presence of the tunable parameter θ makes the overall design interesting. Instead of solving a one-variable optimization in Equation (1.2), we are conceptually solving the joint optimization

$$\hat{\mathbf{x}}, \hat{\theta} = \underset{\mathbf{x}, \theta}{\operatorname{argmin}} \text{ Loss}(\mathcal{G}_\theta(\mathbf{x}), \mathbf{y}). \quad (1.4)$$

This optimization says that while we are looking for the best estimate \mathbf{x} , we are also asking the camera \mathcal{G}_θ to be simultaneously configured. If we change \mathbf{x} , $\mathcal{G}_\theta(\mathbf{x})$ will change and so the loss will change. Similarly, if we change θ , \mathcal{G}_θ will change and so the loss will also change. The conventional camera philosophy tells us to fix a θ by standard metrics (aberrations, field of view, etc.) and estimate the best $\hat{\mathbf{x}}$. The computational camera says to design \mathcal{G}_θ in such a way that aids in our estimation of $\hat{\mathbf{x}}$.

So are we done yet? No. There are a few issues regarding the optimization defined in Equation (1.4). Firstly, it does not capture the “design” of a reconstruction algorithm. The reconstruction algorithm is not about finding one particular \mathbf{x} , but a mapping that takes any measurement \mathbf{y} and return us an estimate. Secondly, when we talk about camera design, we are not interested in one particular ground truth image \mathbf{x} . We are generally interested in all images. Finally, \mathcal{G}_θ is typically realized through hardware. This will complicate our numerical optimization, \mathcal{G}_θ doesn’t necessarily have a gradient, thus *true* joint optimization will be challenging.

To respond to the first two requests, we realize that in any image reconstruction literature, the reconstruction algorithm is always a mapping that takes \mathbf{y} and gives us an estimate $\hat{\mathbf{x}}$. Sometimes, perhaps most of the time, the reconstruction algorithm will require side information such as knowledge about \mathcal{G}_θ . In this way, we can write

$$\hat{\mathbf{x}} = \underbrace{\mathcal{R}(\mathbf{y}, \mathcal{G}_\theta)}_{\text{reconstruction}} ,$$

where \mathcal{R} is a function that takes the measurement \mathbf{y} and return us the estimate $\hat{\mathbf{x}}$.

The reconstruction algorithm can take a variety of forms. For example, in the pre-deep-learning era, the reconstruction is often a

1.2. COMPUTATIONAL IMAGING

regularized minimization, e.g.,

$$\mathcal{R}_\lambda(\mathbf{y}, \mathcal{G}_\theta) = \underset{\mathbf{x}}{\operatorname{argmin}} \quad \|\mathbf{y} - \mathcal{G}_\theta(\mathbf{x})\|^2 + \lambda g(\mathbf{x}),$$

where $g(\mathbf{x})$ is a regularization function and λ is the corresponding regularization parameter. If we choose $g(\mathbf{x}) = \|\mathbf{x}\|_{\text{TV}}$, then we have a total-variation minimization. We put a subscript λ underneath \mathcal{R} to emphasize that the reconstruction mapping is dependent on the regularization parameter λ .

In the deep-learning era, we often use a deep neural network as the reconstruction. In this case, we parameterize \mathcal{R} as \mathcal{R}_ψ so that

$$\hat{\mathbf{x}} = \underbrace{\mathcal{R}_\psi(\mathbf{y}, \mathcal{G}_\theta)}_{\text{neural network}}. \quad (1.5)$$

The way to think about ψ is that it represents the weights of the neural network. When the reconstruction neural network is trained, we will send the measurement \mathbf{y} together with the camera model \mathcal{G}_θ to the network. The network will then return us $\hat{\mathbf{x}}$.

For learning-based approaches, the reconstruction can be formulated as

$$\begin{aligned} \hat{\psi} &= \underset{\psi}{\operatorname{argmin}} \quad \mathbb{E}_{\mathbf{x}} [\text{Loss}(\hat{\mathbf{x}}, \mathbf{x})] \\ &= \underset{\psi}{\operatorname{argmin}} \quad \mathbb{E}_{\mathbf{x}} [\text{Loss}(\mathcal{R}_\psi(\mathbf{y}, \mathcal{G}_\theta), \mathbf{x})]. \end{aligned} \quad (1.6)$$

The symbol $\mathbb{E}[\cdot]$ in the equation denotes the statistical expectation. We introduce the expectation in the equation because $\text{Loss}(\hat{\mathbf{x}}, \mathbf{x})$ is only for one particular image \mathbf{x} . If we want to co-design the system for *all* images, then we need to take the average over all possible images we want to study. For example, if we are building a computational camera for human faces, then the expectation is taken over all the face images. In deep learning, we can consider \mathbf{x} as a training sample. By taking the statistical expectation over \mathbf{x} , we ask the minimization to be taken with respect to the entire training set.

Regarding the issue of the gradient of \mathcal{G}_θ , if the camera \mathcal{G}_θ is simplistic enough that a gradient calculation is possible, this can be

CHAPTER 1. INTRODUCTION

formulated as a joint optimization problem:

$$\begin{aligned} (\hat{\psi}, \hat{\theta}) &= \underset{\psi, \theta}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x}} [\text{Loss}(\hat{\mathbf{x}}, \mathbf{x})] \\ &= \underset{\psi, \theta}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x}} [\text{Loss}(\mathcal{R}_{\psi}(\mathbf{y}, \mathcal{G}_{\theta}), \mathbf{x})]. \end{aligned} \quad (1.7)$$

This typically requires some potentially limiting assumptions. One potential assumption is that **geometric optics** is sufficient in describing the behavior of the system. We present the following example of a simple \mathcal{G}_{θ} which may allow for such gradient computation.

Example. Consider a compressed sensing camera where we are interested in capturing a low-dimensional (compressed) signal instead of the full signal \mathbf{x} . The compression scheme we use is a wide matrix $\mathbf{G} \in \mathbb{R}^{m \times d}$ where $m \ll d$ so that the measured signal

$$\mathbf{y} = \mathcal{G}_{\theta}(\mathbf{x}) \stackrel{\text{def}}{=} \underbrace{\mathbf{G}\mathbf{x}}_{\text{encoded signal}} + \underbrace{\mathbf{n}}_{\text{noise term}}$$

has a much lower dimension than the true signal \mathbf{x} . Notice that we introduce a noise term to make the measurement more realistic. To reconstruct the signal we create a reconstruction algorithm (could be a neural network) that does

$$\hat{\mathbf{x}} = \mathcal{R}_{\psi}(\mathbf{y}).$$

Here, we can think of \mathcal{R}_{ψ} as a neural network parameterized by the weight ψ .

To train the neural network and simultaneously find the optimal sensing matrix \mathbf{G} , we perform the joint optimization

$$\hat{\mathbf{G}}, \hat{\psi} = \underset{\mathbf{G}, \psi}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x}, \mathbf{n}} [\text{Loss}(\mathcal{R}_{\psi}(\mathbf{G}\mathbf{x} + \mathbf{n}), \mathbf{x})].$$

The expectation is taken with respect to both the signal \mathbf{x} and noise \mathbf{n} because they are random.

If we consider the hardware feasibility, we can further pose constraints on \mathbf{G} . For example, we can require \mathbf{G} to be binary so

1.2. COMPUTATIONAL IMAGING

that it can be implemented through digital micro-mirror devices (DMD).

Figure 1.8 illustrates the conceptual diagram of a typical computational camera setup. There is a camera \mathcal{G}_θ with a few configurable knobs θ , e.g., exposure pattern, gain pattern, aperture mask, etc. Using the compressed sensing example above, θ would mean the sensing matrix \mathbf{G} . As the ground truth image \mathbf{x} is passed to the camera, we obtain a measurement \mathbf{y} . The measurement is passed to a reconstruction method \mathcal{R}_ψ to obtain an estimate $\hat{\mathbf{x}}$. The estimate is compared with the ground truth \mathbf{x} to generate a loss. The loss $\text{Loss}(\hat{\mathbf{x}}, \mathbf{x})$ is fed back to a training algorithm to update the configurable knobs θ and the reconstruction algorithm ψ . The process continues until the expected loss is minimized.

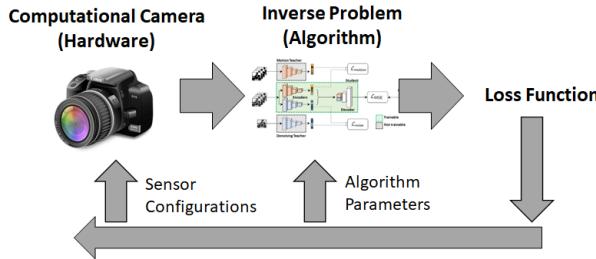


Figure 1.8: Computational camera aims at co-designing the acquisition unit (often a camera) and the image reconstruction algorithm.

The iterative procedure we described above is for *training*. During inference (i.e., deployed to the field), we fix θ to take measurements. For every captured signal \mathbf{y} , we send it to the reconstruction algorithm to produce an estimate $\hat{\mathbf{x}}$. Since the whole computational camera is trained to minimize the *average* loss over a *large* collection of typical scenarios, it will work reasonably well on similar scenarios. Some reconstruction algorithms do not require training, e.g., total variation minimization. Some camera configurations would prefer on-the-fly characteristics (i.e., they can change from one configuration to another as the scene changes). So, there is a wide range of variety here.

1.2.2 Computational Image Formation

A particular challenge for the computational camera line of work is that the gradient of \mathcal{G}_θ may be difficult or impossible to derive for complex systems. This leads us to draw a distinction that is separate from the computational camera. One essential requirement of the camera and algorithm co-optimization is that the reconstruction algorithm \mathcal{R}_ψ can “understand” what the camera \mathcal{G}_θ is doing, and vice versa. In particular, the optimization we need to solve is a *joint* optimization

$$\hat{\psi}, \hat{\theta} = \underset{\psi, \theta}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x}} \left[\text{Loss}(\mathcal{R}_\psi(\mathbf{y}, \mathcal{G}_\theta), \mathbf{x}) \right].$$

To explain the issue of “mutual understanding”, we need a neural network concept known as **back propagation**. To train a neural network \mathcal{R}_ψ , we need to back propagate the gradient of the loss $\nabla_\psi \text{Loss}(\hat{\mathbf{x}}, \mathbf{x})$ so that we can make an update of the network weights through a stochastic gradient descent algorithm. Similarly, to update the camera model \mathcal{G}_θ , we also need some kind of optimization algorithm to inform us how to update the camera parameter θ .

For generality, let’s suppose that we use a gradient descent algorithm to update both θ and ψ , denoting the overall parameter as $\varphi = [\psi, \theta]$. Back propagation creates some difficulty. By chain rule, finding $\nabla_\varphi \text{Loss}$ requires $\nabla_\varphi \mathcal{R}_\psi(\mathbf{y}, \mathcal{G}_\theta)$. This will in turn require us to take the gradient of the camera $\nabla_\theta \mathcal{G}_\theta$. However, $\nabla_\theta \mathcal{G}_\theta$ is impossible to obtain unless \mathcal{G}_θ takes a certain mathematical form that allows differentiation. A purely physical device would not allow it to happen, because there is nothing called the “gradient of Canon 5D $f/2.4$ camera, with respect to the exposure time”. Because of this physical-to-algorithmic gap, we almost always use mathematical equations \mathcal{G}_θ to represent the actual hardware.

When we use a mathematical model to substitute the true camera \mathcal{G} , we are no longer working with a computational *hardware*. To make things clear, we need to separate the true camera and a model to approximate the camera. Let’s denote the true camera as \mathcal{G} . The true camera is something we can operate, but we cannot compute the gradient. One workaround solution is to use a neural network \mathcal{H}_θ to approximate \mathcal{G} . Neural rendering, for example, is exactly following this line of thought. Instead of trying to directly handle the camera (which does not really have a “gradient”), we train a neural network \mathcal{H}_θ such that $\mathcal{H}_\theta \approx \mathcal{G}$. The training of such a neural network (if we can ignore

1.2. COMPUTATIONAL IMAGING

the reconstruction part) can be done using

$$\hat{\theta} = \operatorname{argmin}_{\theta} \mathbb{E}_x [\operatorname{Loss}(\mathcal{H}_\theta(x), \mathcal{G}(x))]. \quad (1.8)$$

Basically, we are looking for a set of network weights such that the overall network \mathcal{H}_θ can mimic the true camera. Substituting this idea into the computational camera pipeline, we can draw Figure 1.9.

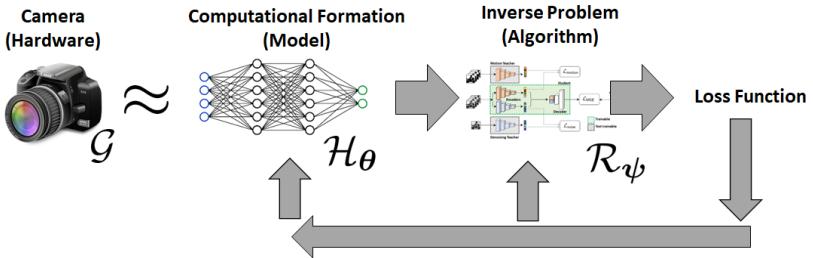


Figure 1.9: Computational camera aims to co-design the acquisition unit (often a camera) and the image reconstruction algorithm.

The co-optimization of the image formation model (as opposed to the camera) and the reconstruction algorithm can be formulated via a joint optimization

$$\begin{aligned} \hat{\theta}, \hat{\psi} = \operatorname{argmin}_{\theta, \psi} & \underbrace{\mathbb{E}_x [\operatorname{Loss}(\mathcal{H}_\theta(x), \mathcal{G}(x))]}_{\text{learn the camera model}} \\ & + \underbrace{\mathbb{E}_y [\operatorname{Loss}(\mathcal{R}_\psi(y, \mathcal{H}_\theta), x)]}_{\text{reconstruct the signal}}. \end{aligned} \quad (1.9)$$

This is a harder problem, but the two terms are intuitive. The first term is the *computational image formation* problem where we try to find a model that describes the camera. The second term is the reconstruction problem where we construct an algorithm to recover the image.

Using a neural network as \mathcal{H}_θ is one of many options. Some problems, such as developing new optical designs through nano-photonics **meta-materials** [6], would fall under the same umbrella. A piece of meta-material can be considered as an engineered glass with some desired phase properties. To design the phase profile, we have to use numerical wave approximations to mimic how light passes through the

CHAPTER 1. INTRODUCTION

meta-material. If the true meta material is denoted as \mathcal{G} , then any numerical wave approximation is \mathcal{H}_{θ} . As we try to find the best approximation \mathcal{H}_{θ} , we also need to bare in mind that the reconstruction \mathcal{R}_{ψ} can support. Therefore, even though the joint optimization looks abstract and complicated, it does capture the essence of the co-design of optics and reconstruction.

As we use a numerical model to approximate the hardware, we are no longer working with a computational camera as we are not configuring the camera directly. What we are instead doing is working with a computational model which is summarized as an equation, algorithm, or neural network. This progression from a camera to a computational camera model creates a sub-field in computational imaging, which we call **computational image formation**.

1.2.3 Connecting with Imaging Through Turbulence

At this point, we can go back to atmospheric turbulence. Readers who are familiar with a computational camera might think that for atmospheric turbulence, we can probably build some kind of computational camera to offset the reconstruction difficulty. In fact, the subject of **adaptive optics** is developed for that purpose. However, adaptive optics has both merits (such as a much better phase compensation) and limitations (such as big, expensive, limited resolution, and requiring a reference star). We shall not discuss it in detail here.

In imaging through turbulence, the degradation process is governed by *nature*. Only God knows exactly how each point on the object plane is mapped to a digital value on the image plane. Using the terminology we defined previously, we must acknowledge that there is an *unknown* target function \mathcal{G} that takes the input and gives us the output¹:

$$\mathbf{y} = \mathcal{G}(\mathbf{x}). \quad (1.10)$$

We stress that the function \mathcal{G} is unknown because it represents nature.

The complexity of turbulence is not only that we do not know the equation of the distortion. Since each turbulent instant is a realization

¹The terminology “target function” we use here comes from the literature of machine learning. A target function in machine learning refers to the ground truth mapping that we aim to learn from data. A target function is never known in machine learning because it is the subject of interest. In learning, our goal is to find a meaningful approximation to the target function by minimizing some kind of training loss.

1.2. COMPUTATIONAL IMAGING

of a random event, there is literally no way we will know exactly how light is being distorted. This is a very different (and harder) scenario compared to classical problems such as non-blind image deconvolution where we *know* the blur kernel and we also *know* the spatial invariance of the convolution. Even in the case of blind deconvolution, we know the structure of the problem.

But how about the models from the work of Kolmogorov and others created many decades ago? Aren't they experimentally verified to be accurate? Yes, but what they have proposed are models, not the target function. The models are what we called the **hypothesis functions**. Some hypothesis functions are more accurate whereas some are easier for computation. Mathematically, if we denote these hypotheses as \mathcal{H}_θ , then we have

$$\hat{\mathbf{y}} = \mathcal{H}_\theta(\mathbf{x}), \quad (1.11)$$

where $\hat{\mathbf{y}} \in \mathbb{R}^d$ denotes the prediction made by the hypothesis function \mathcal{H}_θ . Simply put, we can call $\hat{\mathbf{y}}$ as the numerically *simulated* image, and \mathcal{H}_θ as a **simulator**. The hypothesis function \mathcal{H}_θ is parameterized by the parameter $\theta \in \mathbb{R}^p$. For example, the vector θ can contain the turbulence strength, temperature, etc.

The essence of \mathcal{H}_θ is the **parameterization** of nature. We are not saying that \mathcal{H}_θ is nature, but \mathcal{H}_θ is a good *approximation* of nature. Being an approximation means that there is approximation error. It also means that we need criteria to choose a better approximation from a worse approximation. The criteria are often multifaceted. Some may have a stronger emphasis on accuracy over a narrow field of view, while some may want an extremely fast model. The trade-off is often determined *before* we even build the simulator. This means that the choice of \mathcal{H}_θ is isolated from the choice of any downstream image processing tasks including image restoration.

At this point, we should be able to draw the connection between computational imaging and imaging through turbulence. Instead of having hardware (e.g. a camera) \mathcal{H}_θ to perform the image formation, we use a computational model \mathcal{H}_θ to perform the light propagation (see Figure 1.10). Although \mathcal{H}_θ is not the same as nature, it creates an artificial process that mimics it. This artificial process adds a tremendous amount of degrees of freedom, giving us an optimization problem. Just like the camera configurations in computational cameras which should be co-optimized with the downstream reconstruction, in imaging through turbulence, the co-optimization is about the model \mathcal{H}_θ and the reconstruction algorithm \mathcal{R}_ψ . Therefore, this concept falls

CHAPTER 1. INTRODUCTION

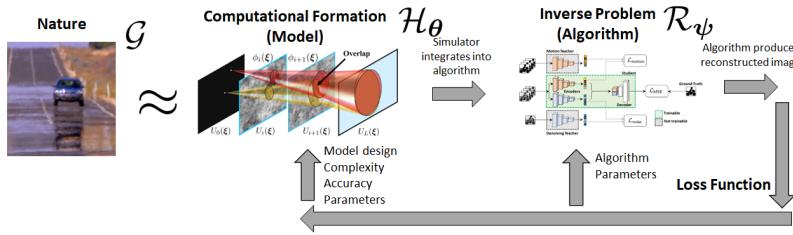


Figure 1.10: Computational imaging through turbulence focuses on co-designing the approximation to nature (which is a computational building block) with the image reconstruction algorithm.

squarely within the framework of computational image formation along with differentiable light transport and differentiable optics.

1.3 Outline of This Book

After describing computational imaging, we can now elaborate on the approaches we take in this book. There are several components:

Chapter 2. Image Formation Model. In this Chapter, we present the foundations of wave propagation physics. To this end, we need to discuss the concept of Fourier Optics. The way we view Fourier Optics is that it is a *signals and systems* perspective for wave optics. It provides us with the necessary mathematical tools such as the Huygens-Fresnel principle, Fresnel diffraction, Fraunhofer diffraction, spatial coherence, point spread function, optical transfer function, and the image formation equation.

To readers who have prior knowledge about this subject, we have a tendency to treat Fourier Optics as the ground truth \mathcal{G} . We argue that this is not the case. Under all the sufficient conditions laid out by Fourier Optics, we can build a highly reliable model \mathcal{H}_θ that can approximate \mathcal{G} up to the assumptions we make. For example, if we assume that the object is sufficiently far away from us, then Fraunhofer diffraction theory would be enough to describe how wave diffracts and propagates. However, as we step immediately outside the assumption, our model \mathcal{H}_θ will fail. Therefore, one important emphasis as we discuss the concepts is the conditions under which our derivations are correct.

Chapter 3. Modeling Turbulence. The objective of this Chapter is to provide an overview of the other half of the founda-

1.3. OUTLINE OF THIS BOOK

tion. Since we are interested in atmospheric turbulence, we need to understand its origin and its impact to the image formation process. The approach we take here is to first describe the index of refraction and its statistical properties under the Kolmogorov model. From there, we can discuss how light propagates through a stack of phase screens determined by the index of refraction statistics. The wave propagation model can be implemented numerically through a technique known as split-step propagation.

The model we conclude this Chapter with, namely the split-step propagation, is another model \mathcal{H}_θ . Compared to the model we derive in Chapter 2, the new model is customized for atmospheric turbulence. However, the design of \mathcal{H}_θ in this Chapter is solely aiming to match \mathcal{G} . We care little about the speed, and we almost never care about what image reconstruction algorithm we are going to use to recover the image. In fact, historically speaking, turbulence simulators are purely evaluated based on the statistics of the simulated data compared to the theoretical statistics. The expectation is that if the simulator is faithful, we can use it to *evaluate* any reconstruction algorithm but not *co-optimize* with the reconstruction algorithm.

We should also clarify a misconception about the known theoretical turbulence model. Under various assumptions, the theoretical model people developed over the past century is at best a *model*. Therefore, it is not the true \mathcal{G} but only a very good $\mathcal{H}_{\text{theoretical}}$. The numerical simulator we use in practice, including the split-step propagation, is another level of approximation to the theoretical model. Therefore, the relationship is best described as

$$\underbrace{\mathcal{G}}_{\text{nature}} \xrightarrow{\text{unknown}} \underbrace{\mathcal{H}_{\text{theoretical}}}_{\text{theoretical model}} \xrightarrow{\text{varies}} \underbrace{\mathcal{H}_{\text{split step}}}_{\text{numerical model}} \quad (1.12)$$

When we say “our statistics match with the theoretical prediction”, as many papers in the literature declare, we mean that $\mathcal{H}_{\text{theoretical}} \approx \mathcal{H}_{\text{split step}}$. The implicit assumption/hope here is that the approximation error in $\mathcal{H}_{\text{theoretical}} \approx \mathcal{G}$ is sufficiently small. In many situations, we know that this assumption is valid. But in some cases, for example, when we start to consider amplitude attenuation on top of the typical phase distortion, we need to resort to a more advanced theory to reduce the gap between $\mathcal{H}_{\text{theoretical}}$ and \mathcal{G} .

Chapter 4. Propagation-Free Modeling and Simulation. This Chapter is one of the two core Chapters of our book. Taking

CHAPTER 1. INTRODUCTION

into consideration of the reconstruction algorithm, existing turbulence simulators cannot serve the reconstruction purpose for many reasons:

- **Slow.** Methods such as split-step is remarkably computationally expensive. We need to loop through all the pixels of the image one by one and propagate them from one phase screen to another. The speed is too low to be even meaningful for the co-design with reconstruction algorithms.
- **Non-differentiable.** For an image formation model to be integrable as part of the reconstruction, a fundamental criterion is being *differentiable* in the sense of back propagation. Models such as the split-step propagation do not satisfy this criterion because the simulation steps are sequentially executed. Therefore, the gradient $\nabla_{\theta}\mathcal{H}_{\theta}$ is very hard to compute.
- **Inaccurate.** Some methods such as geometric deformation + Gaussian blur are fast and differentiable, but they barely mimic nature. As a result, they provide a poor approximation of the equation $\mathcal{H}_{\theta} \approx \mathcal{G}$.

With these deficiencies in mind, we present a new line of work in the recent literature that aims to bring the model fast, differentiable, and accurate. The idea is based on a propagation-free method where the distortion is implemented through a random *sampling* process rather than propagation. For the sampling to make sense, we need to derive a collection of statistical results. Some of the results are based on known literature, whereas some are new. More interestingly, some implementations require (shallow) neural networks to bridge the numerical gap.

The conclusion of this Chapter is a series of approximations that maintains the majority of the accuracy of the split-step method while offering a significant speed up and differentiability. This will allow us to integrate the image formation model with the reconstruction.

Chapter 5. Image Restoration. In this final Chapter, we discuss how the computational image formation model \mathcal{H}_{θ} can be used together with the image restoration algorithm \mathcal{R}_{ψ} . We will review the classical results in lucky imaging, frame alignment, and image deconvolution. These results will be used to guide the design of many of the latest deep neural network designs. In particular, we show how \mathcal{H}_{θ} can be used to synthesize training data. This data synthesis process is not only done prior to the training process but it can be integrated *during* the training by providing multi-scale distortions. Furthermore,

1.3. OUTLINE OF THIS BOOK

building blocks of \mathcal{H}_θ such as geometric warping and spatially varying blur can be decoupled during the training process. Since we have a powerful \mathcal{H}_θ , we can synthesize partially degraded images to guide the reconstruction at different stages. The simulator \mathcal{H}_θ can also be used during inference as a mechanism to estimate the turbulence strength. Some recent methods send feedback signals from the reconstruction algorithm \mathcal{R}_ψ to the simulator \mathcal{H}_θ by asking \mathcal{H}_θ to generate a “re-distorted” image so that we know how well our turbulence estimation is. As you can see here, a plethora of approaches can be developed when \mathcal{H}_θ is jointly optimized with \mathcal{R}_ψ .

2. Image Formation Model

The subject of this book is imaging through atmospheric turbulence. To describe such a phenomenon, we must understand both the process by which an image is formed and the way the atmosphere interferes with this process. This Chapter details image formation without consideration of any degradation by the atmosphere. Our approach here is to describe the signals and systems concepts that are familiar to readers coming from an image processing background. We will begin with a high-level discussion of optics and image formation.

2.1 Ray Optics or Wave Optics?

2.1.1 From Ray Optics to Wave Optics

When modeling a physical phenomenon, one must carefully select a mathematical model which serves the problem one seeks to analyze. For the purpose of modeling light, there are two commonly utilized models in the image processing and computer vision communities. They are the **ray tracing model** and the **wave model**. Ray tracing is useful in various forms of computer graphics, especially for lens design and rendering. In ray tracing, we “trace” light rays through space or materials, with the end result being the distribution of rays in a scene. The wave model arises from **Maxwell’s equations** [7, 8]. The wave model is incredibly accurate, but it is also analytically more difficult compared to ray tracing.

Readers with a computer graphics background may say: While it is hard to deny that the wave model is more carefully aligned with nature, it is arguably overkill for computer graphics applications. Modern GPUs are capable of performing ray tracing in real time, however,

2.1. RAY OPTICS OR WAVE OPTICS?

the same can not be said about evaluating the wave equations. Therefore, given its appropriateness, the ray model is far preferable, allowing us to form strikingly realistic images for various graphics applications [9, 10]. We completely agree with this argument. In fact, we believe that ray optics is simple, elegant, and appropriate for applications exactly in the space such as rendering. However, for imaging through atmospheric turbulence, the random fluctuation of the medium is not as easy as one would imagine. If we choose ray tracing, we may miss critical subtleties in the nature of light.

Readers with the optics background may then say: If ray tracing has deficiencies in modeling nature, we should then pursue the wave model. As long as we are not working with quantum matters, the wave model is sufficient to describe essentially all phenomena we would possibly need for engineering purposes. Specifically, we can describe the diffraction effects, phase delays, amplitude attenuation, etc, to a high degree of accuracy. We completely agree on this point, too. In fact, for almost one century since Kolmogorov, we have seen the remarkable power of wave optics theory in describing imaging through atmospheric turbulence. However, the price we need to pay is the level of mathematical rigor and computation. Wave models can be extremely complicated and sometimes intractable.

So, what should we do then? In this book, we will take an *approximate* form of the wave model. Our model is **Fourier Optics** combined with the **thin lens approximation** as described by Goodman [11]. The motivation for this model arises when one attempts to describe a system with a lens using the wave model. Before we discuss Fourier Optics and the lens models, we shall first highlight the key aspects at a high level.

2.1.2 This Book: Fourier Optics

To distinguish between the ray tracing and the model taken in this book, we visually present how these two models describe light's interaction with a lens in Figure 2.1. Beginning with ray tracing, this model describes light as rays that travel in straight lines and bend when passing through media such as a lens via **Snell's law**,

$$\frac{\sin \theta_1}{\sin \theta_2} = \frac{n_2}{n_1}.$$

This bending of light is known as **refraction**, a key feature of ray tracing. If we know the angle which the incident ray forms with the surface

CHAPTER 2. IMAGE FORMATION MODEL

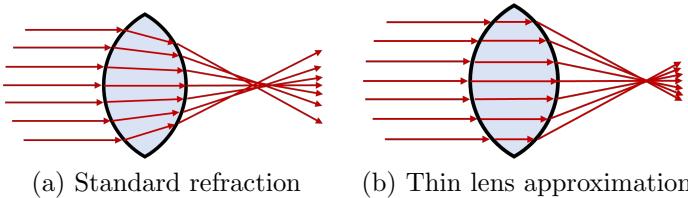


Figure 2.1: Visualizing the difference between (a) ray tracing and (b) the thin lens approximation. The thin lens model will not account for refractive issues such as the circle of confusion which arises in (a).

normal, θ_1 , and the **indices of refraction** n_1 and n_2 for both regions of space, we can compute the direction of the ray as it travels through the lens. This is shown in Figure 2.1 by the change in direction of the rays as they pass through the entrance and exit of the lens.

Instead of rays of light, the wave model describes a wave as it passes through the lens. This is given as the solution to a differential equation with the boundary conditions as determined by the lens. The wave model captures both refractive and the wave-like **diffractive** effects of an imaging system. The trade-off is the difficulty in finding general solutions to systems [11], even in the case of the simplistic system shown in Figure 2.1. Therefore, the full wave model will not be suitable for our more application-driven goals.

The Fourier Optics model combined with the thin lens approximation does something a bit different. Instead of needing to solve a differential equation, a simple rule is proposed: the thicker the lens, the more the wave is delayed in phase – a point we will revisit more appropriately later in this Chapter. In Figure 2.1, the thickest part of the lens corresponds to the wave moving slower for longer, hence being delayed relative to the wave that traversed through thinner sections of the lens. Roughly speaking, the less the wave is delayed, the less it will bend. We visualize just the effect of the thin lens approximation in Figure 2.1, though when combined with the concept of Fourier Optics, a proper depiction would include waves. For simplicity, we opt for rays in this case.

Although refraction is approximated for a simpler rule (i.e. by the thickness of the lens and not Snell's law), the model successfully captures the diffractive effects of the wave model. For brevity we will often refer to the Fourier Optics model with the thin lens approximation

2.1. RAY OPTICS OR WAVE OPTICS?

as simply “Fourier Optics”.

We summarize the key features of ray tracing, Fourier Optics, and the full wave model as follows:

1. **Ray tracing:** Ray tracing models the light as rays, which change direction upon interaction with a surface boundary. The change in direction is dictated by Snell’s law. Ray tracing describes the refractive effects of light.
2. **Wave model:** The wave model requires analysis of a differential equation which often *cannot* be solved in closed form. The wave model incorporates *both* refractive and diffractive effects.
3. **Fourier Optics:** Fourier Optics models the light as a superposition of planar waves. Additionally, it is often paired with the thin lens approximation giving us a simple analytic tool that results in a **linear space-invariant model** of an imaging system. Fourier Optics can describe the diffractive effects seen in imaging.

The advantage of Fourier Optics is exactly the fact that an imaging system with a lens can be written in terms of a Fourier transform. Additionally, the entire imaging system’s response can be written in a traditional linear systems framework, which leads us to our next introductory topic.

2.1.3 Linear Space-Invariant Optical Systems

A concept typical to an image processing person is that of an input signal, $J(\mathbf{x})$, passed through a system, $h(\mathbf{x})$, to produce $I(\mathbf{x})$, the output signal. A general system of this sort is one of the core focuses of the classical book from Oppenheim and Willsky [12]. Assuming the system is linear and space-invariant (LSI), we may write the input-output relationship via **convolution** as

$$I(\mathbf{x}) = h(\mathbf{x}) \circledast J(\mathbf{x}), \quad (2.1)$$

where \circledast denotes the convolution operator. Of course, it is also known that this may be equivalently written in the Fourier domain as

$$\mathfrak{F}\text{ourier}\{I(\mathbf{x})\} = \mathfrak{F}\text{ourier}\{h(\mathbf{x})\} \mathfrak{F}\text{ourier}\{J(\mathbf{x})\},$$

where $\mathfrak{F}\text{ourier}\{I(\mathbf{x})\}$ denotes the Fourier transform of the input signal. The goal of this book is to impart the knowledge of turbulent imaging and embed it within such a simple linear systems relationship.

CHAPTER 2. IMAGE FORMATION MODEL

It may be natural at this point to wonder what the system h would entail in an optical system. We will denote $I(\mathbf{x})$ to represent the observed optical signal at a location \mathbf{x} . For the sake of modeling, we are more interested in the terms on the right-hand side of Equation (2.1). In the case of $J(\mathbf{x})$, this will represent the “true” object, to be defined more precisely within our context in due time. As a result, $h(\mathbf{x})$ must represent everything else; it must represent how the imaging system forms an image, including effects such as diffraction, any errors in our focusing, perturbations by the medium, and so on. For the purposes of this book, $h(\mathbf{x})$ is where the majority of the optics exist.

The key result of this Chapter will be the equations that describe image formation for two different types of light. We will take the following imaging system to be our example:

$$I(\mathbf{x}) = |h(\mathbf{x})|^2 \circledast J(\mathbf{x}).$$

While this expression is slightly different from our first example, the idea remains the same. In fact, we will repeatedly return to this equation in a slightly more complicated form. More specifically, the expression written in terms of more physical entities will prove to be the more useful:

$$I(\mathbf{x}) = |\mathfrak{F}\text{ourier}\{P(\mathbf{u})e^{j\phi(\mathbf{u})}\}|^2 \circledast J(\mathbf{x}). \quad (2.2)$$

This equation is a direct result of the Fourier Optics model. Here we define $P(\mathbf{u})$ to be an indicator function describing the **aperture** of the imaging system and $\phi(\mathbf{u})$ as the **phase error** of the imaging system. Regardless of your familiarity with these concepts at this point, the purpose of presenting Equation (2.2) is to demonstrate a simple fact: we can think of our optical system using typical Fourier-based analysis and reasoning if we know what to “plug in”.

This, to many engineers, should come as a bit of a relief; the end goal is to arrive at the familiar destination of linear systems theory. There will be many details within this Chapter that are critical to arriving at and understanding Equation (2.2), though once the assumptions and limitations are understood, some of these details can be disregarded and we can continue onward from this more comfortable perspective.

2.2 Waves

2.2.1 The Scalar Wave Equation

We begin by considering the one-dimensional wave equation. Let us define a **template function** $g(x)$ over a space-time domain $S = \{(x, t) \mid x \in \mathbb{R}, t \in \mathbb{R}\}$ [13]. Using $g(x)$, we construct another function $f : S \rightarrow \mathbb{R}$ which is a shifted version of $g(x)$:

$$f(x, t) = g(x - vt), \quad (2.3)$$

where $v \in \mathbb{R}$ is the speed of propagation along a particular direction [m/s]. A different magnitude v will lead to a different trajectory of the template function through S , as shown in Figure 2.2. When light propagates through a vacuum, its speed is given as $v = c$ where $c = 3 \times 10^8$ [m/s] is the speed of light.

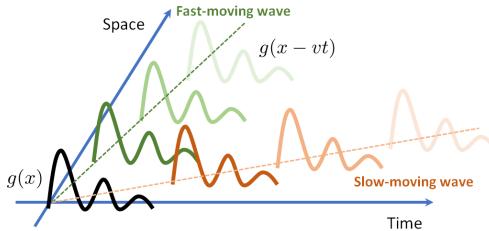


Figure 2.2: A visualization of two waves moving at two different speeds through a space-time domain S . The fast-moving wave takes a shorter time to reach a new position compared to a slow-moving wave.

Our goal is to derive an equation to summarize the behavior of $f(x, t)$ over S . To this end, let us take the first-order derivative in space:

$$\frac{\partial f(x, t)}{\partial x} = \frac{\partial g(x - vt)}{\partial x} = \frac{\partial g(x - vt)}{\partial(x - vt)} \cdot \frac{\partial(x - vt)}{\partial x} = \frac{\partial g(r)}{\partial r}, \quad (2.4)$$

where we defined $r = x - vt$. Similarly, the first-order derivative in time is

$$\frac{\partial f(x, t)}{\partial t} = \frac{\partial g(x - vt)}{\partial t} = \frac{\partial g(x - vt)}{\partial(x - vt)} \cdot \frac{\partial(x - vt)}{\partial t} = -v \frac{\partial g(r)}{\partial r}, \quad (2.5)$$

CHAPTER 2. IMAGE FORMATION MODEL

Taking the derivatives of Equation (2.4) and Equation (2.5) again, we obtain a pair of second-order derivatives:

$$\frac{\partial^2 f(x, t)}{\partial x^2} = \frac{\partial}{\partial x} \frac{\partial f(x, t)}{\partial x} = \frac{\partial^2 g(r)}{\partial r^2}, \quad (2.6)$$

$$\frac{\partial^2 f(x, t)}{\partial t^2} = \frac{\partial}{\partial t} \frac{\partial f(x, t)}{\partial t} = v^2 \frac{\partial^2 g(r)}{\partial r^2}. \quad (2.7)$$

Combining Equation (2.6) and Equation (2.7), we obtain a partial differential equation that connects space and time,

$$\frac{\partial^2 f(x, t)}{\partial x^2} = \frac{1}{v^2} \frac{\partial^2 f(x, t)}{\partial t^2}. \quad (2.8)$$

Equation (2.8) is known as the **one-dimensional wave equation**. If we were to instead write the velocity as $v = c/n$, then Equation (2.8) will become

$$\frac{\partial^2 f(x, t)}{\partial x^2} = \frac{n^2}{c^2} \frac{\partial^2 f(x, t)}{\partial t^2}. \quad (2.9)$$

Writing the velocity in this way motivates us to consider c to be the maximum speed and n to be the attenuation. In the case of optics, n is known as the index of refraction.

We may also extend the wave equation to higher dimensions. This will give us the high-dimensional **scalar wave equation**:

$$\nabla^2 f(\mathbf{x}, t) - \frac{n^2}{c^2} \frac{\partial^2 f(\mathbf{x}, t)}{\partial t^2} = 0, \quad (2.10)$$

where $\mathbf{x} = [x, y, z] \in \mathbb{R}^3$ are the spatial coordinates of the wave, and

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

is the Laplacian operator. If polarization is involved, we can extend the scalar wave equation to a **vector field** $\mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}, t), \dots, f_p(\mathbf{x}, t))$, representing the different polarizations of the wave.

2.2.2 Helmholtz Equation

We now seek to utilize Equation (2.10) using a set of elementary solutions. It is likely not a surprise an elementary solution would be a sinusoidal *wave*,

$$f(\mathbf{x}, t) = A(\mathbf{x}) \cos(2\pi\nu t + \theta(\mathbf{x})), \quad (2.11)$$

with spatially varying amplitude $A(\mathbf{x})$ and phase $\theta(\mathbf{x})$. While this form is certainly valid, it is often mathematically convenient to instead use a complex representation. Accordingly, we will typically consider a **complex wave function**:

$$u(\mathbf{x}, t) = U(\mathbf{x})e^{-j2\pi\nu t}. \quad (2.12)$$

The function $U(\mathbf{x})$ represents the complex envelope, or **phasor** [13, 11], which is then modulated by the time-evolving term $e^{-j2\pi\nu t}$.

This particular form of elementary solution presented in Equation (2.12) is useful as it separates space and time into two distinct terms. In most situations we will be interested in finding $U(\mathbf{x})$; if we further wish to describe the time behavior, this can be achieved by simply appending the temporal modulation term. We note the convention of a negative sign in $e^{-j2\pi\nu t}$ results in a clockwise rotation for an increase in radians. In order to relate the two forms of solutions, we can look at the real component of $u(\mathbf{x}, t)$, writing Equation (2.12) as

$$\begin{aligned} \text{Real}\{u(\mathbf{x}, t)\} &= A(\mathbf{x}) \cos(2\pi\nu t + \theta(\mathbf{x})), \\ &= f(\mathbf{x}, t), \end{aligned}$$

where $A(\mathbf{x})$ is the scalar amplitude and $\theta(\mathbf{x})$ is the phase, and further defining

$$U(\mathbf{x}) = A(\mathbf{x}) \exp\{-j\theta(\mathbf{x})\}. \quad (2.13)$$

Let us investigate a few physical properties of our complex envelope. Assuming a non-trivial phase function $\theta(\mathbf{x})$, the form of $U(\mathbf{x})$ suggests the cyclical nature of the envelope. For a constant value $\varphi \in [0, 2\pi)$, the set $\{\mathbf{x} \mid \angle U(\mathbf{x}) = \varphi\}$ defines a set of **wavefronts**. The distance between neighboring wavefronts is then said to be the **wavelength** of the wave, denoted as λ [m]. We also define the **optical frequency** ν (pronounced as “nu”) [s^{-1}] to be

$$\nu = \frac{v}{\lambda}, \quad (2.14)$$

where v is the velocity of the wave [m/s]. The optical frequency describes the rate at which the wave oscillates in time as it arrives.

This concept of wavefronts and wavelengths may benefit from being presented visually. To this end, we provide Figure 2.3, where we show the following consideration:

$$V(\mathbf{x}) = \left| \sum_{i=1}^N U(\mathbf{x}; \mathbf{x}_i) \right| = \left| \sum_{i=1}^N \frac{\exp\{-j(2\pi/\lambda)|\mathbf{x} - \mathbf{x}_i|\}}{|\mathbf{x} - \mathbf{x}_i|} \right|,$$

CHAPTER 2. IMAGE FORMATION MODEL

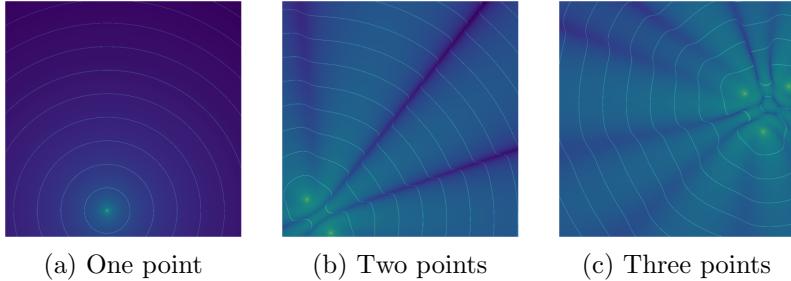


Figure 2.3: Here we show the magnitude of the field created by one, two, and three points. A bright spot corresponds to a large magnitude while a dark region indicates a smaller magnitude. In addition, we also show the wavefronts which are approximately 0 radians.

where we have chosen the amplitude function $A(\mathbf{x}; \mathbf{x}_i) = 1/|\mathbf{x} - \mathbf{x}_i|$ and phase function $\phi(\mathbf{x}; \mathbf{x}_i) = (2\pi/\lambda)|\mathbf{x} - \mathbf{x}_i|$. Simply put, there are sources of waves located at various positions \mathbf{x}_i , known as **point sources**, whose amplitude and phase vary as a function of distance from the point \mathbf{x}_i . In addition to the magnitude of this summation, we also show the wavefronts $\{\mathbf{x} \mid \angle V(\mathbf{x}) = 0\}$. We show examples of one, two, and three sources of waves. In this figure, the background shading indicates the magnitude of the complex wave, which varies in a somewhat complicated fashion due to the interaction of the waves in the complex space, along with the wavefronts indicated as solid lines.

With the intent of using our form of the complex wave function $u(\mathbf{x}, t)$ within the scalar wave equation of Equation (2.10), we present the following pair of derivatives:

$$\nabla^2 u(\mathbf{x}, t) = \{\nabla^2 U(\mathbf{x})\} e^{-j2\pi\nu t}, \quad (2.15)$$

$$\frac{\partial^2 u(\mathbf{x}, t)}{\partial t^2} = (j2\pi\nu)^2 U(\mathbf{x}) e^{-j2\pi\nu t}. \quad (2.16)$$

Substituting Equation (2.15) and Equation (2.16) into Equation (2.10), we can write the scalar wave equation in terms of the phasor,

$$\nabla^2 U(\mathbf{x}) = -(2\pi\nu)^2 \frac{n^2}{c^2} U(\mathbf{x}).$$

Letting $k = 2\pi\nu n/c = 2\pi/\lambda_\nu$ be the **wave number** (where $\lambda_\nu = c/(\nu n)$ is the wavelength), we arrive at the **Helmholtz equation**.

Definition 2.1 (Helmholtz Equation). Let the complex wave equation be $u(\mathbf{x}, t) = U(\mathbf{x})e^{-j2\pi\nu t}$ where $\mathbf{x} = [\mathbf{x}, z]^T \in \mathbb{R}^3$ and ν is optical frequency. The Helmholtz equation of the scalar field $U(\mathbf{x})$ is given by

$$\nabla^2 U(\mathbf{x}) + k^2 U(\mathbf{x}) = 0, \quad (2.17)$$

where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is the Laplacian operator, and $k = 2\pi/\lambda$ is the wave number.

2.2.3 Types of Waves

All complex wave functions taking the form of $u(\mathbf{x}, t) = U(\mathbf{x})e^{-j2\pi\nu t}$ where $U(\mathbf{x})$ is twice differentiable will satisfy the Helmholtz equation. Three forms that will be particularly useful to us are planar, spherical, and parabolic waves:

Definition 2.2 (Planar wave). The scalar field of the **planar wave** is

$$U(\mathbf{x}) = A_0 e^{-j\mathbf{k}^T \mathbf{x}} = A_0 e^{-j(k_x x + k_y y + k_z z)}, \quad (2.18)$$

where $\mathbf{k} = [k_x, k_y, k_z]^T$ is the **wave vector** such that $k_x^2 + k_y^2 + k_z^2 = k^2$ and $k = 2\pi/\lambda$ is the wave number. The constant A_0 represents the amplitude.

Definition 2.3 (Spherical wave). The scalar field $U(r)$ of the **spherical wave** is

$$U(r) = \frac{A_0}{r} e^{-jkr}, \quad (2.19)$$

where $r = |\mathbf{x}| = \sqrt{x^2 + y^2 + z^2}$ is the radius between \mathbf{x} and the origin.

Definition 2.4 (Paraboloidal wave). The scalar field of a

paraboloidal wave is given by

$$U(\mathbf{x}) = \frac{A_0 e^{-jkz}}{z} e^{-jk\left(\frac{|\mathbf{x}|^2}{2z}\right)}. \quad (2.20)$$

Suppose we have the case of a spherical wave emitting energy as shown in Figure 2.4. As we consider the wavefronts of the phasor increasingly further from the origin of the wave, we notice that the spherical wave begins to look like a parabolic wave. Increasing our distance further, we note the wave's tendency towards appearing as a plane wave. This approximate nature of a spherical wave turning into a parabolic wave and then a planar wave will be a useful conceptual and analytic tool for the purposes of this Chapter.

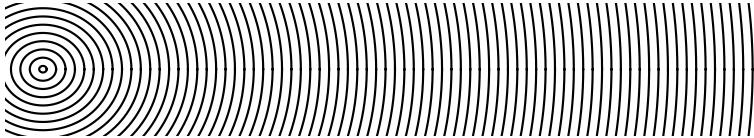


Figure 2.4: As a spherical wave propagates outwards, at a sufficient distance it becomes suitably paraboloidal (or parabolic), and further yet planar. We note that one can approximate a planar wave with a parabolic one. These ranges are not mutually exclusive, but rather the most appropriate approximation (in most cases) for the associated wave at a distance.

2.2.4 Why Waves?

Thus far we have dedicated our energy towards describing the mathematics behind wave motion in a general sense. There is nothing tying it specifically to optics at this point! There are two ways in which one may consider this reality, one more formally from Maxwell's equations and the other more intuitively through experiment.

Maxwell's equations in differential form can be written as

$$\begin{aligned} \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t}, & \nabla \times \mathbf{B} &= \mu_0 \left(\epsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J} \right), \\ \nabla \cdot \mathbf{E} &= \rho/\epsilon_0, & \nabla \cdot \mathbf{B} &= 0. \end{aligned} \quad (2.21)$$

As a first step towards developing the wave equation, we restrict our analysis to a vacuum, thus finding the “pure” form of electromagnetic

wave propagation undisturbed by outside forces. If we further define $\epsilon_0\mu_0 = 1/c^2$, this allows our equations (2.21) to take the form

$$\begin{aligned}\nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t}, & \nabla \times \mathbf{B} &= \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t}, \\ \nabla \cdot \mathbf{E} &= 0, & \nabla \cdot \mathbf{B} &= 0,\end{aligned}\quad (2.22)$$

where c is the speed of light.

Through the use of the triple scalar product and $BAC - CAB$ [14] rule, we can obtain the wave equation. Starting with taking the curl of the first equation in (2.22),

$$\begin{aligned}\nabla \times (\nabla \times \mathbf{E}) &= \nabla \times \left(-\frac{\partial \mathbf{B}}{\partial t} \right), \\ \nabla (\nabla \cdot \mathbf{E}) - \nabla \cdot \nabla \mathbf{E} &= -\frac{\partial}{\partial t} (\nabla \times \mathbf{B}), \\ \nabla^2 \mathbf{E} &\stackrel{(a)}{=} \frac{1}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2},\end{aligned}$$

where arriving at the LHS of (a) utilized the divergence property of the field \mathbf{E} in free space while the RHS used the definition of the curl of the magnetic field. By symmetry, we can state the same equation for the magnetic field \mathbf{B} . We therefore present the following definition:

Definition 2.5 (Electromagnetic Wave Equation). *The homogeneous, 3-dimensional form of the electromagnetic wave equation is given by*

$$\nabla^2 \mathbf{E} - \frac{1}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} = 0, \quad (2.23)$$

and predicts that an electromagnetic wave travels at the speed of light – a constant c .

This suggests that electromagnetic radiation travels at a constant speed c , the very same speed of light we are familiar with. Maxwell himself suggested that light must be an electromagnetic wave [11], with experimental evidence supporting this assertion in the proceeding years. Therefore, if we accept that light is electromagnetic radiation then it may be modeled as a wave.

This represents a mathematical reason why light should be treated as a wave. There are, however, more experimental reasons as to why we may choose to do so. One of the most convincing experimental reasons to model light as a wave is the phenomenon of diffraction.

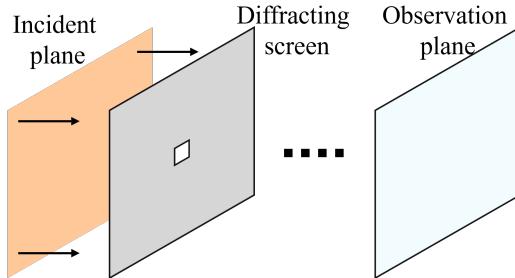


Figure 2.5: (a) The physical system we will be analyzing which gives rise to diffraction. An incident wave propagates toward the diffracting screen. We are interested in describing the wave at the observation plane. (b) Two possible diffraction patterns from [Top] a square aperture and [Bottom] a circular aperture.

2.3 The Diffraction Problem

The primary concern of Chapter 2.3 is the phenomena of diffraction. Diffraction is a clear case for the wave model and is seen most obviously in the case of a wave that is incident upon a large, non-transmitting screen with a small hole. The wave on the side opposite to that of the incidence is said to demonstrate diffractive effects. We show the system we will analyze to arrive at the mathematical description of diffraction in Figure 2.5.

Due to the wave-like behavior, we will require the Helmholtz equation (Equation (2.17)) to describe it mathematically. Problems within optics that arise from the Helmholtz equation are typically stated as boundary problems: if we know a subset of values for $U(\mathbf{x})$, we are then interested in knowing the distribution of the field at *another* set of locations. While the nature of the boundaries and the sufficient conditions for problems to be well-posed (see [7]) are beyond the scope of this book, they underlie the problems in the proceeding discussions. We instead opt for a simpler introduction first and present the main mathematical result afterward.

2.3.1 The Huygens-Fresnel Principle

The **Huygens-Fresnel principle (HFP)** is a natural choice in presenting the intuition behind how light is modeled as a wave. We begin with

2.3. THE DIFFRACTION PROBLEM

the definition:

Definition 2.6 (Huygens-Fresnel principle). *The propagation of the wave follows two principles:*

1. *Sources of waves radiate outwards spherically;*
2. *Every point along any wavefront can be considered a secondary source, referred to as wavelets, which interact.*

The Huygens-Fresnel principle tells us in simple terms how to begin with a particular wave distribution on a surface and end with the distribution on a different surface. We visualize what happens to a plane wave arriving at a diffracting screen in Figure 2.6. Here we can see the secondary wavefronts cause the wave to “leak” beyond the shadow of the screen.

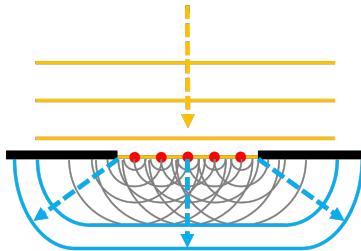


Figure 2.6: The Huygens-Fresnel Principle states that the point source radiates outwards spherically, and every point along the wavefront can be considered as a secondary source.

Let us contrast the two forms of propagation models as shown in Figure 2.7: ray optics and wave optics as described by the HFP. Ray tracing would predict the pattern an observer would see to be the same shape as the aperture, only rays that are propagating through the gap will arrive on the observation plane. Furthermore, utilizing the fact that rays travel in a straight line, we should expect to see a copy of the aperture as an observer. The HFP instead predicts that some light will leak out of the shadow of the aperture and cause the pattern to be distributed over a larger area than that predicted by ray optics.

One may notice the rippling effects in the diffraction pattern. This can be explained by the second rule of the HFP which states that the

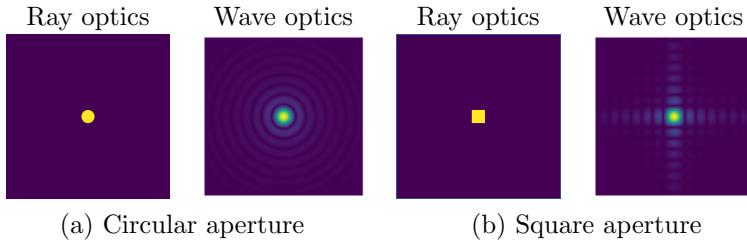


Figure 2.7: A representation of the field on the observation plane as predicted by ray tracing and the wave model. The ray optics solution does not explain how the energy leaks out beyond the shadow of the aperture, which is a mismatch from physical reality.

waves interact. This means that the waves may either constructively or destructively interfere with one another. Using the concept of phasors as before, we can understand this interference to arise from the real and imaginary components, thus contributing interference. Furthermore, the distance between the peaks in the diffraction pattern will be directly related to the wavelength of the incident light, a point we will present later in this Chapter.

The preceding discussion, of course, contains no mathematical concepts though it is often useful to have the Huygens-Fresnel principle in mind when solving problems of this nature. It is a helpful guiding tool that can aid in the visualization of propagation through various objects, and can even be used to explain refraction, phase perturbations due to atmospheric turbulence, and so on.

2.3.2 Rayleigh-Sommerfeld Diffraction

One of the mathematical approaches which describe diffractive effects is known as the Rayleigh-Sommerfeld diffraction integral, with a closely related alternative being the Fresnel-Kirchhoff diffraction integral [11]. The Rayleigh-Sommerfeld approach is often regarded as the most rigorous approach to formulating the solution to the diffraction effect that is in agreement with Maxwell's equations. The formulation seeks to describe a wave after incidence upon a diffracting screen. The problem is easily stated, yet the details involved are beyond the scope of this book. We would suggest the interested reader to Goodman [11] for an easy-to-follow treatment, and those who are interested in the rigorous

2.3. THE DIFFRACTION PROBLEM

development to Born and Wolf [7].

We depict the geometry of the problem in Figure 2.8. As in the case of the HFP, we wish to write the wave in the observation plane as a superposition of the secondary wavelets. The Rayleigh-Sommerfeld diffraction integral does just this, it describes the phasor at point \mathbf{x} due to the field at the aperture is given by the integration over the aperture Σ , which is stated in Theorem 2.1.

Theorem 2.1 (Rayleigh-Sommerfeld diffraction). *The wave observed at a point \mathbf{x} after diffracting through a surface Σ is described by*

$$U(\mathbf{x}) = \frac{1}{j\lambda} \iint_{\Sigma} U(\boldsymbol{\xi}) \frac{\exp\{jkr\}}{r} \cos \theta \, d\boldsymbol{\xi}, \quad (2.24)$$

where Σ is the source surface composed of many individual point sources, θ as the angle between the surface normal and the vector \mathbf{r} pointing from $\boldsymbol{\xi}$ to \mathbf{x} (with $r = |\mathbf{r}|$), and $k = 2\pi/\lambda$ as the wavenumber.

While Equation (2.24) is provided without proof, it can be understood somewhat intuitively. In fact, it resembles a mathematical form of the Huygens-Fresnel principle! The Rayleigh-Sommerfeld integral describes the wave at a point \mathbf{x} as a superposition of spherical waves which are weighted by their source phasor amplitudes, similar to the case of secondary wavelets in the HFP. Due to the fact that the spherical wave is modeled as a complex exponential, we will have our interaction as required. There is additionally a directivity constant which can be understood to account for the direction of propagation of the original wave. The scaling constant $1/(j\lambda)$ is a technical point that can be explained rigorously via the Rayleigh-Sommerfeld theory [11]. For the purpose of this book, we can treat the term as something that comes out of the derivative of the exponential e^{jkr} .

2.3.3 Fresnel Diffraction

The Rayleigh-Sommerfeld diffraction in Theorem 2.1 is a useful result for our specific configuration of Figure 2.5. This result can describe the wave along any parallel plane on the right side of the diffracting screen. However, this ability comes at a cost: the complicated behavior of the wave near the screen will make finding general closed forms solutions

CHAPTER 2. IMAGE FORMATION MODEL

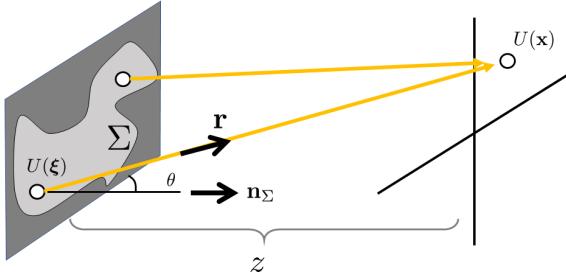


Figure 2.8: When there are more than two points in the object plane, the observed field at position \mathbf{x} will be the superposition of the fields emitting from the source plane Σ .

challenging. As a result, it will prove to be analytically intractable in most situations of interest. To make sense of the result and to derive something we can apply to image formation we need a set of approximations. The first approximation we will introduce is known as the **Fresnel approximation**. The Fresnel approximation takes inspiration from the fact that at a sufficient distance, a spherical wave can be approximated as a parabolic wave. Therefore, if we are only interested in describing the plane at a distance sufficiently far away from the screen, we may be able to obtain a more useful result via an approximation of the general Rayleigh-Sommerfeld result.

To begin our approximate analysis, we first note that $\cos \theta = z/r$ in Equation (2.24), thus it may instead be written as

$$U(\mathbf{x}) = \frac{z}{j\lambda} \iint_{\Sigma} U(\xi) \frac{\exp\{jkr\}}{r^2} d\xi. \quad (2.25)$$

We first consider the approximation for r^2 . By using Taylor's expansion, $\sqrt{1+b} \approx 1 + \frac{1}{2}b - \frac{1}{8}b^2 + \dots$. Substituting this into the distance r , we can show that

$$r = \sqrt{z^2 + |\mathbf{x} - \xi|^2} = z \sqrt{1 + \left| \frac{\mathbf{x} - \xi}{z} \right|^2} \approx z \left[1 + \frac{1}{2} \left| \frac{\mathbf{x} - \xi}{z} \right|^2 \right] \approx z, \quad (2.26)$$

if $|\mathbf{x} - \xi|^2 \ll z^2$.

With additional calculations by substituting Equation (2.26) into

2.3. THE DIFFRACTION PROBLEM

Equation (2.24), we can show that

$$\begin{aligned} U(\mathbf{x}) &= \frac{z}{j\lambda} \iint_{\Sigma} U(\boldsymbol{\xi}) \frac{\exp \left\{ jkz \left[1 + \frac{1}{2} \left| \frac{\mathbf{x}-\boldsymbol{\xi}}{z} \right|^2 \right] \right\}}{z^2} d\boldsymbol{\xi} \\ &= \frac{e^{jkz}}{j\lambda z} \iint_{-\infty}^{\infty} U_{\Sigma}(\boldsymbol{\xi}) \exp \left\{ \frac{jk}{2z} |\mathbf{x} - \boldsymbol{\xi}|^2 \right\} d\boldsymbol{\xi}, \end{aligned} \quad (2.27)$$

where $U_{\Sigma}(\boldsymbol{\xi})$ is the incident field restricted to the aperture, defined as

$$U_{\Sigma}(\boldsymbol{\xi}) = \begin{cases} U(\boldsymbol{\xi}), & \boldsymbol{\xi} \in \Sigma, \\ 0, & \text{otherwise.} \end{cases}$$

Note that we didn't utilize the approximation of Equation (2.26) in the exponential term in (2.27). This is for two reasons: (1) it is multiplied by the wavenumber k which is typically a large value, certainly in the case of the visible spectrum; (2) a small change in phase will cause a potentially significant change in the exponential. Thus, it is hard to justify the approximation in this particular case [11].

Expanding the quadratic term inside the exponent of Equation (2.27) allows us to write

$$|\mathbf{x} - \boldsymbol{\xi}|^2 = |\mathbf{x}|^2 - 2\mathbf{x}^T \boldsymbol{\xi} + |\boldsymbol{\xi}|^2. \quad (2.28)$$

Substituting Equation (2.28) into Equation (2.27), we can show that the observed field is

$$U(\mathbf{x}) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}|\mathbf{x}|^2} \iint_{-\infty}^{\infty} \left\{ U_{\Sigma}(\boldsymbol{\xi}) e^{j\frac{k}{2z}|\boldsymbol{\xi}|^2} \right\} e^{-j\frac{2\pi}{\lambda z} \mathbf{x}^T \boldsymbol{\xi}} d\boldsymbol{\xi}. \quad (2.29)$$

An important observation here is that the integration is the Fourier transform of the field $U_{\Sigma}(\boldsymbol{\xi}) e^{j\frac{k}{2z}|\boldsymbol{\xi}|^2}$. Splitting the terms in $U_{\Sigma}(\boldsymbol{\xi}) e^{j\frac{k}{2z}|\boldsymbol{\xi}|^2}$, it can be seen that the field is composed of three parts: (1) the incident field $U(\boldsymbol{\xi})$, (2) the aperture Σ , and a quadratic phase distortion $e^{j\frac{k}{2z}|\boldsymbol{\xi}|^2}$. This result is known as the **Fresnel diffraction integral**.

Theorem 2.2. *The Fresnel diffraction integral is defined as*

$$U(\mathbf{x}) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}|\mathbf{x}|^2} \iint_{-\infty}^{\infty} \left\{ U_{\Sigma}(\boldsymbol{\xi}) e^{j\frac{k}{2z}|\boldsymbol{\xi}|^2} \right\} e^{-j\frac{2\pi}{\lambda z} \mathbf{x}^T \boldsymbol{\xi}} d\boldsymbol{\xi}, \quad (2.30)$$

where $U_{\Sigma}(\boldsymbol{\xi})$ is the incident field passing through a finite aperture

CHAPTER 2. IMAGE FORMATION MODEL

Σ . Equivalently, Equation (2.30) can be written as

$$U(\mathbf{x}) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}|\mathbf{x}|^2} \mathfrak{F}\text{ourier} \left\{ U_\Sigma(\boldsymbol{\xi}) e^{j\frac{k}{2z}|\boldsymbol{\xi}|^2} \right\} \Big|_{\mathbf{f}=\frac{\mathbf{x}}{\lambda z}}, \quad (2.31)$$

where the Fourier frequencies are evaluated at $\mathbf{f} = \mathbf{x}/(\lambda z)$.

2.3.4 Fraunhofer Approximation

The Fresnel approximation is valid when we may suitably approximate the emergent waves as parabolic. For propagation distances even further away we may instead choose to use planar waves, allowing us to further simplify the result of Fresnel diffraction. With the approximation of planar waves, the quadratic phase function in Equation (2.30) can be approximated as

$$e^{j\frac{k}{2z}|\boldsymbol{\xi}|^2} \approx 1.$$

When such an approximation is valid, the Fresnel diffraction integral is simplified to

$$U(\mathbf{x}) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}|\mathbf{x}|^2} \iint_{-\infty}^{\infty} U_\Sigma(\boldsymbol{\xi}) e^{-j\frac{2\pi}{\lambda z} \mathbf{x}^T \boldsymbol{\xi}} d\boldsymbol{\xi}, \quad (2.32)$$

which we recognize as the Fourier transform of the incident field $U_\Sigma(\mathbf{x})$. This integral is known as the **Fraunhofer diffraction integral**.

Theorem 2.3. The **Fraunhofer diffraction integral** is defined as

$$U(\mathbf{x}) = \frac{e^{jkz}}{j\lambda z} \iint_{-\infty}^{\infty} U_\Sigma(\boldsymbol{\xi}) e^{-j\frac{2\pi}{\lambda z} \mathbf{x}^T \boldsymbol{\xi}} d\boldsymbol{\xi}, \quad (2.33)$$

or equivalently, in terms of Fourier transform:

$$U(\mathbf{x}) = \frac{e^{jkz}}{j\lambda z} \mathfrak{F}\text{ourier} \left\{ U_\Sigma(\boldsymbol{\xi}) \right\} \Big|_{\mathbf{f}=\frac{\mathbf{x}}{\lambda z}}. \quad (2.34)$$

The Fraunhofer diffraction integral presents an important physical result. The behavior of a wave incident upon a diffracting screen can be modeled as a device that is closely related to the Fourier transform of the input wave (specifically, the part which passes through the screen).

2.3. THE DIFFRACTION PROBLEM

Furthermore, since the Fourier transform is well-studied, we may apply standard Fourier results and intuitions to these systems. We present two examples of Fraunhofer patterns below.

Example. (Square Aperture) Suppose that the incident field is given by the square aperture

$$U_\Sigma(\xi) = \text{rect}\left(\frac{\xi}{2w_X}\right) \text{rect}\left(\frac{\eta}{2w_Y}\right), \quad (2.35)$$

where w_X and w_Y are the half-widths of the aperture. Using Equation (2.34), the Fraunhofer diffraction pattern is given by

$$U(\mathbf{x}) = \frac{e^{jkz} e^{j\frac{k}{2z}|\mathbf{x}|^2}}{j\lambda z} \mathfrak{F}\text{ourier}\left\{U_\Sigma(\xi)\right\} \Big|_{f_x = \frac{x}{\lambda z}, f_y = \frac{y}{\lambda z}}.$$

The Fourier transform of $U_\Sigma(\xi)$ is

$$\mathfrak{F}\text{ourier}\left\{\text{rect}\left(\frac{\xi}{2w_X}\right) \text{rect}\left(\frac{\eta}{2w_Y}\right)\right\} = A \text{sinc}(2w_X f_X) \text{sinc}(2w_Y f_Y),$$

where $A = 4w_X w_Y$. Thus, the Fraunhofer diffraction pattern is

$$U(\mathbf{x}) = \frac{e^{jkz} e^{j\frac{k}{2z}|\mathbf{x}|^2}}{j\lambda z} A \text{sinc}\left(\frac{2w_X x}{\lambda z}\right) \text{sinc}\left(\frac{2w_Y y}{\lambda z}\right), \quad (2.36)$$

where we substituted $f_X = x/(\lambda z)$ and $f_Y = y/(\lambda z)$. \square

Example. (Circular Aperture) Consider the circular aperture function of aperture radius D :

$$U_\Sigma(\rho) = \text{circ}\left(\frac{\rho}{D}\right) = \begin{cases} 1, & \rho < D, \\ \frac{1}{2}, & \rho = D, \\ 0, & \text{otherwise.} \end{cases} \quad (2.37)$$

where $\rho = |\xi| = \sqrt{\xi^2 + \eta^2}$ is the radius in the object plane. Since the circular aperture is circularly symmetric, we perform the calculation in the polar coordinate. The Fraunhofer diffraction

CHAPTER 2. IMAGE FORMATION MODEL

pattern is then given by the Fourier-Bessel transform **Bessel**:

$$U(r) = \frac{e^{j k z}}{j \lambda z} \exp \left\{ j \frac{k r^2}{2 z} \right\} \text{Bessel} \left\{ \text{circ} \left(\frac{\rho}{D} \right) \right\} \Big|_{f=r/(\lambda z)},$$

where $r = |\mathbf{x}| = \sqrt{|\mathbf{x}|^2}$ is the radius of the coordinate in the image plane. The Fourier-Bessel transform of the circular aperture is

$$\text{Bessel} \left\{ \text{circ} \left(\frac{\rho}{D} \right) \right\} = D^2 \frac{J_1(2\pi D f)}{D f} = 2 \cdot \pi D^2 \cdot \frac{J_1(2\pi D f)}{2\pi D f}. \quad (2.38)$$

Substituting $f = r/(\lambda z)$ into Equation (2.38) yields

$$U(r) = \frac{e^{j k z}}{j \lambda z} \exp \left\{ j \frac{k r^2}{2 z} \right\} \cdot (\pi D^2) \cdot \underbrace{\left[2 \frac{J_1(2\pi D r / (\lambda z))}{2\pi D r / (\lambda z)} \right]}_{\text{jinc}(r)}, \quad (2.39)$$

where J_1 is the Bessel function of the first kind (first order). The Fraunhofer diffraction pattern is shown in Figure 2.9. \square

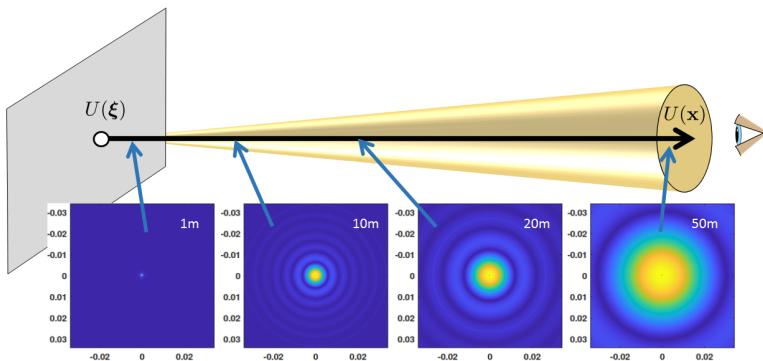


Figure 2.9: Fraunhofer diffraction pattern of a circular aperture. Notice that the spread of the diffraction pattern grows as the wave propagates.

2.3.5 Diffraction Patterns

Thus far, we have developed a way of describing the field as a result of incidence with a diffracting screen. If we were to place a sensor to

2.3. THE DIFFRACTION PROBLEM

detect the diffracted wave, we would not observe the complex wave directly. Instead, we would see the power per unit area.

The notion of power in our system is a straightforward consideration by the fact that our waves are modeled as sinusoidal. Therefore, the power at a position \mathbf{x} can be written as

$$\begin{aligned} P(\mathbf{x}) &\propto \mathbb{E}[|A(\mathbf{x}) \cos(2\pi\nu t + \theta(\mathbf{x}))|^2], \\ &\propto |A(\mathbf{x})|^2/2, \end{aligned}$$

where the proportionality is a result of the quantum efficiency and other physical properties of the system and $\mathbb{E}[\cdot]$ denotes the statistical **expectation**. We will return to the reason for this expectation in a later subsection, for now we can just consider it as a time averaging operation. Ignoring the physical quantities defining the power, we define a function known as the **intensity** of a wave to be proportional to the power [11]. The intensity is given as

$$\begin{aligned} I(\mathbf{x}) &= \mathbb{E}[|U(\mathbf{x}) \exp\{-j2\pi\nu t\}|^2], \\ &= |U(\mathbf{x})|^2. \end{aligned} \tag{2.40}$$

Instead of referring to the power directly, we typically prefer intensity for its simplicity. Furthermore, we drop the expectation in (2.40) due to the time invariance of the phasor.

This result implies that the observed pattern on a screen or sensor will simply be proportional to the magnitude square of the phasor field. Therefore, our discussions resulting in Fresnel and Fraunhofer diffraction can be applied to imaging scenarios through the application of Equation (2.40).

It is useful to note that in the above discussion, we have used the term intensity instead of **irradiance**, with the former term being the one used in Goodman's book. The units that we wish to describe are $[\text{W}/\text{m}^2]$, whereas sometimes the term intensity corresponds to units $[\text{W}/\text{m}^2/\text{sr}]$ where sr is known as a steradian. In this book we will use the term intensity to refer to power per unit area, thus being identical to the irradiance. This is done to be in coordination with Goodman and to adopt the terminology most often used by the computer vision and image processing communities.

2.3.6 Diffraction Limit

The Fraunhofer diffraction pattern derived for a circular aperture Equation (2.39) plays a significant role in imaging. If we consider the inten-

CHAPTER 2. IMAGE FORMATION MODEL

sity of the field, we can show that

$$I(r) = |U(r)|^2 = \left(\frac{2\pi D^2}{\lambda z} \right)^2 \left[2 \frac{J_1 \left(\frac{2\pi D r}{\lambda z} \right)}{\frac{2\pi D r}{\lambda z}} \right]^2. \quad (2.41)$$

The intensity distribution $I(r)$ is called the **Airy disc**. The shape of the Airy disc is shown in Figure 2.10. We marked the width of the central lobe of the Airy disc, which is measured between the two zero-crossings:

$$d = 1.22 \frac{\lambda z}{D}. \quad (2.42)$$

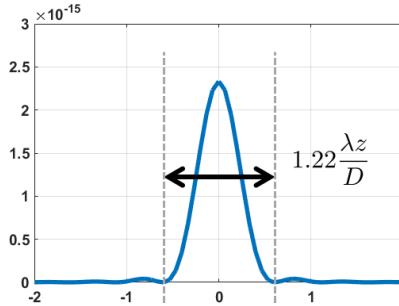


Figure 2.10: The width of the central lobe of the Airy disc is given by $d = 1.22\lambda z/D$.

Suppose that there is an aperture containing two pinholes where the centers are separated by a distance δ . As the wave leaves these two pinholes, two diffraction patterns are generated. If δ is large, we expect that the two diffraction patterns barely overlap, and so we can resolve the two points. However, as δ reduces, we will eventually reach the point where the two diffraction patterns blend into each other to the point of being indistinguishable. Between these two extremes, there exists a critical state which corresponds to the minimum separation where the two patterns are still able to be resolved. At this threshold, we say that we have reached the **Rayleigh criteria**. The situation is illustrated in Figure 2.11.

Mathematically, the minimum resolvable separation is defined at the point of half of the main lobe width,

$$\delta = 0.61 \frac{\lambda z}{D}. \quad (2.43)$$

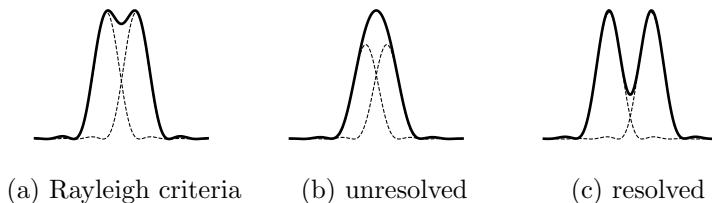


Figure 2.11: As the wave leaves from a screen consisting of two pinholes, the superposition of the diffracted waves will make the two points unresolvable. The critical separation of the two pinholes is called the Rayleigh criteria.

The factors that define Equation (2.43) are the wavelength λ , the propagation path z , and the aperture radius D . A short wavelength λ causes the spread of the diffraction to be small. Thus, we can resolve the two points with a small δ . A long propagation path z increases the spread of the diffraction pattern. Thus, the two points are better resolved if z is small. Due to the Fourier relationship of diffraction, a large aperture will shrink the size of the diffraction pattern, which will decrease δ . The opposite case similarly increases δ through the same Fourier scaling relationship.

This leads us to introduce the **numerical aperture** (NA) as $NA = D/z$. The numerical aperture allows us to write the Rayleigh criteria as

$$\delta = 0.61 \frac{\lambda}{NA}. \quad (2.44)$$

Specifically, the numerical aperture measures the half-angle subtended by the exit pupil when viewed from the image plane. NA is large when $z \ll D$.

2.4 Lenses

An imaging system almost always contains a lens, and with good reason. Previously, our discussion of waves has taken place with diffracting screens and observations at *long* distances. This feels different from the case of an imaging system where we are interested in relatively short distances. Fortunately, the way a thin lens affects propagation will allow us to use our previous results.

2.4.1 Phase Distortion by a Lens

A lens is typically composed of glass, thus having a higher index of refraction relative to free space. As a result of this relative increase, the speed of light is attenuated according to the index of refraction. Consider a lens with a shape outlined in Figure 2.12. We denote the thickness of the lens at coordinate ξ as $\Delta(\xi)$, and the maximum thickness as Δ_0 . The **phase delay** $\phi(\xi)$ caused by the lens consists of two parts: the high refractive index region provided by the lens and the free space region which is anywhere outside the lens. The sum of the two terms gives the phase delay $\phi(\xi)$

$$\phi(\xi) = \underbrace{\frac{2\pi}{\lambda} n \Delta(\xi)}_{\text{lens}} + \underbrace{\frac{2\pi}{\lambda} [\Delta_0 - \Delta(\xi)]}_{\text{free space}}, \quad (2.45)$$

where n is the refractive index of the lens.

At this point, we have defined the function $\phi(\xi)$, which we intend to apply to the wave. If we wish to apply it to the input field, i.e. the wave incident upon the lens surface, we face the problem of bending through the lens. However, if we consider the “rays” to enter and leave without deviating from their original path of propagation, then we can simply write the delay in the wave to be a function of *only* the phase delay at a point ξ .

This is the model of a **thin lens**, which is an idealized scenario in which the lens is thin enough such that any refractive effects can be approximated by the thickness. Continuing with our usage of the complex representation, we take the complex exponential of $\phi(\xi)$ in Equation (2.45), giving us the **lens transformation**:

$$t_\ell(\xi) = \exp \{j\phi(\xi)\} = \exp \{jk\Delta_0\} \exp \{jk(n-1)\Delta(\xi)\}. \quad (2.46)$$

When an incident field $U_\ell(\xi)$ reaches the lens, the lens changes its phase by multiplying $U_\ell(\xi)$ with $t_\ell(\xi)$:

$$U'_\ell(\xi) = U_\ell(\xi)P(\xi)t_\ell(\xi), \quad (2.47)$$

where $P(\xi)$ is the **pupil function** such that

$$P(\xi) = \begin{cases} 1, & \xi \text{ is inside the aperture,} \\ 0, & \text{otherwise.} \end{cases} \quad (2.48)$$

The resulting field $U'_\ell(\xi)$ is located immediately behind the lens, as shown in Figure 2.12.

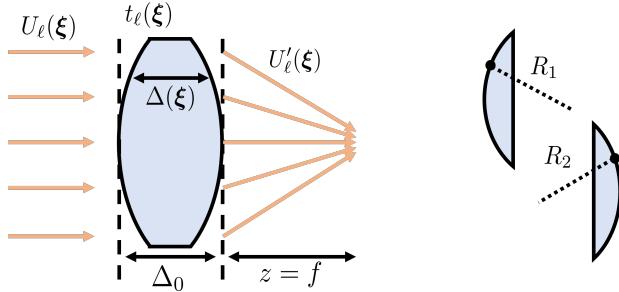


Figure 2.12: The geometry of a lens. As a ray travels from the left to the right, the radius of the left surface is denoted as R_1 , whereas the radius of the right surface is denoted as R_2 .

The lens in Figure 2.12 can be modeled as an intersection of two circles plus a constant, with radii R_1 and R_2 for left and right circles accordingly. Assuming ξ is sufficiently smaller than the two radii, the **paraxial approximation** allows us to write the thickness of the lens as the parabolic function [11]:

$$\Delta(\xi) \approx \Delta_0 - \frac{|\xi|^2}{2} \left(\frac{1}{R_1} - \frac{1}{R_2} \right). \quad (2.49)$$

Combined with the previous definitions, if we define the **focal length** f of a lens to be

$$\frac{1}{f} = (n - 1) \left(\frac{1}{R_1} - \frac{1}{R_2} \right), \quad (2.50)$$

we can write Equation (2.47) as

$$U'_\ell(\xi) = U_\ell(\xi) P(\xi) \exp \left\{ -j \frac{k}{2f} |\xi|^2 \right\}, \quad (2.51)$$

where we have removed any constant phase terms.

As in the case of our analysis of diffraction, we are additionally interested in how the wave behaves as it moves away from the lens. For this particular problem, we are concerned with the distribution of the field at the plane of distance $z = f$. Application of Fresnel integral

CHAPTER 2. IMAGE FORMATION MODEL

results in

$$U_f(\mathbf{u}) = \frac{e^{jkf}}{j\lambda f} e^{j\frac{k}{2f}|\mathbf{u}|^2} \iint_{-\infty}^{\infty} \left\{ U'_\ell(\boldsymbol{\xi}) e^{j\frac{k}{2f}|\boldsymbol{\xi}|^2} \right\} e^{-j\frac{2\pi}{\lambda f} \boldsymbol{\xi}^T \mathbf{u}} d\boldsymbol{\xi}$$

$$= \frac{e^{jkf}}{j\lambda f} e^{j\frac{k}{2f}|\mathbf{u}|^2} \iint_{-\infty}^{\infty} U_\ell(\boldsymbol{\xi}) P(\boldsymbol{\xi}) e^{-j\frac{2\pi}{\lambda f} \boldsymbol{\xi}^T \mathbf{u}} d\boldsymbol{\xi}. \quad (2.52)$$

Comparing with Equation (2.33), we realize that Equation (2.52) is the Fraunhofer diffraction pattern of the windowed field incident on the lens.

This is a somewhat surprising result. In our previous discussion of diffraction, the Fraunhofer pattern without a lens occurred a very far distance away! Now, it occurs at the **focal plane**. This highlights the key mathematical simplification by a lens, specifically an idealistic thin lens.

2.4.2 Impulse Response of a Lens

In the previous subsection, we discussed how the phase of an incident wave is transformed by a lens. In this subsection, we take a step further to study how the wave emitted from a point source at a specified distance is transformed by a lens. To analyze such an impulse response, one way is to derive the field observed at various stages as in Figure 2.13.

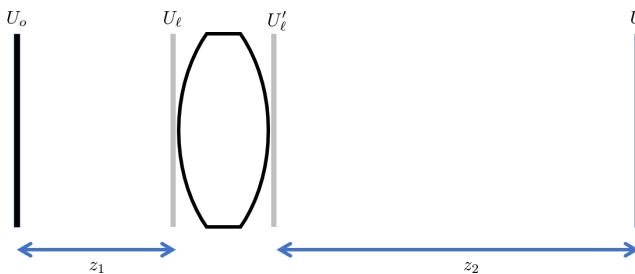


Figure 2.13: Configuration for studying the impulse response of a lens, with the “impulse” being a field we specify at a distance.

Starting from the left-hand side of Figure 2.13, we consider a **point source** as the input field U_o . A point source is described mathematically by a delta function:

$$U_o(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{u}), \quad (2.53)$$

where \mathbf{u} specifies the center of the delta function $\delta(\cdot)$. Through operator notation, we wish to present the result of

$$U_i(\mathbf{x}) = \mathfrak{Fresnel}[z_2] \{ t_\ell(\boldsymbol{\xi}) P(\boldsymbol{\xi}) \mathfrak{Fresnel}[z_1] \{ \delta(\mathbf{x} - \mathbf{u}) \} \}, \quad (2.54)$$

with $\mathfrak{Fresnel}[z]\{\cdot\}$ denoting Fresnel diffraction of $\{\cdot\}$ evaluated at a distance z . For a detailed derivation, we would point the reader towards Goodman's Fourier Optics [11]. We do, however, highlight the fact that within this derivation, the assumption most relevant to us is the assumption that the **lens law**

$$\frac{1}{z_1} + \frac{1}{z_2} = \frac{1}{f}, \quad (2.55)$$

is satisfied. This effectively assumes that the object is being focused properly.

With simplifications as fully elaborated by [11], Equation (2.54) can be simplified to

$$\begin{aligned} U_i(\mathbf{x}) \approx & \frac{1}{\lambda^2 z_1 z_2} \iint_{-\infty}^{\infty} P(\boldsymbol{\xi}) \\ & \times \exp \left\{ -jk \left[\left(\frac{u}{z_1} + \frac{x}{z_2} \right) \xi + \left(\frac{v}{z_1} + \frac{y}{z_2} \right) \eta \right] \right\} d\boldsymbol{\xi}. \end{aligned} \quad (2.56)$$

Defining the **magnification factor** (where the minus sign accounts for the image inversion)

$$M = -z_2/z_1, \quad (2.57)$$

it follows that

$$\underbrace{U_i(\mathbf{x})}_{h(\mathbf{x}, \mathbf{u})} \approx \frac{1}{\lambda^2 z_1 z_2} \iint_{-\infty}^{\infty} P(\boldsymbol{\xi}) \exp \left\{ -j \frac{2\pi}{\lambda z_2} (\mathbf{x} - M\mathbf{u})^T \boldsymbol{\xi} \right\} d\boldsymbol{\xi}. \quad (2.58)$$

This result leads us to define the **impulse response of a lens**.

Theorem 2.4 (Impulse Response of a Lens). Consider the optical setup shown in Figure 2.13. If the input is $U_o(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{u})$, the output is the impulse response, defined as

$$h(\mathbf{x}, \mathbf{u}) \approx \frac{1}{\lambda^2 z_1 z_2} \iint_{-\infty}^{\infty} P(\boldsymbol{\xi}) \exp \left\{ -j \frac{2\pi}{\lambda z_2} (\mathbf{x} - M\mathbf{u})^T \mathbf{u} \right\} d\boldsymbol{\xi}. \quad (2.59)$$

where $M = -z_2/z_1$ and $P(\mathbf{u})$ is the aperture.

2.4.3 The Amplitude Spread Function

The impulse response in Equation (2.59) is so named because $h(\mathbf{x}, \mathbf{u})$ is the outcome of the optical system with a lens when the input $U_o(\mathbf{u})$ is the delta function. For general input $U_o(\mathbf{u})$, the output is given by

$$U_i(\mathbf{x}) = \iint_{-\infty}^{\infty} h(\mathbf{x}, \mathbf{u}) U_o(\mathbf{u}) d\mathbf{u}. \quad (2.60)$$

We can think of Equation (2.60) as the superposition of the incident field U_o at different locations \mathbf{u} weighted by $h(\mathbf{x}, \mathbf{u})$. To make the equation useful, we present a sequence of normalization so that Equation (2.60) becomes a convolution.

Inspecting Equation (2.59), we consider a change of variables $\tilde{\mathbf{u}} = M\mathbf{u}$, also noting $h(\mathbf{x}, \mathbf{u})$ may be written as $h(\mathbf{x} - \tilde{\mathbf{u}})$. Without loss of generality, we may set $\tilde{\mathbf{u}} = 0$, resulting in the definition of the **amplitude spread function (ASF)**:

Theorem 2.5 (Amplitude spread function). Consider the optical setup shown in Figure 2.13. The amplitude spread function of the system is

$$h(\mathbf{x}) = \frac{A}{\lambda z_2} \iint_{-\infty}^{\infty} P(\boldsymbol{\xi}) \exp \left\{ -j \frac{2\pi}{\lambda z_2} \mathbf{x}^T \boldsymbol{\xi} \right\} d\boldsymbol{\xi}, \quad (2.61)$$

which is the Fourier transform of $P(\boldsymbol{\xi})$ evaluated at frequencies $(\frac{x}{\lambda z_2}, \frac{y}{\lambda z_2})$ (multiplied with a constant) and with $A = 1/(\lambda z_1)$, though often for notational convenience, we can set the constant $\frac{A}{\lambda z_2}$ to unity. The resulting field formed in the image plane is

$$U_i(\mathbf{x}) = \tilde{h}(\mathbf{x}) \circledast U_g(\mathbf{x}), \quad (2.62)$$

where \circledast denotes the 2D convolution, with $\tilde{h}(\mathbf{x}) = \frac{1}{M} h(\mathbf{x})$ and $U_g(\tilde{\mathbf{u}}) = \frac{1}{|M|} U_o \left(\frac{\tilde{\mathbf{u}}}{M} \right)$.

The result in Equation (2.62) is significant for a few reasons:

- The incident field $U_g(\tilde{\mathbf{u}})$ is related to what we will soon introduce as the **ideal image** produced by **geometrical optics**. Except for

the magnification factor $|M|$, there are no other distortions caused by the optics.

- In the presence of a finite-aperture lens, Equation (2.62) implies that diffraction will cause a **diffraction-limited blur** via the convolution with $h(\mathbf{x})$.
- To eliminate the effect of the diffraction-limited blur, one can make the aperture much bigger than the image sensor. According to Equation (2.61), a pupil function with an infinitely large radius will turn $h(\mathbf{x})$ to the delta function. In this case, the observed field U_i in Equation (2.62) is the geometrical optics result U_g .
- The ASF is an intrinsic property of an optical system. It is determined by the wavelength λ , propagation distance z_2 , and the geometry of the aperture $P(\mathbf{x})$. For example, a square aperture has a different ASF compared to a circular aperture.

2.5 Image Formation

Thus far discussions have primarily been about the electromagnetic field. In Chapter 2.5, we will further discuss how *images* are formed. Towards the end of Chapter 2.5, we will introduce the concepts of **amplitude transfer function** and **optical transfer function**, two major concepts within Fourier optics.

2.5.1 Coherent and Incoherent Imaging

The images we observe (by our eyes or by image sensors) are the intensities of the electromagnetic fields. Using the definitions of $U_i(\mathbf{x})$ in Equation (2.62) of Theorem 2.5 and intensity in Equation (2.40), we consider the image intensity written in terms of the source phasor:

$$\begin{aligned} I_i(\mathbf{x}) &= \mathbb{E}[|U_i(\mathbf{x})|^2] = \mathbb{E}\left[|\tilde{h}(\mathbf{x}) \circledast U_g(\mathbf{x})|^2\right] = \mathbb{E}\left[\left|\int \tilde{h}(\mathbf{x} - \tilde{\mathbf{u}})U_g(\xi) d\tilde{\mathbf{u}}\right|^2\right] \\ &= \iint \tilde{h}(\mathbf{x} - \tilde{\mathbf{u}}_1)\tilde{h}^*(\mathbf{x} - \tilde{\mathbf{u}}_2)\mathbb{E}[U_g(\tilde{\mathbf{u}}_1)U_g^*(\tilde{\mathbf{u}}_2)] d\tilde{\mathbf{u}}_1 d\tilde{\mathbf{u}}_2, \end{aligned} \quad (2.63)$$

where $\mathbb{E}[\cdot]$ denotes the expectation and $(\cdot)^*$ denotes the complex conjugate. For the proceeding discussions, we will no longer consider the waves to be fully deterministic; the waves we will now consider will

CHAPTER 2. IMAGE FORMATION MODEL

be stochastic in nature. Let us ask: where does the randomness come from?

A distributed light source generates many waves simultaneously. Depending on the type of the light source, the **instantaneous phases** of the waves are **correlated** to different extents. The correlation of the phases is described through the concept of **coherence**. Coherence can be measured in time, space, and spectrum. For the purposes of this book, we shall focus on **spatial coherence**. We are mostly interested in two types of light sources: coherent sources and incoherent sources. Figure 2.14 shows an example of both.

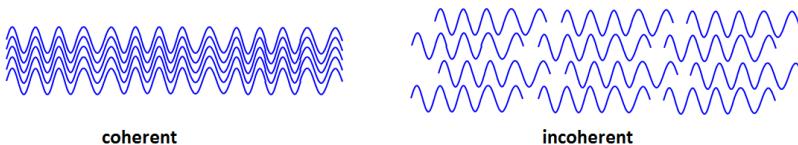


Figure 2.14: A coherent source generates trains of continuous waves where the phases are completely correlated. Note, however, that two correlated waves do not necessarily have the same initial values. They can be 180° out of phase but correlated. An incoherent source generates waves where the instantaneous phases are uncorrelated. One way to understand them is to visualize the waves as functions with discontinuous phases. Source: [<https://www.schoolphysics.co.uk/age16-19/Wave%20properties/Wave%20properties/text/Coherent%20and%20incoherent/index.html>]

The core difference between a coherent source (and thus a coherent illumination) and an incoherent source needs to be traced back to how the waves are generated. If we have a laser exciting electrons to emit energy synchronously, the resulting electromagnetic fields will be trains of continuous waves with a constant phase profile as shown in Figure 2.14. (Note, however, that the initial values of the phases do not need to be identical. For example, one wave can be 180° out of phase with another wave but they propagate at the same frequency and direction.) We say that such illumination is **coherent** because the phases of the two waves are completely correlated — if we know how the first wave propagates, we also know how the other waves propagate (assuming that we know the initial phases.) A coherent illumination is obtained when the waves are originated from the same source.

An **incoherent** illumination has the opposite behavior. Often-

2.5. IMAGE FORMATION

times, an incoherent illumination is obtained when the source produces waves randomly. One way to think of these randomly generated waves is that they are “broken trains of waves” instead of a continuous wave. Many light sources such as a tungsten bulb generate waves in such a way because electrons at different parts of the tungsten emit energy differently. When waves are emitted randomly in an independent fashion, the instantaneous phases are uncorrelated. Most of our everyday experience is illuminated by such incoherent sources.

Returning to Equation (2.63), we say the spatial coherence of two waves is measured using the **mutual intensity**:

$$J_g(\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2) = \mathbb{E} [U_g(\tilde{\mathbf{u}}_1) U_g^*(\tilde{\mathbf{u}}_2)]. \quad (2.64)$$

The way to understand Equation (2.64) is to treat the two incident fields $U_g(\tilde{\mathbf{u}}_1)$ and $U_g^*(\tilde{\mathbf{u}}_2)$ as **random variables** drawn spatially from the same random process U_g . The expectation in Equation (2.64), which applies to the product of the two terms, is the correlation of the two samples.¹

As we have stated, coherent illumination exhibits perfect correlation in the waves. Given that the complex envelope of a wave may be written $U_g(\mathbf{x}) = A_g(\mathbf{x}) \exp\{-j\phi(\mathbf{x})\}$, we may say that for a coherent wave the following is true:

$$\phi(\tilde{\mathbf{u}}_1) - \phi(\tilde{\mathbf{u}}_2) = \phi(\tilde{\mathbf{u}}_2 - \tilde{\mathbf{u}}_1). \quad (2.65)$$

In this case, the phase difference is a **deterministic** function of its separation. The expectation accordingly is dropped, with mutual intensity function

$$\begin{aligned} J_g(\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2) &= A_g(\tilde{\mathbf{u}}_1) A_g(\tilde{\mathbf{u}}_2) \exp\{j\phi(\tilde{\mathbf{u}}_2 - \tilde{\mathbf{u}}_1)\}, \\ &= U_g(\tilde{\mathbf{u}}_1) U_g^*(\tilde{\mathbf{u}}_2). \end{aligned} \quad (2.66)$$

Substituting Equation (2.66) in Equation (2.63), $I_i(\mathbf{x})$ becomes

$$\begin{aligned} I_i(\mathbf{x}) &= \iint \tilde{h}(\mathbf{x} - \tilde{\mathbf{u}}_1) \tilde{h}^*(\mathbf{x} - \tilde{\mathbf{u}}_2) U_g(\tilde{\mathbf{u}}_1) U_g^*(\tilde{\mathbf{u}}_2) d\tilde{\mathbf{u}}_1 d\tilde{\mathbf{u}}_2 \\ &= \left| \iint \tilde{h}(\mathbf{x} - \tilde{\mathbf{u}}_1) U_g(\tilde{\mathbf{u}}_1) d\tilde{\mathbf{u}}_1 \right|^2 = \left| \tilde{h}(\mathbf{x}) \circledast U_g(\mathbf{x}) \right|^2. \end{aligned} \quad (2.67)$$

¹For readers who are less familiar with random processes, we recall that the autocorrelation function $R_X(\xi_1, \xi_2)$ of a random process $X(\xi)$ is defined as $R_X(\xi_1, \xi_2) = \mathbb{E}[X(\xi_1)X(\xi_2)]$. The two instants $X(\xi_1)$ and $X(\xi_2)$ are two random variables; they may or may not be correlated. Nevertheless, the joint expectation $\mathbb{E}[X(\xi_1)X(\xi_2)]$ is well-defined. Mapping the 1D random process $X(\xi)$ to a 2D random process $U_g(\xi)$, J_g is therefore the autocorrelation function of U_g .

CHAPTER 2. IMAGE FORMATION MODEL

Therefore, for coherent light sources, the observed image $I_i(\mathbf{x})$ is the magnitude square of the convolved signal $\tilde{h}(\mathbf{x}) \circledast U_g(\mathbf{x})$. Compared with Equation (2.63), we notice that the only difference is the removal of the expectation operator $\mathbb{E}[\cdot]$.

Theorem 2.6 (Coherent Image Formation). *If the light source is coherent, then the observed image $I_i(\mathbf{x})$ is*

$$I_i(\mathbf{x}) = \left| \tilde{h}(\mathbf{x}) \circledast U_g(\mathbf{x}) \right|^2. \quad (2.68)$$

When the source is incoherent, $U_g(\mathbf{x})$ will be an independent random process. Therefore, two spatial samples $U_g(\tilde{\mathbf{u}}_1)$ and $U_g^*(\tilde{\mathbf{u}}_2)$ are two independent random variables. The mutual intensity function is then given by

$$J_g(\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2) = A_g(\tilde{\mathbf{u}}_1)A_g(\tilde{\mathbf{u}}_2)\mathbb{E}[\exp\{j(\phi(\tilde{\mathbf{u}}_2) - \phi(\tilde{\mathbf{u}}_1))\}]. \quad (2.69)$$

Further assuming that the phase function is uniformly distributed as $\phi(\mathbf{x}) \sim [0, 2\pi)$, we can show that

$$\begin{aligned} J_g(\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2) &= \begin{cases} I_g(\tilde{\mathbf{u}}_1) & \text{if } \tilde{\mathbf{u}}_1 = \tilde{\mathbf{u}}_2, \\ 0, & \text{otherwise,} \end{cases} \\ &= I_g(\tilde{\mathbf{u}}_1)\delta(\tilde{\mathbf{u}}_1 - \tilde{\mathbf{u}}_2), \end{aligned} \quad (2.70)$$

where we defined

$$I_g(\tilde{\mathbf{u}}_1) \stackrel{\text{def}}{=} \mathbb{E}[|U_g(\tilde{\mathbf{u}}_1)|^2]. \quad (2.71)$$

In other words, $J_g(\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2)$ is either $I_g(\tilde{\mathbf{u}}_1)$ (when the coordinates $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{u}}_2$ coincide) or zero (when the coordinates are different). Substituting Equation (2.70) in Equation (2.63), we show that

$$\begin{aligned} I_i(\mathbf{x}) &= \iint \tilde{h}(\mathbf{x} - \tilde{\mathbf{u}}_1)\tilde{h}^*(\mathbf{x} - \tilde{\mathbf{u}}_2) \left[I_g(\tilde{\mathbf{u}}_1)\delta(\tilde{\mathbf{u}}_1 - \tilde{\mathbf{u}}_2) \right] d\tilde{\mathbf{u}}_1 d\tilde{\mathbf{u}}_2 \\ &= \iint |\tilde{h}(\mathbf{x} - \tilde{\mathbf{u}}_1)|^2 I_g(\tilde{\mathbf{u}}_1) d\tilde{\mathbf{u}}_1 = \left| \tilde{h}(\mathbf{x}) \right|^2 \circledast I_g(\mathbf{x}). \end{aligned} \quad (2.72)$$

Therefore, the observed image is the convolution of $\left| \tilde{h}(\mathbf{x}) \right|^2$ and $|U_g(\mathbf{x})|^2$.

Theorem 2.7 (Incoherent Image Formation). *If the light source is incoherent, then the observed image $I_i(\mathbf{x})$ is*

$$I_i(\mathbf{x}) = \left| \tilde{h}(\mathbf{x}) \right|^2 \circledast I_g(\mathbf{x}). \quad (2.73)$$

Equation (2.73) describes a fundamental relationship we will consider throughout the course of this book. This equation describes the relationship between the input field's intensity and the output fields intensity through the quantity $\left| \tilde{h}(\mathbf{x}) \right|^2$, which we refer to as the **point spread function (PSF)**.

2.5.2 Imaging as a Space-Invariant System

The PSF brings us back to our intent which was stated towards the beginning of this Chapter, to write an imaging system with a lens in a standard LSI fashion. So much of computer vision and image processing takes place using incoherent imaging, thus the PSF represents the quantity that will describe a majority of imaging systems of interest to the average computer vision researcher. Furthermore, the goal of Chapter 2 will be to develop the proper background to properly describe a turbulent PSF.

We now wish to interpret Equation (2.73) beginning with recalling Equation (2.61):

$$h(\mathbf{x}) = \frac{A}{\lambda z_2} \iint_{-\infty}^{\infty} P(\boldsymbol{\xi}) \exp \left\{ -j \frac{2\pi}{\lambda z_2} \mathbf{x}^T \boldsymbol{\xi} \right\} d\boldsymbol{\xi},$$

and $h(\mathbf{x}) = M\tilde{h}(\mathbf{x})$. We can observe this result is merely the Fraunhofer pattern Equation (2.61) evaluated at distance at z_2 and with leading complex terms removed. Equivalently, we may write this using the Fourier transform as

$$h(\mathbf{x}) = \frac{A}{\lambda z_2} \mathfrak{F}\text{ourier} \{ P(\boldsymbol{\xi}) \} \Big|_{\mathbf{f} = \frac{\mathbf{x}}{\lambda z_2}}. \quad (2.74)$$

Examining Equation (2.73), we may then think of incoherent image formation in the following way. First, we form the geometric optics predicted image. This is simply ray tracing; each point will map to a single point on the imaging plane with no effects by diffraction. Second,

CHAPTER 2. IMAGE FORMATION MODEL

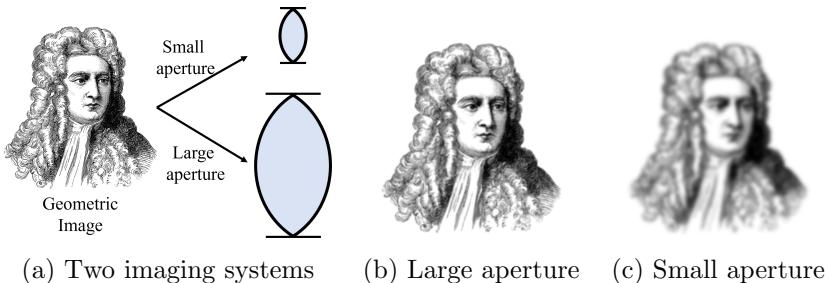


Figure 2.15: If we have a geometric image such as a perfect intensity representation of Newton (a), two imaging systems will view it differently. (b) A larger aperture will produce a smaller PSF while (c) a small aperture produces a larger PSF.

we take this geometrically predicted image and blur it according to the magnitude-squared diffraction pattern.

A typical lens-aperture system blurs the “ideal” image by a diffraction kernel. Mathematically, we may write the image formation by a perfectly in-focus system as

$$I_i(\mathbf{x}) = |\mathfrak{F}\text{ourier}\{P(\boldsymbol{\xi})\}|^2 \circledast I_g(\mathbf{x}).$$

We will typically leave off such terms which correspond to the resizing of the Fourier transform for brevity as this convolution should be understood to be done according to the proper system parameters. This accounts for the diffraction limit by the blur acting as a low pass filter. Due to the scaling property of the Fourier transform, a larger aperture corresponds to a smaller blur, which we present a visualization of this in Figure 2.15.

This leads us to consider the case in which the imaging system is not perfectly in-focus. An assumption within our development of Equation (2.73) is that our system is focused, assuming that lens law (see Equation (2.55)) was satisfied. Let us remove this assumption, instead assuming there is a slight focusing error such that

$$\frac{1}{z_1} + \frac{1}{z_2} - \frac{1}{f} = \epsilon. \quad (2.75)$$

The resulting impulse response will then be given by [11]

$$h(\mathbf{x}) = \frac{A}{\lambda z_2} \iint_{-\infty}^{\infty} P(\boldsymbol{\xi}) \exp \left\{ j \frac{k\epsilon}{2} |\boldsymbol{\xi}|^2 \right\} \exp \left\{ -j \frac{2\pi}{\lambda z_2} \mathbf{x}^T \boldsymbol{\xi} \right\} d\boldsymbol{\xi}. \quad (2.76)$$

2.5. IMAGE FORMATION

Using this in place of our derivation for incoherent imaging, the image will then be formed according to

$$I_i(\mathbf{x}) = |\mathfrak{F}\text{ourier}\{P(\boldsymbol{\xi})e^{jke^{|\boldsymbol{\xi}|^2/2}}\}|^2 \circledast I_g(\mathbf{x}).$$

More generally, we may write

$$I_i(\mathbf{x}) = |\mathfrak{F}\text{ourier}\{P(\boldsymbol{\xi})e^{jk\phi(\boldsymbol{\xi})}\}|^2 \circledast I_g(\mathbf{x}), \quad (2.77)$$

where $\phi(\boldsymbol{\xi})$ gives the path-error in meters. We will return to this concept in more detail for the purposes of imaging through atmospheric turbulence.

2.5.3 Amplitude Transfer Function

The image formation equations for a coherent system in Theorem 2.6 and an incoherent system in Theorem 2.7 have one thing in common: they are both expressed using a convolution. The underlying mathematical assumption we made is that the image formation process is spatially invariant and so the system is determined by the impulse response. For such a system, it is also common to describe image formation in the Fourier domain. To see how this can be made possible, we define

$$\begin{aligned} G_g(\mathbf{f}) &\stackrel{\text{def}}{=} \mathfrak{F}\text{ourier}\{U_g(\mathbf{x})\} = \int_{-\infty}^{\infty} U_g(\mathbf{x}) \exp\{-j2\pi\mathbf{f}^T \mathbf{x}\} d\mathbf{x} \\ G_i(\mathbf{f}) &\stackrel{\text{def}}{=} \mathfrak{F}\text{ourier}\{U_i(\mathbf{x})\} = \int_{-\infty}^{\infty} U_i(\mathbf{x}) \exp\{-j2\pi\mathbf{f}^T \mathbf{x}\} d\mathbf{x}, \end{aligned}$$

where $\mathbf{f} = [f_X, f_Y]$ is the frequency. In addition, we define the amplitude transfer function to be the Fourier transfer of the ASF:

Definition 2.7 (Amplitude transfer function). *The amplitude transfer function is*

$$H(\mathbf{f}) \stackrel{\text{def}}{=} \mathfrak{F}\text{ourier}\{h(\mathbf{x})\} = \int_{-\infty}^{\infty} h(\mathbf{x}) \exp\{-j2\pi\mathbf{f}^T \mathbf{x}\} d\mathbf{x}.$$

For **coherent imaging systems**, the observed field is given by (see Equation (2.62))

$$U_i(\mathbf{x}) = h(\mathbf{x}) \circledast U_g(\mathbf{x}), \quad (2.78)$$

CHAPTER 2. IMAGE FORMATION MODEL

where we have dropped the constant phase terms. Taking the magnitude squares on both sides, we obtain

$$I_i(\mathbf{x}) = |h(\mathbf{x}) \circledast U_g(\mathbf{x})|^2. \quad (2.79)$$

Therefore, for coherent illumination, the observed intensity is the magnitude square of the convolution.

Going back to Equation (2.78), the standard property of the Fourier analysis tells us that

$$G_i(\mathbf{f}) = H(\mathbf{f})G_g(\mathbf{f}). \quad (2.80)$$

The amplitude transfer function, according to Equation (2.61) is therefore

$$\begin{aligned} H(\mathbf{f}) &\stackrel{\text{def}}{=} \mathfrak{F}\text{ourier} \left\{ \frac{A}{\lambda z_2} \int_{-\infty}^{\infty} P(\boldsymbol{\xi}) \exp \left\{ -j \frac{2\pi}{\lambda z_2} \mathbf{x}^T \boldsymbol{\xi} \right\} d\boldsymbol{\xi} \right\} \\ &= \int_{-\infty}^{\infty} \left\{ \frac{A}{\lambda z_2} \int_{-\infty}^{\infty} P(\boldsymbol{\xi}) e^{-j \frac{2\pi}{\lambda z_2} \mathbf{x}^T \boldsymbol{\xi}} d\boldsymbol{\xi} \right\} e^{-j 2\pi \mathbf{f}^T \mathbf{x}} d\mathbf{x} \end{aligned}$$

which is nothing but taking Fourier transform twice. It is not difficult to show that a function $f(\mathbf{x})$ transformed twice by the Fourier transform is the flipped version $f(-\mathbf{x})$. Therefore,

$$H(\mathbf{f}) = P(-\lambda z_2 \mathbf{f}) = P(\lambda z_2 \mathbf{f}),$$

where we ignored the constant $A/(\lambda z_2)$ and used the property of a symmetry circular pupil function that $P(\mathbf{x}) = P(-\mathbf{x})$. Therefore, the amplitude transfer function is essentially the pupil function with a scaled coordinate.

Example. Suppose that the pupil function is

$$P(\boldsymbol{\xi}) = \text{rect} \left(\frac{\boldsymbol{\xi}}{2w} \right) \text{rect} \left(\frac{\boldsymbol{\eta}}{2w} \right). \quad (2.81)$$

The amplitude transfer function is just the pupil function with a

scaled coordinate

$$\begin{aligned} H(\mathbf{f}) &= P(\lambda z_2 \mathbf{f}) \\ &= \text{rect}\left(\frac{\lambda z_2 f_\xi}{2w}\right) \text{rect}\left(\frac{\lambda z_2 f_\eta}{2w}\right). \end{aligned}$$

2.5.4 Optical Transfer Function

For **incoherent imaging systems**, the observed intensity $I_i(\mathbf{x})$ follows from Equation (2.73) that

$$\begin{aligned} I_i(\mathbf{x}) &= |h(\mathbf{x})|^2 \circledast I_g(\mathbf{x}) \\ &= \kappa \int_{-\infty}^{\infty} |h(\mathbf{x} - \mathbf{u})|^2 I_g(\mathbf{u}) d\mathbf{u}, \end{aligned} \quad (2.82)$$

where κ is the constant $1/|M|^2$ resulting from Theorem 2.5. The convolution here is defined for $|h|^2$ and I_g . Thus, we consider the following terms:

$$\begin{aligned} \mathcal{G}_g(\mathbf{f}) &= \mathfrak{Fourier}\{I_g(\mathbf{x})\} = \frac{\int_{-\infty}^{\infty} I_g(\mathbf{x}) \exp\{-j2\pi\mathbf{f}^T \mathbf{x}\} d\mathbf{x}}{\int_{-\infty}^{\infty} I_g(\mathbf{x}) d\mathbf{x}}, \\ \mathcal{G}_i(\mathbf{f}) &= \mathfrak{Fourier}\{I_i(\mathbf{x})\} = \frac{\int_{-\infty}^{\infty} I_i(\mathbf{x}) \exp\{-j2\pi\mathbf{f}^T \mathbf{x}\} d\mathbf{x}}{\int_{-\infty}^{\infty} I_i(\mathbf{x}) d\mathbf{x}}, \end{aligned}$$

where we normalize the Fourier transforms by the signal powers so that the maximum intensities of \mathcal{G}_g and \mathcal{G}_i are the unity. We then define the optical transfer function:

Definition 2.8 (Optical transfer function). *The optical transfer function is*

$$\mathcal{H}(\mathbf{f}) = \mathfrak{Fourier}\{|h(\mathbf{x})|^2\} = \frac{\int_{-\infty}^{\infty} |h(\mathbf{x})|^2 \exp\{-j2\pi\mathbf{f}^T \mathbf{x}\} d\mathbf{x}}{\int_{-\infty}^{\infty} |h(\mathbf{x})|^2 d\mathbf{x}}.$$

In Fourier domain Equation (2.73) becomes

$$\mathcal{G}_i(\mathbf{f}) = \mathcal{H}(\mathbf{f}) \mathcal{G}_g(\mathbf{f}). \quad (2.83)$$

Comparing Equation (2.79) with Equation (2.83), we observe that the intensity of the image in an incoherent illumination is the convolution of the magnitude squares.

CHAPTER 2. IMAGE FORMATION MODEL

With the ATF and OTF defined as

$$H(\mathbf{f}) = \text{Fourier}\{h(\mathbf{x})\},$$

$$\mathcal{H}(\mathbf{f}) = \text{Fourier}\{|h(\mathbf{x})|^2\},$$

we can write the relationship between the optical transfer function \mathcal{H} and the amplitude transfer function H is given by

$$\mathcal{H}(\mathbf{f}) = \frac{\int_{-\infty}^{\infty} H(\mathbf{p}') H^*(\mathbf{p}' - \mathbf{f}) d\mathbf{p}'}{\int_{-\infty}^{\infty} |H(\mathbf{p}')|^2 d\mathbf{p}'}, \quad (2.84)$$

where we note the term $|h(\mathbf{x})|^2$ turned into convolution in the Fourier domain. Therefore, $\mathcal{H}(\mathbf{f})$ is the auto-correlation of the $H(\mathbf{x})$. The geometry of $\mathcal{H}(\mathbf{f})$ can be seen by doing a simple change of variable. Letting $\mathbf{p}' + \frac{\mathbf{f}}{2}$, Equation (2.84) can be re-centered as

$$\mathcal{H}(\mathbf{f}) = \frac{\int_{-\infty}^{\infty} H(\mathbf{p}' + \frac{\mathbf{f}}{2}) H^*(\mathbf{p}' - \frac{\mathbf{f}}{2}) d\mathbf{p}'}{\int_{-\infty}^{\infty} |H(\mathbf{p}')|^2 d\mathbf{p}'}. \quad (2.85)$$

Referring to Figure 2.16, $\mathcal{H}(\mathbf{f})$ is the ratio

$$\mathcal{H}(\mathbf{f}) = \frac{\text{area of overlap}}{\text{total area}}.$$

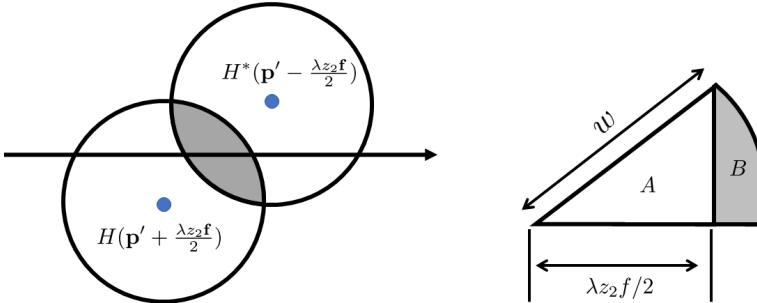


Figure 2.16: The optical transfer function $\mathcal{H}(\mathbf{f})$ is the autocorrelation function of $H(\mathbf{f})$ evaluated at position given by the two circles. This translates to computing the ratio between the overlapping area and the total area.

For a circular aperture, the amplitude transfer function is

$$H(\mathbf{f}) = P(\lambda z_2 \mathbf{f}). \quad (2.86)$$

2.6. SPACE-VARIANT SYSTEMS

Substituting into Equation (2.85), we have

$$\mathcal{H}(\mathbf{f}) = \frac{P(\lambda z_2 \mathbf{f}) \circledast P(\lambda z_2 \mathbf{f})}{P(0, 0) \circledast P(0, 0)}, \quad (2.87)$$

which we can recognize as simply an autocorrelation of the scaled pupil. The overlapping region can be computed by referring to Figure 2.16. The details for the evaluation of the circular OTF may be found in Goodman [11]. Using geometry, it can be shown that

$$\mathcal{H}(f) = \begin{cases} \frac{2}{\pi} \left[\arccos\left(\frac{f}{2f_0}\right) - \frac{f}{2f_0} \sqrt{1 - \left(\frac{f}{2f_0}\right)^2} \right], & f \leq 2f_0, \\ 0, & \text{otherwise,} \end{cases} \quad (2.88)$$

where we defined the cutoff frequency

$$f_0 = \frac{w}{\lambda z_i}, \quad (2.89)$$

where here f_0 corresponds to the cutoff frequency of the coherent system of the same aperture, meaning the incoherent version of the same system will have a frequency cutoff that is twice the size. The OTF obtained in Equation (2.88) is important, as it is the OTF of an incoherent **diffraction-limited** system. Assuming a perfect lens without distortions such as defocus, the system has a fundamental limit due to diffraction.

2.6 Space-Variant Systems

Thus far we have discussed optical systems with a single invariant point spread function $|h(\mathbf{x})|^2$. This, of course, is in line with the standard signals and systems view of LSI systems. As we progress towards imaging through turbulence, we will no longer be able to justify this assumption of spatial invariance; the situations we will encounter may even have a different PSF (alternatively, an impulse response) for virtually *every location in the image*. These differences will be correlated in space but may be wildly varying across the entire field of view. Therefore, we must re-examine our view of convolution to include spatially varying systems correctly.

2.6.1 Two Models for Convolution

Convolution between an input $J(\mathbf{x})$ and an impulse response $h(\mathbf{x})$ is often written as

$$I(\mathbf{x}) = J(\mathbf{x}) \circledast h(\mathbf{x}) = \int J(\mathbf{u})h(\mathbf{x} - \mathbf{u})d\mathbf{u}. \quad (2.90)$$

This is a simplification of a more general equation. The convolution equation is a special case of the **superposition integral**

$$I(\mathbf{x}) = \int h(\mathbf{x}, \mathbf{u})J(\mathbf{u})d\mathbf{u}. \quad (2.91)$$

If the system is space-invariant, the impulse response can be simplified to $h(\mathbf{x}, \mathbf{u}) = h(\mathbf{x} - \mathbf{u})$.

The superposition integral represents a more general framework for describing linear systems, of which the LSI systems are a subset. The LSI system particularly lends itself to the interpretation offered by Oppenheim and Willsky [12] in which the impulse response is flipped, shifted, then multiplied and integrated. This is a literal interpretation of (2.90). The alternative to this is to interpret convolution through (2.91) which suggests the response is a weighted summation of impulse responses centered at each \mathbf{u} .

We now introduce the possibility for the impulse response to vary as a function of position. However, we will choose two specific parameterizations, one in which the impulse response is parameterized by \mathbf{x} , and the other where it is parameterized by \mathbf{u} . We first consider the impulse response to vary as a function of \mathbf{u} , leading us to write Equation (2.91) as

$$I(\mathbf{x}) = (J \circledast h_{\mathbf{u}})(\mathbf{x}) = \int J(\mathbf{u})h_{\mathbf{u}}(\mathbf{x} - \mathbf{u})d\mathbf{u}, \quad (2.92)$$

This implies $h(\mathbf{x}, \mathbf{u}) = h_{\mathbf{u}}(\mathbf{x} - \mathbf{u})$. The alternative form of convolution rests upon the shift-and-integrate algorithm towards convolution often presented in a signals and systems course. The output at the coordinate \mathbf{x} will be dictated by the impulse response at \mathbf{x} . Following the flip and shift principle, this leads us to write

$$I(\mathbf{x}) = (J \circledast h_{\mathbf{x}})(\mathbf{x}) = \int J(\mathbf{u})h_{\mathbf{x}}(\mathbf{x} - \mathbf{u})d\mathbf{u} \quad (2.93)$$

This implies $h(\mathbf{x}, \mathbf{u}) = h_{\mathbf{x}}(\mathbf{x} - \mathbf{u})$.

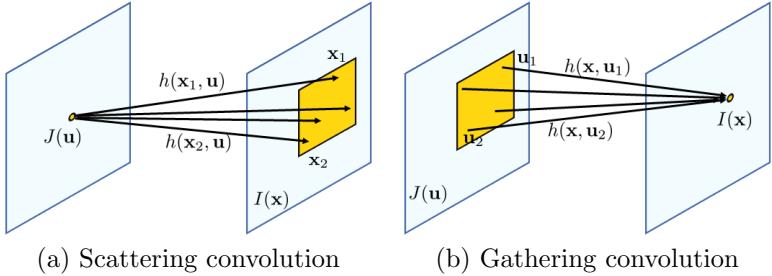


Figure 2.17: A visualization of the difference between scattering and gathering convolutions. Source: [15].

Note that (2.92) writes the impulse response as a function of the *source* location. That is, the location of an impulse response in the input signal will dictate the form of the impulse response as suggested by $h_{\mathbf{u}}$. Equation (2.93) instead parameterizes the impulse response by the *receiver* location $h_{\mathbf{x}}$. This leads us to refer to the two forms of convolution in the following way: we will refer to (2.92) as **scattering convolution** and (2.93) as **gathering convolution**. We visualize this difference in interpretation in Figure 2.17.

To provide some insight into the terminology used for these two forms of convolution, first consider a signal which is comprised of a single impulse response located at \mathbf{u} . As \mathbf{x} is varied, because of the fact that $I(\mathbf{x}) = h_{\mathbf{u}}(\mathbf{x} - \mathbf{u})$, the output is simply the impulse response $h_{\mathbf{u}}(\mathbf{x})$ centered at \mathbf{u} . Thus, it tells us how the impulse response located at \mathbf{u} *spread* to the neighboring points. Thus, the energy has been scattered from one point to many.

The alternative is to suppose we are interested in a receiver location \mathbf{x} . This fixes our impulse response as $h_{\mathbf{x}}$. However, note that this is already what (2.93) is doing; $h_{\mathbf{x}}$ is fixed in this integral. To compute the output, the impulse response, parameterized by source location \mathbf{x} is flipped and shifted \mathbf{x} , then multiplied and integrated by the signal $J(\mathbf{u})$. Furthermore, through integration, the convolution takes the energy that exists within the input signal, multiplies and integrates over it, and places it in one location at \mathbf{x} . Because of the fact the energy goes from many locations to one, it is referred to as gathering convolution.

To contrast the two forms in a different way, consider the input signal to be an impulse response $J(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{u})$. The output by

CHAPTER 2. IMAGE FORMATION MODEL

scattering would be

$$I(\mathbf{x}) = h_{\mathbf{u}}(\mathbf{x} - \mathbf{u}),$$

whereas the output by gathering would be

$$I(\mathbf{x}) = h_{\mathbf{x}}(\mathbf{x} - \mathbf{u}).$$

Due to the fact that \mathbf{x} need not equal \mathbf{u} , the two responses may be different. Furthermore, the gathering result is harder to interpret conceptually as the impulse response changes with each \mathbf{x} , where in the case of scattering the impulse response is fixed as $h_{\mathbf{u}}$. With this book focusing on imaging through turbulence, a system that varies spatially, we need to determine which one is appropriate for our purposes.

2.6.2 Which Model is Correct for Imaging?

The short answer is that for modeling the wave propagation phenomenon, the scattering equation is the correct model. Here are some elaborated answers.

As we showed, the incoherent imaging system is linear in intensity, and thus far we have further assumed it to be spatially invariant. If we relax this assumption of spatial invariance, we must rely on the superposition integral to relate the input and output of a general system,

$$I(\mathbf{x}) = \int h(\mathbf{x}, \mathbf{u}) J(\mathbf{u}) d\mathbf{u}.$$

This leaves us with a choice in determining how to choose $h(\mathbf{x}, \mathbf{u})$. Specifically, should $h(\mathbf{x}, \mathbf{u})$ be parameterized by the *source* or *receiver* variable? This choice is analogous to the case of deciding whether or not to perform scattering or gathering convolution [15], with parameterization by source location \mathbf{u} corresponding to scattering and receiver location \mathbf{x} to gathering.

One way this question can be answered is through the terminology used by the optics community. Instead of using the term impulse response, filter, or kernel when discussing Fourier Optics, we have opted to use the term “point spread function”. This term emphasizes the scattering model. The point spread function determines how a *point* (or impulse) will *spread* across the sensor plane (or, more generally, the surface of interest). The point spread function may change as a function of point location, but it will still describe how a particular point will spread across the sensor plane. It is also suggested through the

2.7. SUMMARY

visualization of the two in Figure 2.17 that this matches our intuition with image formation; a point source located at a position will spread energy across the focal plane via propagation.

These concepts can be demonstrated more mathematically and through numerical experiments. With the current background, the introduction of these concepts would be somewhat challenging. Therefore, we would suggest the interested reader to a paper by Chimitt et al. [15] which provides the proof and helpful visualizations comparing the two forms of convolution and showcases why scattering is the proper choice in modeling imaging.

To tie the end of this Chapter with the beginning, the response of a linear, space-invariant imaging system is given as

$$I_i(\mathbf{x}) = |\mathfrak{F}\text{ourier}\{P(\boldsymbol{\xi})e^{j\phi(\boldsymbol{\xi})}\}|^2 \circledast I_g(\mathbf{x}). \quad (2.94)$$

For the rest of this book, we will often allow the PSF to vary spatially through the following equation,

$$I_i(\mathbf{x}) = \left(|\mathfrak{F}\text{ourier}\{P(\boldsymbol{\xi})e^{j\phi_{\mathbf{u}}(\boldsymbol{\xi})}\}|^2 \overset{\mathbf{u}}{\circledast} I_g \right) (\mathbf{x}). \quad (2.95)$$

This is a special class of PSFs that are parameterized by their phase error $\phi_{\mathbf{u}}$, which we allow to be spatially varying as a function of point source location \mathbf{u} . Our notation here is used to make the operation clear. First, the phase is defined by $\boldsymbol{\xi}$ which spans the aperture plane. From this, the PSF is formed and, after being parameterized by coordinate $\mathbf{x}/(\lambda z)$, applied to I_g . This scaling of coordinate \mathbf{x} will typically be left off for simplicity in notation. We additionally emphasize the fact that the convolution is spatially varying with dependence on the *source* location by the notation above.

2.7 Summary

This Chapter presented Fourier optics combined with the thin lens model as well as the image formation process and spatially varying convolution. We summarize these as:

Part 1 Wave equation: We started by showing a class of waves that satisfy the scalar wave equation $\nabla^2 u(\mathbf{x}, t) = \frac{n^2}{c^2} \frac{\partial^2 u(\mathbf{x}, t)}{\partial t^2}$. If we further assume that $u(\mathbf{x}, t) = U(\mathbf{x}) \exp\{-j2\pi\nu t\}$, then the waves should satisfy the Helmholtz equation: $\nabla^2 U(\mathbf{x}) + k^2 U(\mathbf{x}) = 0$. All our subsequent discussions are about waves that satisfy the Helmholtz equation.

CHAPTER 2. IMAGE FORMATION MODEL

Part 2 Diffraction: We used the Huygens-Fresnel principle to model how waves move from one point to another point. Based on HFP, we then show the Rayleigh-Sommerfeld diffraction integral equation: $U(\mathbf{x}) = \frac{1}{j\lambda} \int_{\Sigma} U(\boldsymbol{\xi}) \frac{\exp\{jkr\}}{r} \cos \theta d\boldsymbol{\xi}$. The Rayleigh-Sommerfeld equation is general, and so we introduce two approximations. The first approximation leads to the Fresnel diffraction. With a second approximation, the Fresnel diffraction is simplified to the Fraunhofer diffraction.

Part 3 Lens: All imaging systems use a lens with a finite aperture. The lens is equivalent to applying a convolutional kernel $|h(\mathbf{x})|^2$ to the field $U(\mathbf{x})$. The convolutional kernel $|h(\mathbf{x})|^2$ is the Fourier transform of the pupil function. For example, if the lens is circular, then $|h(\mathbf{x})|^2$ is the Bessel function (or the Airy disc). The kernel $h(\mathbf{x})$ is also known as the point spread function. In the absence of any distortion, the resulting field will have an intrinsic resolution limit determined by the width of the $|h(\mathbf{x})|^2$.

Part 4 Image Formation: We are mostly interested in two types of sources: coherent and incoherent. For coherent light, the image (not the field) is $I_i(\mathbf{x}) = |h(\mathbf{x}) \circledast U_g(\mathbf{x})|^2$. For incoherent light, the image is $I_i(\mathbf{x}) = |h(\mathbf{x})|^2 \circledast I_g(\mathbf{x})$. Unless we use a laser source, typical long-range imaging through turbulence using a conventional camera is incoherent. The amplitude transfer function is $H(\mathbf{f}) = \mathcal{F}\text{ourier}\{h(\mathbf{x})\}$. The optical transfer function is $\mathcal{H}(\mathbf{f}) = \mathcal{F}\text{ourier}\{|h(\mathbf{x})|^2\}$.

Part 5 Spatially Varying Convolution: A set of tools for representing spatially varying convolution were presented along with the notation $h_{\mathbf{u}}(\mathbf{x}) \circledast^{\mathbf{u}} J(\mathbf{x})$. This is based on the perspective of superposition, with the chosen model for imaging as scattering convolution.

3. Modeling Turbulence

The aim of this Chapter is to present the statistical model for imaging through the atmosphere. This begins with first understanding the statistical model for the atmosphere's index of refraction. The model adopted here is that the atmosphere's index of refraction is a random, **turbulent** process that changes according to altitude, temperature, pressure, wind, humidity, and other factors. This Chapter will introduce one of the more prevalent models, how it fits into our understanding of image formation, and how we may simulate turbulent effects on an image.

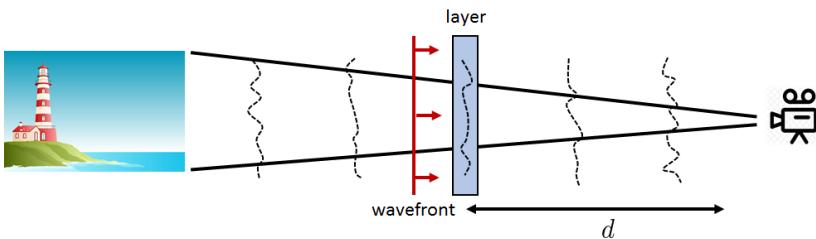


Figure 3.1: A schematic diagram illustrating the partition of a horizontal optical path into many small segments. In this Chapter, we will analyze the statistical properties of the layers and their impact on waves propagating through them.

Consider the imaging situation presented in Figure 3.1, in which a camera is placed a long distance away from an object (for our purposes, 500 meters or more will be suitable). In this situation, a wave that would have propagated as a planar wave in a vacuum is instead distorted by the atmosphere. These distortions are analogous to the case of a thin lens, however, instead of the thickness of the lens being

CHAPTER 3. MODELING TURBULENCE

carefully chosen to produce a converging, focused wave, the atmosphere generates a low quality “dented” wavefront. This denting of the wavefront will affect the image formation process by injecting a phase error into our system. We start with an overview to give the reader a sense of what these effects look like on an image and follow this with a brief summary of the rich history of the subject.

3.1 Overview and History

3.1.1 The Image Formation Model

In Chapter 2, we introduced the spatially invariant optical system’s image formation equation,

$$I_i(\mathbf{x}) = |h(\mathbf{x})|^2 \circledast I_g(\mathbf{x}), \quad (3.1)$$

which relates the observed image $I_i(\mathbf{x})$ to the geometric image $I_g(\mathbf{x})$ through the application of a spatially invariant PSF $|h(\mathbf{x})|^2$,

$$|h(\mathbf{x})|^2 = \left| \mathfrak{F}\text{ourier} \{ P(\xi) e^{j\phi(\xi)} \} \right|_{\mathbf{f}=\mathbf{x}/(\lambda z)}^2. \quad (3.2)$$

This particular PSF is described both by the aperture $P(\xi)$ and phase error $\phi(\xi)$. In Chapter 2, we presented an example where $\phi(\xi)$ described the focusing error by an imaging system. In this Chapter, we inject the turbulence model into $\phi(\xi)$. Therefore, if we have a model for $\phi(\xi)$, our efforts from the previous Chapter will be put to good use.

This leads us to encounter one of our first problems: the assumption of spatial invariance does *not* hold true in the case of observations through the atmosphere. Figure 3.2 illustrates this effect, where the degradations appear to vary from point to point. This suggests a **spatially variant** imaging model, a point we will visit more appropriately in later portions of the Chapter. With what we have described so far, ϕ is where the turbulence will be inserted. Therefore, we must allow ϕ to vary as a function of position \mathbf{u} in the object plane. This leads us to write the PSF equation as

$$|h_{\mathbf{u}}(\mathbf{x})|^2 = \left| \mathfrak{F}\text{ourier} \{ P(\xi) e^{j\phi_{\mathbf{u}}(\xi)} \} \right|_{\mathbf{f}=\mathbf{x}/(\lambda z)}^2. \quad (3.3)$$

along with the image formation equation for a spatially variant system to be

$$I_i(\mathbf{x}) = \left(|h_{\mathbf{u}}|^2 \circledast I_g \right) (\mathbf{x}). \quad (3.4)$$

3.1. OVERVIEW AND HISTORY

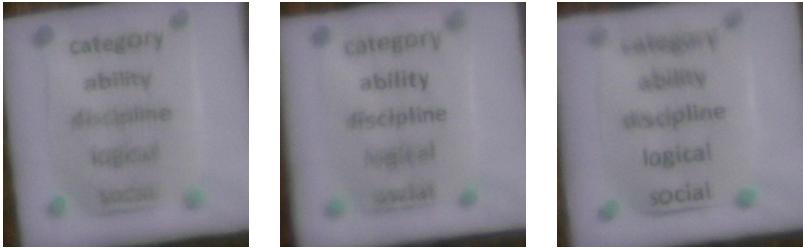


Figure 3.2: Three images of a board with printed text taken over a long distance (approximately 500 meters). Note the spatially varying effects, some words are blurred while others are legible.

Here, we denote \mathbf{u} as the input or source location and \mathbf{x} as the output or receiver location.

It is important to note that Equation (3.3) is an approximation. Throughout this Chapter, and in fact the entire book, we will rely upon the assumption as follows:

Assumption. We assume that the turbulent distortions can be adequately summarized by the phase distortions.

The assumption means that the amplitude distortions are negligible. If we take into account the amplitude distortions, a more general PSF equation would be

$$|h_{\mathbf{u}}(\mathbf{x})|^2 = \left| \mathfrak{F} \text{ourier}\{A_{\mathbf{u}}(\xi)P(\xi)e^{j\phi_{\mathbf{u}}(\xi)}\} \right|_{\mathbf{f}=\mathbf{x}/(\lambda z)}^2, \quad (3.5)$$

where $A_{\mathbf{u}}(\xi)$ is a spatially varying amplitude component which also varies per location in the object plane. Our assumption enables us to describe levels of turbulence that fall in the weak to moderate levels of distortions. We can justify this in two ways: (1) The assumption of phase-exclusive distortions is common to many varieties of simulation and turbulent imaging literature; (2) the reconstruction of wide FOV incoherent images through strong turbulence is ill-posed. We view (2) as the proper justification for the purposes of this book; the problems we want to solve will be challenging even in this weak to moderate range. Therefore, the lack of generality will not cost us too greatly. We will comment more directly on the limitations of this model when the proper background has been introduced.

3.1.2 Characteristics of Turbulence-Distorted Images

At this point, a reader who is familiar with the image processing literature may see the images in Figure 3.2 and feel the need to jump to the conclusion that this is a standard deblurring problem. We emphasize that this is not the case. While Chapter 5 will provide a thorough discussion of the algorithms for these types of distortions, we wish to describe a few key characteristics of the problem here. This should hopefully convey the difficulties of reconstruction as well as give a sense as to why such a careful description of the atmospheric model is needed.

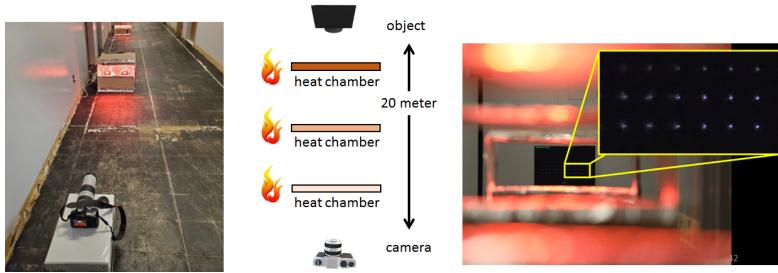


Figure 3.3: A physical setup of observing the PSF through a series of heat chambers. On the monitor, we display a grid of points. As light passes through the heat chambers, the image captured by the camera will suffer from turbulence.

Turbulent images are comprised of two key effects: pixel-shifting and blur. Both of these effects will vary per pixel. We provide a visualization of this in Figure 3.3. We further note that a blur kernel $|h_{\mathbf{u}}|^2$ has no closed form expression in the **image space**. So, we will need to describe it exclusively by $\phi_{\mathbf{u}}$. Commonly used parametric models such as the Gaussian blurs are not appropriate to be the **prior model** because the turbulence blur kernels are not even symmetric. Note that the point sources on the screen are very small near the bottom right of the zoomed in portion while the points near the upper left are wider and more complicated, once again highlighting the spatially varying nature of the problem.

One must also consider the fact that the atmosphere changes over time. This means that the degradation will fluctuate between multiple frames. This has a significant impact on reconstruction approaches, influencing the notion of **lucky imaging**. Lucky imaging rests upon the fact that due to the atmosphere changing in time, every once in

3.1. OVERVIEW AND HISTORY

a while one will observe a frame *or region* which has minimal degradation. Furthermore, pixel-shifting can be so strong that for a video with a moving object, it can be difficult to algorithmically determine if a motion is due to object motion or turbulence. All of these degradations compound to create a problem that requires the incorporation of temporal information, lucky imaging methodologies, deblurring, and un-warping.

3.1.3 Historical Developments

The theory of atmospheric modeling that we will be discussing in this book arises from statistical models of turbulence. The modern form of statistics and probability can be traced back to Kolmogorov's axiomatic approach to probability [16] (originally published in 1933). Amazingly, a core insight giving rise to the dominant form of modern turbulence theory is *also* a result of Kolmogorov's work [17, 18] in which he proposed a statistical model which describes how turbulence transfers energy down from the large fluctuations to the small fluctuations of the medium. Kolmogorov and a student of his, Obukhov, published additional papers on this subject (such as [19, 20, 21]), further elaborating on various elements of the model.

These insights alone are not enough to bring us to where the field stands today. There is a big gap missing – the works of Obukhov and Kolmogorov just model the turbulence. The question as to what happens to an optical wave propagating through the medium is not addressed. Continuing from the previous lineage, a student of Obukhov by the name of Tatarskii (alternatively spelled Tatarski in some sources) did exactly this in the form of an excellent manuscript [22] (the manuscript being a converted version of his Ph.D. thesis!). In this book, Tatarskii carefully applied the Kolmogorov theory within Maxwell's equations, resulting in the foundation that so many works derive from in the field of optical turbulence. There is little denying that the work of Tatarskii is likely *the* foundation of the field.

From here, a wide range of applications emerged¹. One of the more prominent researchers from this era (1960s - 1980s) is Fried. Fried published a vast amount of papers over a long career, where we highlight

¹As we write this book, we recognize our limitations of not being trained in a traditional physics manner. It is therefore sometimes hard for us to appreciate the raw physics that has emerged from Tatarskii's work. However, on the applications side, especially towards generating large-scale datasets, we hope to share our experience with the readers.

CHAPTER 3. MODELING TURBULENCE

the following papers which were particularly helpful when we started to work on this topic: [23, 24, 25, 26, 27]. Fried proposed the **coherence diameter** r_0 , which will be of significant importance for the latter majority of this book and is often referred to as the **Fried parameter**. Of course, we will return to its definition once the proper preliminaries have been presented.

Aside from Fried, there are others such as Ishimaru [28, 29, 30] who worked with Tatarskii and wrote his own book on the subject. Greenwood contributed to adaptive optics in the presence of turbulence, with the **Greenwood frequency** attributed to his name which quantifies aspects of the temporal evolution of the atmosphere within the context of adaptive correction [31]. Hufnagel [32, 33] contributed to what is now regarded as the Hufnagel-Valley turbulence path model, a model we will introduce later in this Chapter. Roggemann and Welsh [34, 35, 36] have both published numerous works within the field and an influential book that offers a thorough explanation of the necessary turbulent optics which is arguably more accessible to the average reader than previous texts. Additionally, there are those such as Tyson [37, 38] particularly oriented in the direction of astronomical imaging and adaptive optics.

With some of our admitted subjectivity, there is one paper that stands out among the rest. This is the paper by Noll [39] which may very well be our favorite paper on the topic. Noll proposes a basis expansion for the phase distortion (Fried also did this almost 10 years earlier [23], however, Noll's has emerged as the more dominant approach). We would also pair this endorsement of Noll's paper with Roddier [40] which offers a slight correction to Noll's paper and additional exposition, among other contributions. This basis expansion will solve the issue regarding the phase distortion ϕ 's description in a simplistic manner. Noll's paper is so influential to the authors, in fact, that virtually the entire simulation approach proposed by us and collaborators ([41, 42, 43]) is inspired by Noll's work. This paper, and our various modifications and interpretation, will be the core of the discussion in Chapter 4.

This Chapter will follow the same chronology as the historical development described here. We will first introduce the Kolmogorov model and related models. From here, we will apply it to optical propagation through the atmosphere (following the more simplistic approach along the lines of Roggemann and Welsh [34]) following various applications. Finally, we will close the Chapter with the classical approach to simulating these effects in an imaging system known as **split-step**

3.2. THE ATMOSPHERE'S INDEX OF REFRACTION

propagation, often referred to as simply “split-step”.

3.2 The Atmosphere's Index of Refraction

A key component to the theory we are to discuss is that the atmospheric index of refraction (which we will typically refer to as simply the atmosphere) is distributed randomly in a turbulent fashion. A reasonable question may then be: why is the atmosphere turbulent? While the true reasoning is well beyond the scope of this book, we feel it important to give some perspective on the concept of turbulence. The atmosphere can be modeled as a fluid. Fluid is said to move in either a laminar or turbulent fashion. If we claim the atmosphere moves in a laminar way, we must be prepared to claim that all particles of the atmosphere are moving in the same direction without any deviation. This clearly cannot be the case, for example, the wind is not blowing in the same direction everywhere in the world at all times. By virtue of the atmosphere not moving as laminar flow, we may then reason that atmospheric motion is turbulent.

We emphasize this point for a key reason: turbulence does not have to be “aggressive”, it can be slow moving and calm such as the smoke that dances above a candle after being blown out. Furthermore, the effect the atmosphere has on an imaging system is not something that only happens *sometimes*. Rather, it is happening all the time and is only easily observable in certain circumstances, such as ground-to-ground imaging over a long distance.

3.2.1 Kolmogorov Power Spectral Density

The statistics of a **random process** X is often characterized by studying the spatial correlation between $X(\mathbf{r}_1)$ and $X(\mathbf{r}_2)$ where $\mathbf{r}_1 \in \mathbb{R}^3$ and $\mathbf{r}_2 \in \mathbb{R}^3$ are two spatial coordinates. We will denote correlation function of X as $\Gamma_X(\mathbf{r}_1, \mathbf{r}_2) = \mathbb{E}[X(\mathbf{r}_1)X(\mathbf{r}_2)]$. A frequently used assumption for simplifying a random process is to assume it is **homogeneous**. (In statistical signal processing, we call it **wide sense stationary**.) For a spatially varying random process, the correlation function of a homogeneous process takes the form of

$$\Gamma_X(\mathbf{r}) = \mathbb{E}[X(\mathbf{r}_1)X(\mathbf{r}_1 - \mathbf{r})], \quad (3.6)$$

where $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$ is the spatial increment. In other words, homogeneity implies the autocorrelation function can be written as a function of the

CHAPTER 3. MODELING TURBULENCE

difference $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$ instead of the pair of absolute locations $(\mathbf{r}_1, \mathbf{r}_2)$.

An alternative to the correlation function is the **power spectral density** (PSD). For a homogeneous random process, the PSD is the Fourier transform of the autocorrelation function by the **Wiener–Khinchin theorem**:

$$\Phi_X(\mathbf{k}) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \Gamma_X(\mathbf{r}) e^{j\mathbf{k}^T \mathbf{r}} d\mathbf{r}, \quad (3.7)$$

where $\mathbf{k} = [k_x, k_y, k_z]$ is the spatial **wavenumber** vector. The scalar wavenumber is $k = |\mathbf{k}| = \sqrt{k_x^2 + k_y^2 + k_z^2}$.

With these concepts defined, we now turn to the fluctuations in the atmosphere. Our discussion of the index of refraction follows the one by Goodman [44]. We will model the atmosphere's index of refraction as a function of spatial location $\mathbf{r} = [x, y, z]$, time t and wavelength λ :

$$n(\mathbf{r}, t, \lambda) = \underbrace{n_0(\mathbf{r}, \lambda)}_{\text{mean index}} + \underbrace{n_1(\mathbf{r}, t)}_{\text{fluctuation}}. \quad (3.8)$$

The mean refractive index $n_0(\mathbf{r}, \lambda)$ can be assumed time *independent* and may change with elevation or climate. The dependency of the wavelength is ignored in $n_1(\mathbf{r}, t)$ because it is generally weakly varying in the visible spectrum. We will assume that the index of refraction $n_1(\mathbf{r})$ is a zero-mean Gaussian process so that $\mathbb{E}[n_1(\mathbf{r})] = 0$ for any \mathbf{r} [44, 22] and denote its autocorrelation function as $\Gamma_n(\mathbf{r}_1, \mathbf{r}_2) = \mathbb{E}[n_1(\mathbf{r}_1)n_1(\mathbf{r}_2)]$.

This leads us to introduce the Kolmogorov model. The Kolmogorov model divides the fluctuations in n_1 into three regimes which are specified by the **outer scale** L_0 and the **inner scale** ℓ_0 . Scales can be thought of as structures that are *on the order of* the specified size. The Kolmogorov model specifically models the energy behavior of the turbulent fluctuations of sizes that are between the outer and inner scales. Thus, for wavenumbers $2\pi/L_0 \leq |\mathbf{k}| \leq 2\pi/\ell_0$, $\Phi_n(\mathbf{k})$ follows the Kolmogorov PSD [22, 17, 21]:

$$\Phi_n(\mathbf{k}) = 0.033C_n^2|\mathbf{k}|^{-11/3}, \quad (3.9)$$

where C_n^2 is called **structure constant** of the index of refraction with units of $[m^{-2/3}]$. The Kolmogorov PSD is shown in Figure 3.4, along with a modified version known as the von Kármán PSD spectrum [45, 34]:

$$\Phi_n(\mathbf{k}) = \frac{0.033C_n^2}{(|\mathbf{k}|^2 + k_0^2)^{11/6}} \exp\left\{-\frac{|\mathbf{k}|^2}{k_m^2}\right\}, \quad (3.10)$$

3.2. THE ATMOSPHERE'S INDEX OF REFRACTION

where $k_0 = 2\pi/L_0$ and $k_m = 5.92/l_0$. The fluctuations are better modeled by the von Kármán PSD in the case of small wavenumbers \mathbf{k} . For a thorough discussion of the varying PSDs (both Kolmogorov and non-Kolmogorov turbulence), we suggest the reader to Korotkova [46].

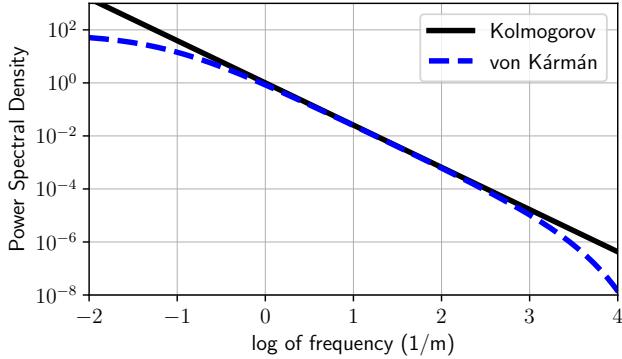


Figure 3.4: Kolmogorov power spectral density and the von Kármán power spectral density, plotted as a function of the scalar wavenumber $k = |\mathbf{k}|$.

3.2.2 Structure Constant C_n^2

The Kolmogorov spectrum presented in Equation (3.9) introduces the **structure constant** C_n^2 . The value of C_n^2 is an important parameter for modeling the turbulence strength. We can think of the structure constant as being analogous to the variance of a random variable. A higher value of C_n^2 suggests a greater strength of turbulence. C_n^2 is related to the altitude, location, time of day, etc. We will typically allow C_n^2 to vary over the path of propagation, thus typically referring to it as $C_n^2(z)$, which will call a **turbulence profile**.

In the literature, there exist many different models of C_n^2 . For example, the Hufnagel-Valley profile [33, 47, 48, 34] models C_n^2 as a function of the altitude h using

$$C_n^2(h) = 5.94 \times 10^{-53} (v/27)^2 h^{10} e^{-h/1000} + 2.7 \times 10^{-16} e^{-h/1500} + A e^{-h/100},$$

CHAPTER 3. MODELING TURBULENCE

where the typical value of A is $A = 1.7 \times 10^{-14} [\text{m}^{-2/3}]$ and the value of v is $v = 21 [\text{m/s}]$. The Submarine Laser Communication Day (SLCD) profile [48, 34] states that

$$C_n^2(h) = \begin{cases} 0, & 0 [\text{m}] < h < 19[\text{m}], \\ 4.008 \times 10^{-13} h^{-1.054}, & 19 [\text{m}] < h < 230 [\text{m}], \\ 1.3 \times 10^{-15}, & 230 [\text{m}] < h < 850 [\text{m}], \\ 6.352 \times 10^{-7} h^{-2.966}, & 850 [\text{m}] < h < 7000 [\text{m}], \\ 6.209 \times 10^{-18} h^{-0.6229}, & 7000 [\text{m}] < h < 20000 [\text{m}]. \end{cases}$$

For ground-to-ground imaging, it may be reasonable to assume a constant C_n^2 profile for simplicity, though it may certainly vary depending on many factors (e.g. a heat source along the path).

Figure 3.5 shows the C_n^2 profile according to the Hufnagel-Valley model. C_n^2 is generally large when the altitude is below 20 [m] and it starts to decrease sharply as the altitude increases. One explanation is that the ground has a higher temperature than air at a higher altitude. For ground-to-ground imaging, the range of C_n^2 is typically around $10^{-13} [\text{m}^{-2/3}]$ (for strong turbulence) to $10^{-17} [\text{m}^{-2/3}]$ (for weak turbulence).

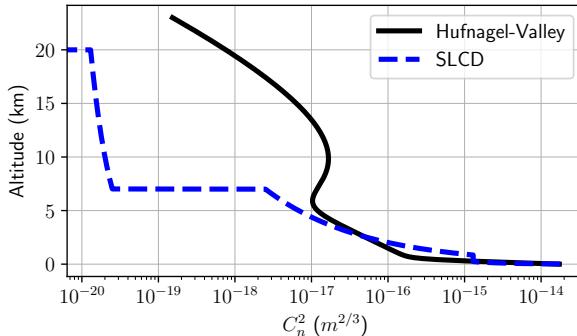


Figure 3.5: Profile of the C_n^2 as a function of the altitude, where we are comparing the SLCD profile vs. the Hufnagel-Valley profile.

3.2.3 The Structure Function $\mathcal{D}_n(\mathbf{r})$

Examining Equation (3.9), we notice a particular difficulty, namely the singularity when $\mathbf{k} = \mathbf{0}$. This will cause an issue for the evaluation of

3.2. THE ATMOSPHERE'S INDEX OF REFRACTION

the correlation function by the Wiener–Khinchin theorem,

$$\Gamma_n(\mathbf{r}) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} 0.033C_n^2|\mathbf{k}|^{-11/3}e^{-j\mathbf{k}^T\mathbf{r}}d\mathbf{k}. \quad (3.11)$$

This leads us to introduce a unique characteristic of the statistical turbulence literature, the **structure function**. The structure function is an alternative to the correlation function as defined in [17]:

$$\mathcal{D}_n(\mathbf{r}) = \mathbb{E}[(n_1(\mathbf{r}_1) - n_1(\mathbf{r}_1 - \mathbf{r}))^2]. \quad (3.12)$$

With the assumption of stationarity, we can show that $\mathcal{D}_n(\mathbf{r}) = 2[\Gamma_n(0) - \Gamma_n(\mathbf{r})]$ and hence [22, 34]

$$\begin{aligned} \mathcal{D}_n(\mathbf{r}) &= 2 \int [\Phi_n(\mathbf{k}) - \Phi_n(\mathbf{k})e^{-j\mathbf{k}^T\mathbf{r}}] d\mathbf{k} \\ &= 2 \int [1 - \cos(\mathbf{k}^T\mathbf{r})] \Phi_n(\mathbf{k}) d\mathbf{k}. \end{aligned} \quad (3.13)$$

The form of the structure function and PSD relationship aids in controlling the divergence near $\mathbf{k} = \mathbf{0}$. It has been shown that this integral converges for a set of cases to which the Kolmogorov PSD belongs [22].

Assuming that $n_1(\mathbf{r})$ is **isotropic** (spherically symmetric), and using the Kolmogorov PSD $\Phi_n(\mathbf{k})$, we can derive the following:

Definition 3.1 (Kolmogorov Structure Function of the Index of Refraction). Assuming the index of refraction is a homogeneous (spatially stationary) and isotropic (spherically symmetric) random process, the Kolmogorov structure function is defined as [22, 49, 34]

$$\mathcal{D}_n(\mathbf{r}) = C_n^2|\mathbf{r}|^{2/3}, \quad (3.14)$$

where C_n^2 is the structure constant of the index of refraction, and $\mathbf{r} = [x, y, z]$ is the coordinate in the 3D space.

The structure function is even well-defined even in the case of inhomogeneities. This is allowed by its invariance to the small wavenumber effects, which correspond to the large scale behavior. As a result of the definition of the structure function in Equation (3.13), the PSD will be attenuated near $|\mathbf{k}| = 0$. Thus, even if the variance of n_1 were to fluctuate over space, the structure function will appropriately mitigate the resulting inhomogeneities. The structure function, as a result,

CHAPTER 3. MODELING TURBULENCE

will be homogeneous and isotropic even in the case of inhomogeneous atmospheric turbulence (for more details on this topic, we suggest the reader refer to Tatarskii [22]).

3.3 Structure Function of the Phase

In Chapter 3.2, we discussed the statistics of the index of refraction. For the purposes of imaging, we are interested in the effects the index of refraction has on a wave. We will focus specifically on how the atmosphere affects the phase of a propagating wave. Our plan of development is to first derive the statistics of the phase for a single layer of homogeneous turbulence, as shown in Figure 3.6. We will then use this result for multiple layers, which will generalize our development to paths with varying C_n^2 profiles.

We wish to emphasize that the following analysis will be for a single point source. The extension to objects would only stand to complicate our approach at this time. We will introduce a way towards modeling the spatially varying nature towards the end of this Chapter regarding simulation. Chapter 3 will be dedicated to the image formation model more completely.

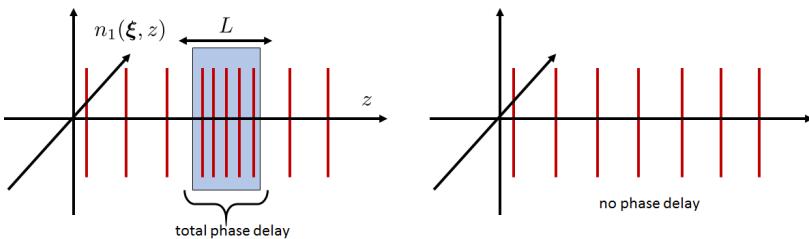


Figure 3.6: Phase lead and lag caused by a medium with a high refractive index. As light passes through the medium, the speed of the light is slowed down because the refractive index is higher than the vacuum. This causes a phase delay. The longer the propagation path, the stronger the phase distortion will be.

3.3.1 A Single Turbulence Layer

We will first begin with the propagation of a plane wave through a single layer of turbulence. It is first important to note our convention

3.3. STRUCTURE FUNCTION OF THE PHASE

of measuring propagation distance: we choose to measure the distance *from* the aperture plane, thus $z = L$ is in the object plane while $z = 0$ is in the aperture plane. For a propagation distance of L , we segment the atmosphere into M homogeneous segments partitioned by distances L_i . For simplicity, let us focus on the case of $z = 0$ to $z = L_1$. At this point, we will drop the subscript i as we are only focusing on one segment. We assume the turbulent medium of thickness L is homogeneous, isotropic, and distributed according to the Kolmogorov spectrum.

Decomposing $\mathbf{r} = [\boldsymbol{\xi}, z]$ where $\boldsymbol{\xi} = [x, y]$ is the coordinate in the 2D plane and z is the coordinate along the propagation path, the phase distortion caused by the medium with a propagation distance L follows the equation

$$\phi(\boldsymbol{\xi}, L) = k \int_0^L n_1(\boldsymbol{\xi}, z) dz, \quad (3.15)$$

where $k = |\mathbf{k}| = 2\pi/\lambda$ is the scalar wavenumber. Note that this medium of refractive index $n_1(\mathbf{r})$ will cause phase effects that vary across the wavefront.

Following a similar development to the thin lens, let us denote the incident wave before the atmospheric layer to be $U(\boldsymbol{\xi})$ and the output wave $U'(\boldsymbol{\xi})$. Our focus is then on the distribution of the emergent wave, $U'(\boldsymbol{\xi})$. For simplicity, let us assume that $U(\boldsymbol{\xi})$ is a plane wave with $U(\boldsymbol{\xi}) = 1$. Accordingly, the input-output relationship is given by

$$U'(\boldsymbol{\xi}) = t_\phi(\boldsymbol{\xi}), \quad (3.16)$$

with $t_\phi(\boldsymbol{\xi}) = e^{j\phi(\boldsymbol{\xi})}$. In general, the distortions will consist of not only the phase term but also an amplitude term. However, for simplicity in this book, we shall skip the discussion of the amplitude term.

As we are interested in the wave's phase structure function, we begin with consideration of the autocorrelation function:

$$\begin{aligned} \Gamma_{U'}(\boldsymbol{\xi}, \boldsymbol{\xi}') &= \mathbb{E}[t_\phi(\boldsymbol{\xi}) t_\phi^*(\boldsymbol{\xi}')] \\ &= \mathbb{E}[e^{j\phi(\boldsymbol{\xi})} e^{-j\phi(\boldsymbol{\xi}')}] \\ &\stackrel{(a)}{=} \mathbb{E}\left[\exp\left\{jk\left(\int_0^L [n_1(\boldsymbol{\xi}, z) - n_1(\boldsymbol{\xi}', z)] dz\right)\right\}\right], \end{aligned} \quad (3.17)$$

where $(\cdot)^*$ denotes the complex conjugate, where we've utilized substitution of Equation (3.15) in (a). At this point, we need to make an assumption about $n_1(\boldsymbol{\xi}, z)$ or otherwise we cannot proceed further:

Assumption 3.1 (Assumption of $n_1(\xi, z)$). We assume that the index of refraction $n_1(\xi, z)$ is a zero-mean Gaussian random process. Moreover, we assume that $n_1(\mathbf{r})$ is homogeneous and isotropic for the region contained within L .

The normality (Gaussian statistics) is partially due to convenience because it is theoretically more tractable. We assume that the random process is zero mean because the index of refraction $n(\xi, z)$ has a constant term $n_0(\xi, z)$ and a random term $n_1(\xi, z)$. The constant term can take care of the offset and so the random term can be zero mean. Our assumption of homogeneity and isotropy will again simplify our analysis. In the next subsection, we will move towards multiple layers, which will allow for large scale inhomogeneities.

With these assumptions in mind, we recall the following moment generating function property of a zero-mean Gaussian random variable.

Lemma 3.1 (Moment Generating Function). Let X be a zero-mean Gaussian random variable, and let s be any constant. Then,

$$\mathbb{E}[e^{sX}] = \exp\left\{\frac{s^2\sigma^2}{2}\right\}, \quad (3.18)$$

where $\sigma^2 = \text{Var}[X]$ is the variance of X .

Using the above lemma, it follows that $\mathbb{E}[e^{-jX}] = \exp\{-\sigma^2/2\}$. Therefore, we can show that

$$\Gamma_{t_\phi}(\xi, \xi') = \exp\left\{-\frac{1}{2}k^2\mathbb{E}\left[\left(\int_0^L [n_1(\xi, z) - n_1(\xi', z)] dz\right)^2\right]\right\}.$$

$$\mathcal{D}_\phi(\xi, \xi') \stackrel{\text{def}}{=} \mathbb{E}[(\phi(\xi) - \phi(\xi'))^2]$$

Notice that the usage of the moment generating function results in the structure function of the indices of refraction appearing in the exponential function. This term within the exponential function represents the structure function of the accumulated phase distortions and is accordingly referred to as the **phase structure function**.

3.3. STRUCTURE FUNCTION OF THE PHASE

Definition 3.2 (Phase Structure Function). Suppose light passes through a random medium of distance L with a fluctuating index of refraction $n_1(\xi, z)$. The structure function of the phase is defined as

$$\mathcal{D}_\phi(\xi, \xi') \stackrel{\text{def}}{=} k^2 \mathbb{E} \left[\left(\int_0^L [n_1(\xi, z) - n_1(\xi', z)] dz \right)^2 \right], \quad (3.19)$$

where the expectation is taken with respect to $n_1(\xi, z)$.

This serves as a useful result, however, there is a fair amount of simplification possible. Rewriting the square as a product of two integrals, we show that

$$\begin{aligned} \mathcal{D}_\phi(\xi, \xi') &= k^2 \mathbb{E} \left[\int_0^L \int_0^L [n_1(\xi, z) - n_1(\xi', z)] \right. \\ &\quad \times [n_1(\xi, z') - n_1(\xi', z')] dz dz' \left. \right]. \end{aligned}$$

Expanding the inner terms algebraically and the fact that $D_n(\xi, z) = 2[\Gamma_n(0, z) - \Gamma_n(\xi, z)]$, recognizing that $D_n(\xi, z) = D_n(-\xi, z)$ we can rewrite the above equation as

$$\begin{aligned} \mathcal{D}_\phi(\xi, \xi') &= k^2 \int_0^L \int_0^L \left[D_n(0, z - z') - \frac{1}{2} D(\xi - \xi', z - z') \right. \\ &\quad \left. - \frac{1}{2} D_n(\xi' - \xi, z - z') \right] dz dz' \\ &= -k^2 L \int_{-L}^L [D_n(0, z) - D_n(\xi - \xi', z)] \left(1 - \frac{|z|}{L} \right) dz. \end{aligned}$$

Substituting the Kolmogorov structure function of the refractive index $D_n(\xi, z) = D_n(\mathbf{r}) = C_n^2 |\mathbf{r}|^{2/3} = C_n^2 (|\xi|^2 + z^2)^{1/3}$ and using a key numerical observation by Fried [24], it follows that

$$\begin{aligned} \mathcal{D}_\phi(\xi, \xi') &= -k^2 L C_n^2 \int_{-L}^L \left[z^{2/3} - (|\xi - \xi'|^2 + z^2)^{1/3} \right] \left(1 - \frac{|z|}{L} \right) dz \\ &= 2.91 k^2 L C_n^2 |\xi - \xi'|^{5/3}. \end{aligned} \quad (3.20)$$

CHAPTER 3. MODELING TURBULENCE

This expression constitutes a key result: we can now statistically describe the phase of a wave after propagation through a turbulent medium defined by the Kolmogorov spectrum. Furthermore, we can see it is homogeneous and isotropic. In what shall follow, we will see that we can use this result, combined with an assumption, to generalize to multiple slices of turbulence.

3.3.2 Multiple Phase Screens

Suppose $C_n^2(z)$ changes appreciably along the path of propagation such as in the case of astronomical viewing. We will then require our previous result to be extended to cases beyond a homogeneous layer of turbulence. For propagation of this variety, there may exist inhomogeneities in the distribution. A more rigorous wave analysis is beyond the discussion of this book, though we would point the interested reader to [44, 22, 50, 46] or [7, Ch. 13]. We instead take a simpler approach described by [34] that will result in the same solutions as more complex methods, leading us to introduce the concept of the **layered model for turbulence** to model the atmosphere, which requires us to introduce an assumption:

Assumption. We assume that the turbulent layers are statistically independent of one another.

We first define our resultant phase as

$$\begin{aligned}\phi(\boldsymbol{\xi}) &= k \int_0^L n_1(\boldsymbol{\xi}, z) dz, \\ &= k \sum_{i=1}^{M-1} \int_{L_{i-1}}^{L_i} n_1(\boldsymbol{\xi}, z) dz.\end{aligned}\tag{3.21}$$

As previously, we are interested in the structure function of this phase realization. As a result of our assumption of phase screen independence,

3.3. STRUCTURE FUNCTION OF THE PHASE

we may write

$$\begin{aligned}
\mathcal{D}_\phi(\boldsymbol{\xi}, \boldsymbol{\xi}') &= k^2 \mathbb{E} \left[\left(\sum_{i=1}^{M-1} \int_{L_{i-1}}^{L_i} [n_1(\boldsymbol{\xi}, z) - n_1(\boldsymbol{\xi}', z)] dz \right)^2 \right], \\
&\stackrel{(a)}{=} k^2 \mathbb{E} \left[\sum_{i=1}^{M-1} \left(\int_{L_{i-1}}^{L_i} [n_1(\boldsymbol{\xi}, z) - n_1(\boldsymbol{\xi}', z)] dz \right)^2 \right], \\
&= k^2 2.91 k^2 |\boldsymbol{\xi} - \boldsymbol{\xi}'|^{5/3} \sum_{i=1}^{M-1} C_n^2[i] \Delta L_i,
\end{aligned} \tag{3.22}$$

where $\Delta L_i = L_{i+1} - L_i$ and (a) utilizes the independence of the atmospheric slices, thus dropping the cross terms in the squared summation. This result shows that the phase structure function is related to a weighted sum of the turbulence strength along the path of propagation.

Two additional extensions of this result will be of importance, the first of which is the extension of Equation (3.22) to a continuous integration which we give without proof,

$$\mathcal{D}_\phi(\boldsymbol{\xi}, \boldsymbol{\xi}') = 2.91 k^2 |\boldsymbol{\xi} - \boldsymbol{\xi}'|^{5/3} \int_0^L C_n^2(z) dz. \tag{3.23}$$

A second extension of our result will be to spherical waves. Thus far, we have only derived phase statistics as a result of plane wave propagation, which significantly simplified our analysis. The extension to spherical waves is done by the introduction of a term into the integrand,

$$\mathcal{D}_\phi(\boldsymbol{\xi}, \boldsymbol{\xi}') = 2.91 k^2 |\boldsymbol{\xi} - \boldsymbol{\xi}'|^{5/3} \int_0^L C_n^2(z) \left(\frac{L-z}{L} \right)^{5/3} dz. \tag{3.24}$$

We again emphasize our convention of defining $z = L$ to be in the object plane, thus, the turbulence near the object contributes less than the turbulence near the imaging system due to the term $(L-z)/L$.

3.3.3 Wave Structure Function

Thus far we have commented on the phase structure function. For our purposes, this will be a sufficient description and indeed our chosen model for the rest of the book. However, it is worthwhile to mention that the phase structure function is a *component* of the overall **wave structure function**.

Definition 3.3 (Wave Structure Function). *The wave structure function is defined to be [51]*

$$\mathcal{D}(\boldsymbol{\xi}, \boldsymbol{\xi}') = \mathcal{D}_l(\boldsymbol{\xi}, \boldsymbol{\xi}') + \mathcal{D}_\phi(\boldsymbol{\xi}, \boldsymbol{\xi}'), \quad (3.25)$$

where $\mathcal{D}_l(\boldsymbol{\xi}, \boldsymbol{\xi}')$ and $\mathcal{D}_\phi(\boldsymbol{\xi}, \boldsymbol{\xi}')$ are the structure functions of the **log-amplitude** and phase, respectively.

The wave structure function leads us to briefly introduce the log-amplitude. The log amplitude quantifies the variation of the *amplitude* of the wave (of course, the wave is comprised of phase and amplitude components.) In a paper by Fried [51], the log-amplitude is defined to be

$$l(\boldsymbol{\xi}) = \ln(A(\boldsymbol{\xi})/\bar{A}) \quad (3.26)$$

where $A(\boldsymbol{\xi})$ is the amplitude at a point \mathbf{x} and \bar{A} is the root mean square (RMS) value of $A(\boldsymbol{\xi})$. It is important to note that the log amplitude was studied in depth in Tatarskii's manuscript [22], though is more easily referenced in Fried's paper with Fried providing a bit of additional exposition on his choice in usage of \bar{A} in a footnote.

We will typically opt to describe only the phase structure function and not the wave structure function. The reason for this can be understood again by Fried [24] in which the following approximations are provided

$$\mathcal{D}_\phi(\boldsymbol{\xi}, \boldsymbol{\xi}') \approx \mathcal{D}(\boldsymbol{\xi}, \boldsymbol{\xi}') \quad D \gg (L\lambda)^{1/2}, \quad (3.27)$$

$$\mathcal{D}_\phi(\boldsymbol{\xi}, \boldsymbol{\xi}') \approx \frac{1}{2}\mathcal{D}(\boldsymbol{\xi}, \boldsymbol{\xi}') \quad D \ll (L\lambda)^{1/2}. \quad (3.28)$$

We refer to the case of (3.27) as the near field and (3.28) as the far field. This gives us *some* sense of how to interpret the fact that the phase structure is typically sufficient. In the near field, the wave structure function is approximately the wave structure function whereas in the far field, it is approximately half of the wave structure function. Thus, in the near field, the phase structure function is sufficient in describing the perturbations.

At this point, we wish to remind the reader that the developed theory thus far, and the one which we will apply throughout the book, is valid for only weak to moderate turbulence. This limitation is most precisely shown in the context of Maxwell's equations for inhomogeneous media, though it is well beyond the scope or intent of this book.

3.4. IMPORTANT APPLICATIONS OF THE MODEL

In Tatarskii's manuscript [22], the inhomogeneous Maxwell's equations were simplified via the **Rytov approximation**. The results which carry out from here are said to describe **weak fluctuations** and give rise to what has been presented thus far (weak fluctuations here refers to the amplitude fluctuations). Within this framework, one may analyze the amplitude behavior, also known as **scintillation**. For further details, we would refer the reader to Goodman [44] as a starting point, though Tatarskii discusses these directly in his manuscript [22] along with Ishimaru [28].

Though one may analyze scintillation in this context (for example, as done by Goodman [44]), there exist alternative approaches which utilize more complex mathematical concepts and provide more generality. One such approach is known as the **path integral formulation** (alternatively known as a Feynman path integral) [52]. The motivation for the path integral arose most famously from quantum mechanics, though the same methods have been applied to wave propagation through random media. These methods were applied to the case of turbulence in works from Tatarskii and collaborators [53, 54, 30, 55] along with others such as Dashen [56]. To our knowledge, this framework represents the most general approach to describing the problem to date. These methods seek to describe the regime in which **strong fluctuations** exist (i.e. strong amplitude fluctuations).

3.4 Important Applications of the Model

The results of Chapter 3.3 comprise some of the most significant results of turbulent imaging for our purposes. We now present a few general results such as the Fried parameter. After this, we move towards defining the isoplanatic angle, along with two optical transfer functions that describe various temporal aspects of imaging through the atmosphere.

3.4.1 Fried Parameter and Isoplanatic Angle

The atmospheric coherence diameter r_0 (or as we shall call it, the **Fried parameter**) is defined to be a measure related to the resolution of an imaging system. We begin with defining this important parameter introduced by Fried [23]:

CHAPTER 3. MODELING TURBULENCE

Definition 3.4 (Fried Parameter). Consider a turbulent medium with a structure constant of the index of refraction C_n^2 , and propagation distance L . For a plane wave incident upon turbulence, the Fried parameter is defined as

$$r_0 = 0.185 \left[\frac{4\pi^2}{k^2 \int_0^L C_n^2(z) dz} \right]^{3/5}, \quad (3.29)$$

and for a spherical wave:

$$r_0 = 0.185 \left[\frac{4\pi^2}{k^2 \int_0^L \left(\frac{L-z}{L} \right) C_n^2(z) dz} \right]^{3/5} \quad (3.30)$$

The Fried parameter can be interpreted as an aperture size that is imposed by the limits of atmospheric viewing (analogous to the numerical aperture in image resolution). It is inversely proportional to the turbulence strength C_n^2 and the path length L . A longer path length and stronger turbulence will give a smaller r_0 . This r_0 will then limit the resolution. If we define D as the aperture diameter, then D/r_0 will then tell us the optical resolution of the image observed through turbulence. Figure 3.7 shows a few examples.

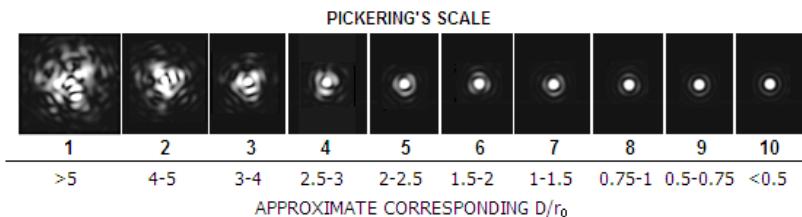


Figure 3.7: Impact of the D/r_0 to the observed point spread function. Here, D is the aperture diameter and r_0 is the Fried parameter. Source: https://eaae-astronomy.org/images/projects/catch-a-star/2015/18_How_to_measure_seeing.pdf

The Fried parameter is defined in careful accordance with the results of Equation (3.22). As a result, we may rewrite the structure function of the phase as follows:

3.4. IMPORTANT APPLICATIONS OF THE MODEL

Definition 3.5 (Kolmogorov Structure Function of the Phase).

Assuming that index of refraction follows the Kolmogorov structure function, then the structure function of the phase is

$$\mathcal{D}_\phi(\xi, \xi') = 6.88 \left(\frac{|\xi - \xi'|}{r_0} \right)^{5/3}, \quad (3.31)$$

where r_0 is the Fried parameter.

Due to isotropy and homogeneity, the structure function depends only on the magnitude of the difference $|\mathbf{x} - \mathbf{x}'|$ instead of the absolute positions. Thus, we can also write the structure function as $\mathcal{D}_\phi(|\xi - \xi'|)$. We further note that Equation (3.31) is valid for both planar and spherical waves (though, the definition of the Fried parameter should change accordingly!).

It should be noted that the Fried parameter varies in its definition from planar to spherical. Therefore, one must be aware of which definition is being considered. To mitigate this, many sources denote the spherical Fried parameter as $r_{0,sw}$ (though, we typically will not follow such notation). Additionally, for a constant C_n^2 profile, the spherical and planar Fried parameters are related by [34]

$$r_{0,sw} = \left(\frac{8}{3} \right)^{3/5} r_0.$$

Example. In MATLAB, the Fried parameter can be determined using the command `integral`. If we let $C_n^2(z) = 1 \times 10^{-15} \text{ m}^{-2/3}$ for all z , $\lambda = 525 \text{ nm}$, and $L = 7000 \text{ m}$, then the Fried parameter is $r_0 = 0.0478 \text{ m}$. For an aperture of $D = 0.2034 \text{ m}$, the ratio D/r_0 is approximately $D/r_0 = 4.26$. This means the smallest spot (i.e., the diffraction PSF) increases 4.26 times due to turbulence.

A closely related quantity to the Fried parameter is the **isoplanatic angle**. The isoplanatic angle defines the field of view within which the turbulence effects are said to be similar. The definition of the isoplanatic angle arises from the adaptive optics community, wherein a guide star is used to “calibrate” the imaging system for another, nearby point. The isoplanatic angle quantifies this angle by the following definition [34]:

Definition 3.6 (Isoplanatic Angle). *The isoplanatic angle is*

$$\theta_0 = 58.1 \times 10^{-3} \lambda^{6/5} \left[\int_0^L z^{5/3} C_n^2(z) dz \right]^{-3/5}, \quad (3.32)$$

which quantifies the maximum angular separation by which adaptive optics methods will perform suitably well.

For the simulation techniques discussed towards the end of this Chapter, both the Fried parameter and isoplanatic angle are the metrics that can be used to quantify how close our simulation matches with the specified turbulence profile and geometry.

3.4.2 Instantaneous OTF

Recalling the results of Chapter 2.5, Theorem 2.7 says that if the geometric image is $I_g(\mathbf{x})$, the observed image $I_i(\mathbf{x})$ will be

$$I_i(\mathbf{x}) = |h(\mathbf{x})|^2 \circledast I_g(\mathbf{x}). \quad (3.33)$$

In the Fourier domain where we define $\tilde{I}_i(\mathbf{f}) = \text{Fourier}[I_i(\mathbf{x})]$, it follows that

$$\tilde{I}_i(\mathbf{f}) = \mathcal{H}(\mathbf{f}) \tilde{I}_g(\mathbf{f}). \quad (3.34)$$

Therefore, the resolution and quality of the observed image are determined by $\mathcal{H}(\mathbf{f})$.

Consider now that there is turbulence present in the system. The turbulence effect is modeled as the product of amplitude distortion, phase distortion, and the pupil function. Substituting our understanding of the complex pupil function into the definition of the OTF, we can show that

$$\begin{aligned} \mathcal{H}(\mathbf{f}) &= H(\mathbf{f}) \circledast H^*(\mathbf{f}) \\ &= \left(P(\lambda z \mathbf{f}) e^{-j\phi(\lambda z \mathbf{f})} \right) \circledast \left(P(\lambda z \mathbf{f}) e^{j\phi(\lambda z \mathbf{f})} \right), \end{aligned}$$

where $(\cdot)^*$ denotes complex conjugate, and \circledast denotes convolution. We refer to this OTF as the **instantaneous OTF**. The instantaneous OTF is what the optical system sees at a particular instant. It is a *random* OTF because the phase function $\phi(\lambda z \mathbf{f})$ is random.

3.4. IMPORTANT APPLICATIONS OF THE MODEL

Definition 3.7 (Instantaneous OTF). Let $P(\mathbf{u})$ be the aperture located at a distance z from the object, and let $\phi(\lambda z \mathbf{f})$ be the random phase distortion. The instantaneous OTF is

$$\mathcal{H}(\mathbf{f}) = \left(P(\lambda z \mathbf{f}) e^{-j\phi(\lambda z \mathbf{f})} \right) \circledast \left(P(\lambda z \mathbf{f}) e^{j\phi(\lambda z \mathbf{f})} \right), \quad (3.35)$$

where \circledast denotes the convolution.

Example. (Linear phase offset.) As an example of how the phase distortion can affect the observed image, we consider the case where

$$\phi(\boldsymbol{\xi}) = -\frac{2\pi}{\lambda z} \boldsymbol{\alpha}^T \boldsymbol{\xi}$$

for some random vector $\boldsymbol{\alpha}$. Assuming a circular aperture $P(\boldsymbol{\xi})$, the amplitude transfer function is

$$H(\mathbf{f}) = P(\lambda z \mathbf{f}) e^{j\phi(\lambda z \mathbf{f})} = P(\lambda z \mathbf{f}) e^{-j2\pi \boldsymbol{\alpha}^T \mathbf{f}},$$

where the inverse Fourier transform is

$$\mathcal{F}^{-1} \left[P(\lambda z \mathbf{f}) e^{-j2\pi \boldsymbol{\alpha}^T \mathbf{f}} \right] = p \left(\frac{\mathbf{x} - \boldsymbol{\alpha}}{\lambda z} \right) \quad (3.36)$$

where $p(\mathbf{x}) = J_1(2\pi|\mathbf{x}|)/|\mathbf{x}|$. Therefore, the instantaneous PSF is

$$\begin{aligned} h(\mathbf{x}) &= \left| p \left(\frac{\mathbf{x} - \boldsymbol{\alpha}}{\lambda z} \right) \right|^2 \\ &= J_1 \left(2\pi \frac{|\mathbf{x} - \boldsymbol{\alpha}|}{\lambda z} \right)^2 / \left(\frac{|\mathbf{x} - \boldsymbol{\alpha}|}{\lambda z} \right)^2, \end{aligned} \quad (3.37)$$

which is simply a shifted version of the Airy disc. Therefore, $\phi(\boldsymbol{\xi}) = -\frac{2\pi}{\lambda z} \boldsymbol{\alpha}^T \boldsymbol{\xi}$ introduces a random tilt to the PSF. The amount of the random tilt is specified by $\boldsymbol{\alpha}$.

What does the instantaneous OTF look like? One way to visualize the effect is to consider the PSF by taking the inverse Fourier transform of the OTF:

$$\underbrace{|h(\mathbf{x})|^2}_{\text{PSF}} = \mathfrak{Fourier}^{-1} \{ \mathcal{H}(\mathbf{f}) \}. \quad (3.38)$$

CHAPTER 3. MODELING TURBULENCE

Since $\phi(\xi)$ is random, $|h(\mathbf{x})|^2$ is also random. Figure 3.8 shows a few snapshots of the PSFs generated from Kolmogorov theory. The shape of the PSF changes according to the strength of the turbulence.

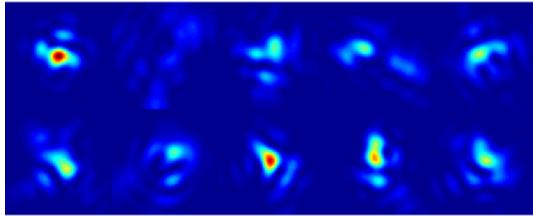


Figure 3.8: Instantaneous point spread functions $|h(\mathbf{x})|^2$ generated from propagation through Kolmogorov phase screens. The PSFs shown in this figure are statistically independent. In practice, the spatial correlation needs to be taken into consideration to generate a realistic turbulence effect.

The randomness of the OTF is governed by two effects:

- **Tilt:** The random pixel displacement caused by the first-order effect of the phase $\phi(\xi)$. The tilt will cause the pixels to depart from their ideal locations to another location.
- **Aberration:** The high-order effects such as blur, spherical aberration, elliptical aberration, etc. caused by the phase delays (and leads) in $\phi(\xi)$. The aberration will cause a mixing of pixel values, thus making sharp edges become blurred.

The decomposition of the phase into tilt and aberration can be summarized by the equation:

$$\underbrace{\phi(\xi)}_{\text{overall distortion}} = \underbrace{\varphi(\xi)}_{\text{aberration}} + \underbrace{\boldsymbol{\alpha}^T \xi}_{\text{tilt}}, \quad (3.39)$$

where $\boldsymbol{\alpha}$ is a vector defining the best-fitted first-order plane to the phase function. In the following discussions, we will introduce two OTFS: a **long exposure OTF** defined through $\phi(\xi)$ and a **short exposure** OTF constructed using $\varphi(\xi)$. The short exposure OTF is also known as the **tilt-free** OTF, which plays an important role in adaptive optics and the short-exposure analysis.

3.4. IMPORTANT APPLICATIONS OF THE MODEL

3.4.3 Long Exposure OTF

The long exposure function can be seen as the OTF that will arise from leaving the camera exposure open infinitely long. To this end, we take the statistical expectation on both sides of the equation

$$\begin{aligned}\mathbb{E}_\phi[\mathcal{H}(\mathbf{f})] &= \mathbb{E}_\phi \left[\left(W(\lambda z \mathbf{f}) e^{-j\phi(\lambda z \mathbf{f})} \right) \circledast \left(W(\lambda z \mathbf{f}) e^{j\phi(\lambda z \mathbf{f})} \right) \right] \\ &= \mathbb{E}_\phi \left[\int W(\boldsymbol{\xi}) e^{-j\phi(\boldsymbol{\xi})} \cdot W(\boldsymbol{\xi} - \lambda z \mathbf{f}) e^{j\phi(\boldsymbol{\xi} - \lambda z \mathbf{f})} d\boldsymbol{\xi} \right] \\ &= \int W(\boldsymbol{\xi}) W(\boldsymbol{\xi} - \lambda z \mathbf{f}) \mathbb{E}_\phi[e^{-j\phi(\boldsymbol{\xi})} e^{j\phi(\boldsymbol{\xi} - \lambda z \mathbf{f})}] d\boldsymbol{\xi},\end{aligned}\quad (3.40)$$

where the subscript $\mathbb{E}_\phi[\cdot]$ emphasizes that the expectation is taken with respect to the random phase ϕ .

Recognizing the expectation within the integrand, we may utilize the moment generating function (3.18) as we did in Chapter 3.3 to directly write

$$\mathbb{E}_\phi[e^{-j\phi(\boldsymbol{\xi})} e^{j\phi(\boldsymbol{\xi} - \lambda z \mathbf{f})}] = \Gamma_{t_\phi}(\boldsymbol{\xi}, \boldsymbol{\xi} - \lambda z \mathbf{f}), \quad (3.41)$$

$$= \exp \left\{ -\frac{1}{2} D_\phi(\lambda z |\mathbf{f}|) \right\}, \quad (3.42)$$

$$= \exp \left\{ -\frac{1}{2} 6.88 \left(\frac{\lambda z |\mathbf{f}|}{r_0} \right)^{5/3} \right\}. \quad (3.43)$$

where we have used the homogeneous property of the phase structure function. This results in the expected value of the OTF to be decomposed as

$$\mathbb{E}_\phi[\mathcal{H}(\mathbf{f})] = \underbrace{\int W(\boldsymbol{\xi}) W(\boldsymbol{\xi} - \lambda z \mathbf{f}) d\boldsymbol{\xi}}_{\mathcal{H}_{\text{diff}}(\mathbf{f})} \times \underbrace{\exp \left\{ -\frac{1}{2} 6.88 \left(\frac{\lambda z |\mathbf{f}|}{r_0} \right)^{5/3} \right\}}_{\mathcal{H}_{\text{LE}}(\mathbf{f})},$$

which is a product of $\mathcal{H}_{\text{diff}}(\mathbf{f})$ and $\mathcal{H}_{\text{LE}}(\mathbf{f})$:

Theorem 3.1 (Decomposition of Average OTF). *The average OTF consists of two terms:*

$$\mathbb{E}_\phi[\mathcal{H}(\mathbf{f})] = \mathcal{H}_{\text{diff}}(\mathbf{f}) \times \mathcal{H}_{\text{LE}}(\mathbf{f}), \quad (3.44)$$

where $\mathcal{H}_{\text{diff}}(\mathbf{f})$ denotes the diffraction-limited OTF and $\mathcal{H}_{\text{LE}}(\mathbf{f})$

CHAPTER 3. MODELING TURBULENCE

denotes the long exposure OTF of the atmosphere.

For a circular aperture, $\mathcal{H}_{\text{diff}}(\mathbf{f})$ follows from Chapter 2.4:

Theorem 3.2 (Diffraction limited OTF). *For a circular aperture, the diffraction limited OTF is*

$$\mathcal{H}_{\text{diff}}(\mathbf{f}) = \frac{2}{\pi} \left[\arccos \left(\frac{|\mathbf{f}|}{2f_0} \right) - \frac{|\mathbf{f}|}{2f_0} \sqrt{1 - \left(\frac{|\mathbf{f}|}{2f_0} \right)^2} \right], \quad (3.45)$$

for $f \leq 2f_0$ where f_0 is the cutoff frequency of the coherent version of the system.

The atmospheric term $\mathcal{H}_{\text{LE}}(\mathbf{f})$ is known as the long exposure OTF, which is what we would observe if we turn on the shutter of the camera for a prolonged period of time.

Definition 3.8 (Long Exposure OTF). *The long exposure optical transfer function $\mathcal{H}_{\text{LE}}(\mathbf{f})$ is defined as*

$$\mathcal{H}_{\text{LE}}(\mathbf{f}) = \exp \left\{ -3.44 \left(\frac{\lambda z |\mathbf{f}|}{r_0} \right)^{5/3} \right\}, \quad (3.46)$$

where z is the path length, and r_0 is the Fried parameter.

The difference between the long exposure OTF and the instantaneous OTF is that the instantaneous OTF is random whereas the long exposure OTF is deterministic. The randomness is absorbed by the structure function of the phase when evaluating the joint expectation.

As a visualization of the instantaneous OTF and the long-exposure OTF, in Figure 3.9 we show a comparison. Here, we plot the cross-section of the PSF (instead of the OTF) using a finite-sample ensemble average of 10, 100, and 5000 instantaneous PSFs. For the weaker turbulence case where $C_n^2 = 2.5 \times 10^{-16} \text{m}^{-2/3}$, we see that the random fluctuation of the instantaneous PSF becomes weak as soon as we use 100 instantaneous PSFs. If we consider a stronger turbulence case where $C_n^2 = 1 \times 10^{-15} \text{m}^{-2/3}$, the random fluctuation is stronger. Moreover, the spread of the long-exposure PSF is also greater when we use a higher C_n^2 .

3.4. IMPORTANT APPLICATIONS OF THE MODEL

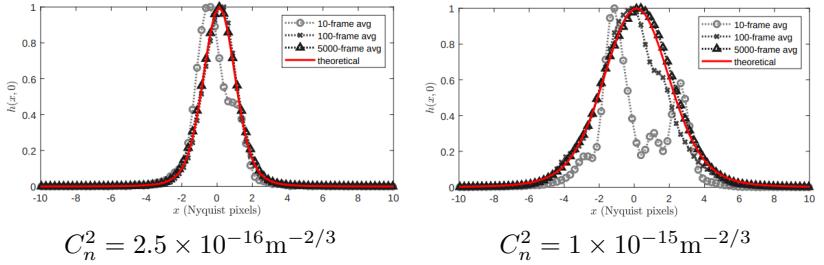


Figure 3.9: The theoretical long-exposure PSF compared with the empirical averaged instantaneous PSFs for $C_n^2 = 2.5 \times 10^{-16} \text{m}^{-2/3}$ and $C_n^2 = 1 \times 10^{-15} \text{m}^{-2/3}$. Shown in each sub-figure are the 10-frame average, 100-frame average, and 5000-frame average PSFs, all in black. The red curve is the theoretically predicted result. The x -axis denotes the Nyquist pixel, where one pixel corresponds to $\lambda z/D$ with D being the aperture diameter.

Figure 3.10 shows another visual comparison between the finite-sample ensemble average and the statistical long-exposure. As we can see from the plots, the more instantaneous PSFs we use to compute the ensemble average, the closer the average will converge to the true long-exposure.

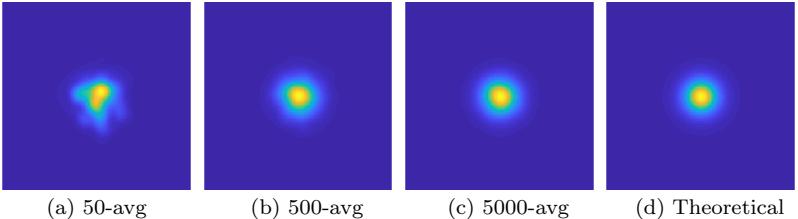


Figure 3.10: Instantaneous PSFs for $C_n^2 = 1 \times 10^{-15} \text{m}^{-2/3}$. (a) 50-frame average. (b) 500-frame average. (c) 5000-frame average. (d) Theoretical long-exposure PSF.

3.4.4 Short Exposure OTF

The long-exposure OTF is the statistical average of an infinite number of instantaneous OTFs observed over a very long period of time. In many advanced imaging systems, fast steering mirrors are installed

CHAPTER 3. MODELING TURBULENCE

to adaptively compensate for the **tilt** observed in the wavefront. The mathematical model removing the tilt from the phase is to subtract a linear term from the observed phase:

$$\varphi(\xi) \stackrel{\text{def}}{=} \phi(\xi) - \boldsymbol{\alpha}^T \xi, \quad (3.47)$$

where $\boldsymbol{\alpha}^T \xi$ is the best linear fit to $\phi(\xi)$.

An image taken without the tilt is known as the **tilt-compensated** frame. If we can obtain many of these tilt-compensated frames, then we can take another statistical average known as the **short exposure** image. The short exposure image can be thought of *freezing* the turbulence and re-aligning the pixels such that they are centered. The tilt contributes a majority of the energy of the distortions (such as in Table 4 of Noll [39] for example), and as such, being able to use hardware to compensate for the tilt can lead to a much better image.

Now consider the tilt-compensated phase realization. In this case, the phase is

$$t_\varphi(\xi) = \exp \{j\varphi(\xi)\} = \exp \{j(\phi(\xi) - \boldsymbol{\alpha}^T \xi)\}. \quad (3.48)$$

The autocorrelation of the tilt-correct wavefront is

$$\begin{aligned} \Gamma_{t_\varphi}(\xi, \xi') &= \mathbb{E}[t_\varphi(\xi)t_\varphi^*(\xi')] \\ &= \mathbb{E}[\exp \{j(\phi(\xi) - \boldsymbol{\alpha}^T \xi) - j(\phi(\xi') - \boldsymbol{\alpha}^T \xi')\}] \\ &= \mathbb{E}[\exp \{j(\phi(\xi) - \phi(\xi') - \boldsymbol{\alpha}^T(\xi - \xi'))\}]. \end{aligned} \quad (3.49)$$

Using the same trick of the moment generating function of a zero-mean Gaussian, we can show that

$$\begin{aligned} \Gamma_{t_\varphi}(\xi, \xi') &= \exp \left\{ -\frac{1}{2}\mathbb{E}[(\phi(\xi) - \phi(\xi'))^2] \right. \\ &\quad \left. + \mathbb{E}[(\phi(\xi) - \phi(\xi'))\boldsymbol{\alpha}^T(\xi - \xi')] - \frac{1}{2}\mathbb{E}[(\boldsymbol{\alpha}^T(\xi - \xi'))^2] \right\}. \end{aligned} \quad (3.50)$$

At this point, we aim to simplify the preceding expression. This leads us to separately analyze the three terms in the exponential:

- The first term is $\mathbb{E}[(\phi(\xi) - \phi(\xi'))^2]$, which is nothing but the structure function of the phase. Therefore, we have

$$\mathbb{E}[(\phi(\xi) - \phi(\xi'))^2] = \mathcal{D}_\phi(\xi - \xi'), \quad (3.51)$$

where we assumed that the phase is homogeneous so that the structure function can be written in terms of $\xi - \xi'$.

3.4. IMPORTANT APPLICATIONS OF THE MODEL

- The second term is the cross correlation between the phase difference $\phi(\xi) - \phi(\xi')$ and the linear term $\alpha^T(\xi - \xi')$. Following the argument of Fried, we may assume that these two terms are statistically independent (though, this approximation has been considered more carefully by Heidbreder [57], showing the term may not always be so easily neglected). Therefore, by using the zero-mean property of the phase, we can show that

$$\mathbb{E}[(\phi(\xi) - \phi(\xi'))\alpha^T(\xi - \xi')] = \underbrace{\mathbb{E}[\phi(\xi) - \phi(\xi')]}_{=0}\mathbb{E}[\alpha^T(\xi - \xi')].$$

Hence, the cross term can be dropped.

- The third term is the second moment of $\alpha^T \xi$, which is

$$\mathbb{E}[(\alpha^T(\xi - \xi'))^2] = \mathbb{E}[|\alpha|^2] \cdot |\xi - \xi'|^2. \quad (3.52)$$

According to Fried [24], the expectation $\mathbb{E}[|\alpha|^2]$ is

$$\mathbb{E}[|\alpha|^2] = 6.88r_0^{-5/3}D^{-1/3}, \quad (3.53)$$

where D is the aperture diameter. Substituting this into Equation (3.52) yields

$$\mathbb{E}[(\alpha^T(\xi - \xi'))^2] = 6.88r_0^{-5/3}D^{-1/3}|\xi - \xi'|^2.$$

Combining all three terms, we can show that

$$\begin{aligned} \Gamma_{t_\varphi}(\xi, \xi') &= \exp \left\{ -\frac{1}{2}\mathcal{D}_\phi(\xi - \xi') - \frac{1}{2}\mathbb{E}[(\alpha^T(\xi - \xi'))^2] \right\} \\ &= \exp \left\{ -\frac{1}{2}6.88 \left(\frac{|\xi - \xi'|}{r_0} \right)^{5/3} - \frac{1}{2}6.88 \frac{|\xi - \xi'|^2}{r_0^{5/3}D^{1/3}} \right\} \\ &= \exp \left\{ -3.44 \left(\frac{|\xi - \xi'|}{r_0} \right)^{5/3} \left(1 - \left(\frac{|\xi - \xi'|}{D} \right)^{1/3} \right) \right\}. \end{aligned} \quad (3.54)$$

Following the same argument as we did in Equation (3.40), we

CHAPTER 3. MODELING TURBULENCE

can show that the short exposure OTF is

$$\mathbb{E}_\varphi[\mathcal{H}(\mathbf{f})] = \underbrace{\int W(\boldsymbol{\xi})W(\boldsymbol{\xi} - \lambda z \mathbf{f}) d\boldsymbol{\xi}}_{\mathcal{H}_{\text{diff}}(\mathbf{f})} \times \underbrace{\exp \left\{ -3.44 \left(\frac{\lambda z |\mathbf{f}|}{r_0} \right)^{5/3} \left(1 - \left(\frac{\lambda z |\mathbf{f}|}{D} \right)^{1/3} \right) \right\}}_{\mathcal{H}_{\text{SE}}(\mathbf{f})}.$$

Thus, we arrived at another important result.

Definition 3.9 (Short Exposure OTF). *The short exposure optical transfer function $\mathcal{H}_{\text{SE}}(\mathbf{f})$ is defined as*

$$\mathcal{H}_{\text{SE}}(\mathbf{f}) = \exp \left\{ -3.44 \left(\frac{\lambda z |\mathbf{f}|}{r_0} \right)^{5/3} \left(1 - \left(\frac{\lambda z |\mathbf{f}|}{D} \right)^{1/3} \right) \right\}, \quad (3.55)$$

where D is the aperture diameter, z is the path length, and r_0 is the Fried parameter.

Figure 3.11 shows the short-exposure point spread functions for two levels of C_n^2 . Compared to Figure 3.9, the shape of the short exposure PSF is narrower. The reason is that when the tilts are compensated, the PSFs will be centered. The random fluctuation is thus reduced compared to the long-exposure case.

3.4.5 Probability of Getting a Lucky Observation

The atmosphere fluctuates in time, and so do the resultant PSFs that distort an image. This leads us to an interesting component of atmospheric imaging. When observing a point source, sometimes the point will look bad and sometimes it will look good! This is simply a result of the random nature of the atmosphere; sometimes we get *lucky* and sometimes we don't.

The short exposure function introduced the tilt-compensated phase term known as the **effective wavefront**. The effective wavefront contributes blur from the higher order aberrations. If we are interested in taking a clean image of a star, whether or not it has shifted by a few pixels is hardly a consideration if it means having a non-blurry image.

3.4. IMPORTANT APPLICATIONS OF THE MODEL

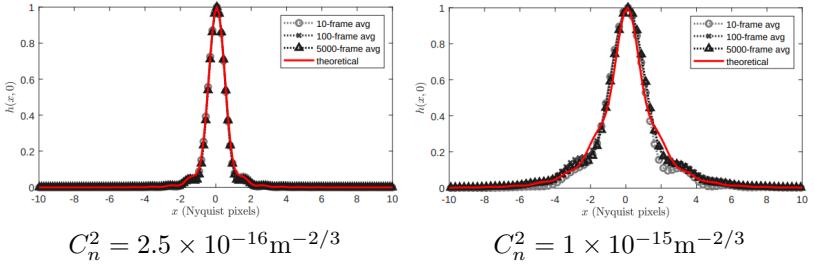


Figure 3.11: The theoretical short-exposure PSF compared with the empirical averaged instantaneous PSFs for $C_n^2 = 2.5 \times 10^{-16} \text{m}^{-2/3}$ and $C_n^2 = 1 \times 10^{-15} \text{m}^{-2/3}$. Shown in each sub-figure are the 10-frame average, 100-frame average and 5000-frame average PSFs, all in black. The red curve is the theoretically predicted result. The x -axis denotes the Nyquist pixel, where one pixel corresponds to $\lambda z/D$ with D being the aperture diameter.

For this reason, Fried refers to φ as the effective wavefront, as it is the component of the wavefront that will effectively limit us in resolution.

In 1978 Fried analyzed the probability of having a lucky observation [27]. We may anticipate that a lucky observation is defined to be a PSF that is near a delta function. However, there is no closed form expression for a turbulent PSF, therefore we cannot analyze the PSF directly. Fried instead focuses on the mean square of the effective wavefront. Fried defines the mean square distortion to be

$$\Delta^2(\varphi) = \left(\frac{\pi}{4}D^2\right)^{-1} \int_{\text{Aperture}} P\left(\frac{\xi}{D}\right) |\varphi(\xi)|^2 d\xi, \quad (3.56)$$

where we are normalizing by the area of the aperture. Fried's analysis ultimately results in the following approximate result:

Definition 3.10 (Probability of a Lucky Event). *The probability of having a lucky observation is approximated as*

$$\mathbb{P}(\Delta^2(\varphi) \leq 1 \text{ rad}^2) \approx 5.6 \exp(-0.1557(D/r_0)^2) \quad (3.57)$$

for $D/r_0 \geq 3.5$ with D as the imaging system's aperture diameter and r_0 as the Fried parameter.

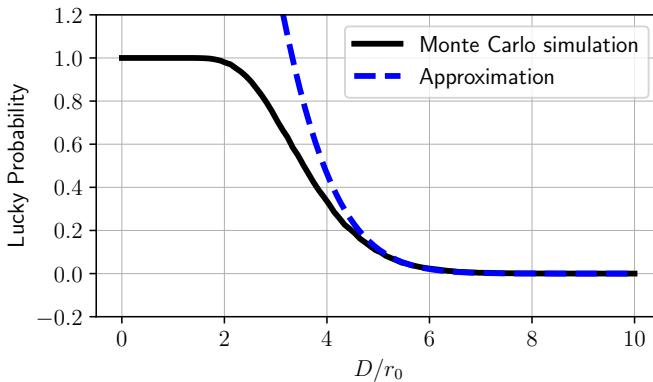


Figure 3.12: Lucky probability curve. As D/r_0 increases the probability of a lucky observation drops with a sharp transition.

This result showcases the fact that this D/r_0 ratio continually arises in the atmospheric turbulence literature. Fried additionally performs a Monte Carlo simulation, which we reproduce in Figure 3.12. We can observe that at the critical threshold near $D/r_0 = 2$, the lucky probability begins its sharp transition. This can help to inform us as to the definition of the Fried parameter. When the Fried parameter is larger than the aperture diameter, the imaging system is near maximum efficiency. However, the moment r_0 drops below the size of the aperture, the system drops in its efficiency, dramatically attenuating as the ratio of D/r_0 increases above 2.

We will revisit the discussions of the lucky effect in Chapter 5. There, we will take the perspective of image reconstruction algorithms.

3.5 Split-Step Simulation

With the theoretical descriptions of the atmospheric model presented, we now describe a way towards simulating these effects. The literature regarding the simulation of imaging through atmospheric turbulence presents a few varieties of simulations, with split-step being likely the most intuitive one. Implementing split-step such that the statistics are correct is a non-trivial task, thus we recommend the book by Schmidt [45] which provides an excellent, in-depth description of numerical op-

3.5. SPLIT-STEP SIMULATION

tical simulations and split-step for atmospheric turbulence along with detailed analysis. This book pairs well with the book from Voelz [58] which deals with general optical simulation principles.

3.5.1 A History of Split-Step Simulation

The modality of split-step simulation is a general approach that has been used both computationally [59, 60, 61] and in a laboratory sense with physical devices such as tunable phase modulators [62, 63, 64], heat sources [65, 66], or plastic screens [67], where a series of physical devices are arranged in such a way that a propagating wave is perturbed by the device. A clear visualization of such an experimental setup with details is provided in a work by Pellizzari et al. [68]. Computationally, the work generating the first phase screen simulator by use of Fourier-based **phase screens** was McGlamery [69].

The scope of split-step is not limited to the propagation of atmospheric turbulence, with an example being an early application of this principle to sound waves through media [70] and its subsequent consideration as a general simulation tool [71]. Applications towards wave propagation through turbulence were later presented [72, 73] which detailed both the approach and constraints in sampling upon properly performing a simulation of this variety. Proper sampling is critical in a split-step simulation. We ignore a majority of these details, opting for a simple rule, though a thorough explanation of the sampling criteria is given by Schmidt [45].

The introduction of randomness and the Kolmogorov turbulent model into the simulation is by a set of turbulent layers which are numerically represented by phase screens. As a result, a great deal of effort has been dedicated to an accurate representation of these effects [35, 74, 75, 76, 36, 77, 78, 79]. We will briefly cover the prevailing approach towards accurately generating phase screens, as not including these more nuanced aspects leads to poor results in terms of statistical accuracy. Our coverage will be somewhat brief, thus our presentation should be considered as an introduction.

3.5.2 Split-Step Simulation: Building Blocks

Conceptually, split-step is a natural approach to the simulation of atmospheric turbulence effects. Our derivation of the distortions in the wave used phase screens to represent the atmosphere distributed across

CHAPTER 3. MODELING TURBULENCE

the path of propagation. Split-step is the numerical equivalence of this process: we will segment the propagation path into a set of screens which we will numerically propagate a wave through.

We present a visualization of the split-step simulation in Figure 3.13. The split-step method is able to match atmospheric statistics and produce realistic generations of images for a wide variety of simulations. We present a step-by-step process of the simulation method, elaborating on the details throughout Chapter 3.5.

1. **Phase screen generation.** The first and arguably most important step is the generation of the numerical phase screens. These phase screens set aspects of turbulence strength and imaging geometry. We will consider a set of phase screens $\{\phi_i\}$.
2. **Numerical wave propagation.** Once the phase screens are generated, a point source is modeled and propagated through the series of phase screens. This is done by numerical wave propagation (via Fresnel or alternative methods) followed by a phase imparting step, directly following Equation (3.16). We emphasize a key point here: this point source generation and propagation is performed *per-pixel*.
3. **PSF formation.** With the wave propagated to the aperture plane, we apply the PSF formation equation Equation (2.61) and accordingly apply it in an incoherent or coherent fashion.

As noted, the operation of numerical wave propagation is done per pixel. This utilizes the superposition property of wave propagation. This represents the main computational bottleneck of split-step.

3.5.3 Generating Phase Screens

The first step of the split-step propagation is the generation of the phase screens $\{\phi_i\}$. As the name suggests, a phase screen is a 2D function of the phase that should be imparted to the incident field to generate the field at the next step. Since the phase screen is a random field, it needs to be sampled from a probability distribution.

The distribution for the phase screen, in most of the literature, is assumed to be a multivariate Gaussian [22]. If we discretize the continuous phase screen as a finite size array, then we are essentially drawing a random vector from a high-dimensional Gaussian distribution. Since a Gaussian is fully characterized by the correlation matrix

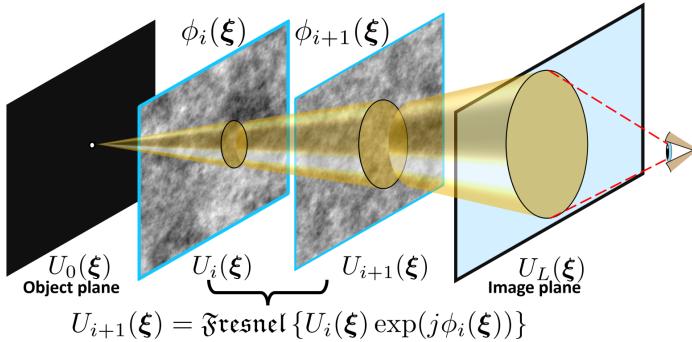


Figure 3.13: Visualization of split-step. A point source (more generally a grid of point sources) is propagated through numerically generated phase screens, traveling between the phase screens by Fresnel propagation.

(or the power spectral density in the Fourier domain), we can use the results such as the Kolmogorov power spectral density. An example of generating a phase screen is presented below. The result of a phase screen from this process is presented in Figure 3.14.

Example. (Drawing a Random Field from the Kolmogorov PSD.) Like a one-dimensional random process, drawing samples from a given power spectral density can be done in the Fourier domain. Denote Γ_n as the correlation matrix with the (i, j) th entry being $[\Gamma_n]_{ij} = \Gamma_n(\mathbf{r}_i - \mathbf{r}_j)$ for coordinates \mathbf{r}_i and \mathbf{r}_j . The Fourier relationship $\Gamma_n(\mathbf{r}) \xrightarrow{\mathfrak{F}\text{ourier}} \Phi_n(\mathbf{k})$ implies that Γ_n is diagonalizable using the discrete Fourier transform \mathbf{F} :

$$\Gamma_n = \mathbf{F} \Phi_n \mathbf{F}^H,$$

where Φ_n is a diagonal matrix representing the power spectral density. The (i, j) th entry is $[\Phi_n]_{ij} = \Phi_n(\mathbf{k}_i, \mathbf{k}_j)$.

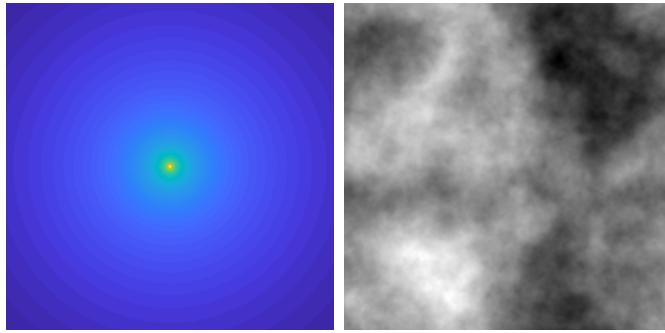
To draw a zero-mean Gaussian random field \mathbf{y} according to Γ_n , we can start with a white noise vector $\mathbf{w} \sim \text{Gaussian}(0, 1)$, and multiply it with

$$\mathbf{y} = \mathbf{F} \Phi_n^{\frac{1}{2}} \mathbf{w}.$$

CHAPTER 3. MODELING TURBULENCE

Then, one can show that $\mathbb{E}[\mathbf{y}\mathbf{y}^T] = \Phi_n$.

A piece of short MATLAB code to simulate the two-dimensional random field is given below. The generated random field is shown in Figure 3.14. In this example, we assumed $C_n^2 = 10^{-16}\text{m}^{-2/3}$ and we use a normalized coordinate $|k| \in [0, 1]$.



(a) Power spectral density
 $\Phi_n(\mathbf{k}) = 0.033C_n^2|\mathbf{k}|^{-11/3}$

(b) Random field
 Sample drawn from $\Phi_n(\mathbf{k})$

Figure 3.14: Kolmogorov power spectral density and a random realization of the random field generating from the Kolmogorov PSD.

```
% MATLAB code
[kx, ky] = meshgrid(linspace(-1,1,512));
PSD_kolm = 0.033*1e-16*(sqrt(kx.^2+ky.^2)).^(-11/3);
w = randn(size(kx));
phi = ifft2(sqrt(PSD_kolm).*w);
imshow(abs(phi),[]);
```

Additionally, we must carefully consider the sampling of the phase screens. The sampling must be done so that (i) the sampling requirements for the phase screens are met and (ii) the sampling criterion for the aperture plane is also satisfied. A full discussion of this balance is provided in [45]. A standard approach towards meeting this criterion is the Voelz criterion for the phase screens [58],

$$\Delta x = \frac{\lambda L}{sD}, \quad (3.58)$$

with $s \geq 4$.

3.5. SPLIT-STEP SIMULATION

These considerations are still not enough to properly sample a phase screen. The Kolmogorov PSD is often underrepresented in the low frequency terms due to sampling due to the decay near the small frequencies. The phase screens empirical structure function will not match the desired theoretical description (see Schmidt [45] Figure 9.3). To fix this, one may perform **subharmonic generation** which adds extra terms to the phase screen which models the low frequency components. The sampling of these phase screens, and which frequencies should be chosen, again must be done in coordination with other sample spacings in the system.

3.5.4 Numerical Wave Propagation

Wave propagation is the process by which the wave traverses the atmosphere, thus it must be a component of our simulation as well. Our modeling of numerical wave propagation will take place between each phase screen. This is the reasoning behind the name “split-step”, the steps of wave propagation and phase imparting are split apart.

Consider a point source located in our object plane defined at location $\mathbf{u} = \mathbf{0}$. We will model this by the assignment of this location to be $U_0(\mathbf{u}) = \delta(\mathbf{u})$. With this, the Fresnel integral of Equation (2.30) for a distance of z_i away is given by

$$U_1(\boldsymbol{\xi}) = \frac{e^{jkz_i}}{j\lambda z_i} e^{j\frac{k}{2z_i}|\boldsymbol{\xi}|^2}. \quad (3.59)$$

This result is sufficient for a point source, however, what should we do with the field after we propagate it to our first phase screen when it is no longer a point source? This leads us to write the Fresnel integral in a slightly different way than before

$$U_{i+1}(\boldsymbol{\xi}) = \int_{\Sigma} U_i(\mathbf{u}) \frac{e^{jkz_i}}{j\lambda z_i} e^{j\frac{k}{2z_i}|\boldsymbol{\xi}-\mathbf{u}|^2} d\mathbf{u}, \quad (3.60)$$

where we note that this *also* applies to the original propagation of a point source. Examining this equation, we notice this is simply a convolution. This gives us a new perspective of the Fresnel equation: Fresnel propagation is given by convolution with a spatially invariant kernel. This motivates us to define the Fresnel kernel,

$$h_{\text{Fres}}(\boldsymbol{\xi}; z) = \frac{e^{jkz}}{j\lambda z} e^{j\frac{k}{2z}|\boldsymbol{\xi}|^2}. \quad (3.61)$$

CHAPTER 3. MODELING TURBULENCE

Viewing Fresnel propagation as a convolutional kernel simplifies our analysis. Specifically, for *any* wave distribution at a particular propagation distance, we may compute the field distribution at a further distance by Fresnel kernel in Equation (3.61). The wave propagation including phase screens will then be achieved by

$$U_{i+1}(\xi) = [h_{\text{Fres}}(\xi; z) \circledast U_i(\xi)] e^{j\phi_i(\xi)}. \quad (3.62)$$

It is worth noting that the propagation (3.62) is notationally simplistic to write in this form, however, computationally we typically opt for **angular spectrum propagation** [11, 45]. Without going into too many details, the idea of the angular spectrum is to decompose a wave by a sum of plane waves. The propagation of a planar wave is simple to write, therefore, if we know the decomposition the propagation rules become easier. Mathematically this can be understood to be a Fourier decomposition.

3.5.5 Image Formation

After propagating our wave through a series of phase screens it will be incident upon our aperture. We will refer to this as $U_{M+1}(\mathbf{u})$. We must then model the wave passing through the lens-aperture system, then propagating to the sensor plane.

We can say our incident wave can be written as

$$U_{M+1}(\xi) = A(\xi) \exp \left\{ j \left(\frac{k|\xi|^2}{2L} + \phi(\xi) \right) \right\}. \quad (3.63)$$

Here we have simplified the terminology somewhat through $A(\xi)$ and $\phi(\xi)$ which are the amplitude and phase of the *resultant* split-step propagation steps. We wish to emphasize this point a bit further as it will be important for the next Chapter. We will refer to $\phi(\xi)$ as a **phase realization** if it is representative of the phase of a wave propagated through turbulence. Notice this is slightly different than a phase *screen*, which models a Chapter of the atmosphere's phase contribution.

At this point, we need to remove the Fresnel pattern in order to access the turbulence phase distortion $\phi(\xi)$ directly. This leads us to recall the thin lens equation

$$U'_M(\xi) = t_{\text{lens}}(\xi) U_M(\xi), \quad (3.64)$$

with our goal being that the response of the lens makes the following to be true,

$$U'_M(\xi) = A(\xi) \exp\{j\phi(\xi)\}. \quad (3.65)$$

3.5. SPLIT-STEP SIMULATION

If the lens is chosen such that t_{lens} is the complex conjugate of the Fresnel pattern, then it will cancel out this portion of the wavefront. Thus, we choose the lens response to be

$$t_{\text{lens}}(\xi) = \exp\left\{-j\frac{k|\xi|^2}{2L}\right\}. \quad (3.66)$$

Here we have assumed that the lens cancels the Fresnel pattern *and* will focus the incident wave to the focal plane. Typically, through careful construction of the point source, we can assume that $A(\mathbf{u}) = 1$ [59]. This leaves us with the standard PSF equation we have presented before,

$$|h(\mathbf{x})|^2 = \mathfrak{Fourier}\{P(\xi)e^{j\phi(\xi)}\}. \quad (3.67)$$

Throughout our derivation we have fixed $\mathbf{x} = \mathbf{0}$, therefore, the generalization of this result is the familiar equation

$$|h_{\mathbf{u}}(\mathbf{x})|^2 = \mathfrak{Fourier}\{P(\xi)e^{j\phi_{\mathbf{u}}(\xi)}\}. \quad (3.68)$$

How should we model the phase's dependency on \mathbf{u} ? We provide a visualization in Figure 3.15. This figure implies that nearby point sources will share components of the same phase screens. This is exactly the mechanism that will model the correlation. For practical implementation, the approach is often to generate large phase screens and propagate through paths over the same grid size whereas in our visualization the propagation window, represented by the cone, “grows”. We suggest the reader to [59] for a sense of how this process is done in practice. This figure also showcases *why* each point must be propagated individually. With the PSFs generated from this process, we can then take the spatially varying convolution

$$I_i(\mathbf{x}) = |h_{\mathbf{u}}(\mathbf{u})|^2 \stackrel{\mathbf{u}}{\circledast} I_g(\mathbf{x}), \quad (3.69)$$

to receive our output image.

3.5.6 Pseudo-Code for Split-Step

For the sake of clarity, we present a pseudo-code description of the split-step algorithm for a grid of point sources. For a full elaboration of the details and additional considerations, we point the reader towards [60, 59, 45, 80].

Summarizing this pseudo-code, we first generate phase screens of varying size, in accordance with Figure 3.13. As we move from point

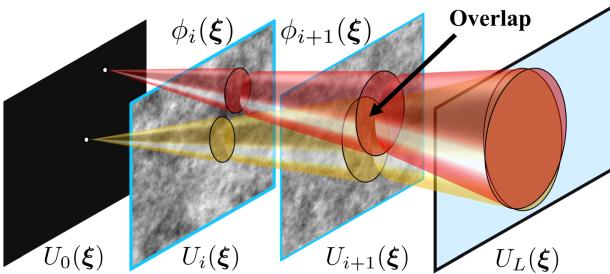


Figure 3.15: Visualization of how correlations arise in split step. The more overlap between the propagation paths of two points, the more correlated their output phase realizations, and thus their PSFs, will be.

Algorithm 1 Split-Step Simulation

- 1: Set sampling parameters for object plane space Ω_U
 - 2: Determine the size in meters for each phase screen
 - 3: Generate M phase screens at specified sizes according to PSD
 - 4: **for** $\mathbf{u} \in \Omega_U$ **do**
 - 5: **for** $i = 1, 2, \dots, M$ **do**
 - 6: $U_i(\xi; \mathbf{u}) = [h_{\text{Fres}}(\xi; z) \circledast U_{i-1}(\xi; \mathbf{u})] e^{j\phi_i(\xi; \mathbf{u})}$
 - 7: **end for**
 - 8: Form PSF, $|h_{\mathbf{u}}(\mathbf{x})|^2 = \mathfrak{Fourier}\{P(\xi) \cdot U_M(\xi)\}$
 - 9: **end for**
 - 10: Perform spatially varying convolution $I_i(\mathbf{y}) = |h_{\mathbf{u}}(\mathbf{x})|^2 \overset{\mathbf{u}}{\circledast} I_g(\mathbf{x})$
-

source to point source, we will propagate through different slices of the atmosphere. This will result in each PSF being different, though the overlap will contribute to the desired correlation. The simulation is completed once every PSF has been formed and the spatially varying convolution is computed.

3.6 Summary

With the details of turbulence discussed, we presented a simplified flow of how images are formed through turbulence. The process can be summarized into four parts:

Part 1 Atmospheric model: We began by discussing the turbulent nature of the atmosphere and characterized its fluctuations as a

3.6. SUMMARY

Gaussian process. The Kolmogorov PSD was presented, along with its according structure function $D_n(\mathbf{r}) = C_n^2 |\mathbf{r}|^{2/3}$. This additionally led us to describe C_n^2 , a parameter that characterizes the strength of turbulent fluctuations.

Part 2 Propagation through turbulence: The layered atmospheric model was introduced and utilized to arrive at our main result of the phase structure function $\mathcal{D}_\phi(|\boldsymbol{\xi} - \boldsymbol{\xi}'|)$ for propagation through a single layer of turbulence. This was then generalized to multiple screens, each of which may have different turbulence strength values. We also discussed the wave structure function, the more general version of the model for a wave propagated through turbulence.

Part 3 OTF and other properties: The result of our analysis produced the structure function of the phase, $\mathcal{D}_\phi(\boldsymbol{\xi}, \boldsymbol{\xi}') = 6.88(|\boldsymbol{\xi} - \boldsymbol{\xi}'|/r_0)^{5/3}$. This led us to define the Fried parameter r_0 as well as the isoplanatic angle θ_0 , both of which are insightful statistics for turbulent imaging. Additionally, the LE and SE OTFs were presented, characterizing the effects within the context of Fourier optics.

Part 4 Split-step simulation: The method of simulation by split-step propagation was presented. Split-step works by assuming thin layers of turbulence, known as phase screens, to model the atmosphere in discrete slices. Waves are numerically propagated through the slices to form the resultant phase realizations and images.

4. Propagation-Free Modeling and Simulation

Thus far, this book has largely derived from more traditional optics considerations, which we've done intentionally. With the fundamentals established, the time to begin the "translation" of optics into computer vision has come. This will lead us to formulate the previous Chapter in terms of standard engineering concepts such as **random vectors** and **basis representation**.

4.1 Motivation

In the previous Chapter, the model for atmospheric turbulence was presented along with the simulation approach of split-step. Split-step clearly arises as a natural extension of the model where we are discretizing the forward process of nature. The present Chapter, however, is dedicated to alternative models. Before presenting these alternatives, we would like to present why such a need exists depending on who you ask.

4.1.1 What's Wrong with Split-Step?

First, we must admit, there is really nothing significantly wrong with split-step. In fact, it is arguably the most general model; the alternatives we present will leave out effects which split-step can represent. One must remember that the authors are *imaging* people – we want to use split-step for something more than forward simulation! To us, split-step poses the following challenges:

1. **Generation of a dataset.** The cost of generality and accuracy is the speed at which split-step can run. One of our papers led us to estimate the time to generate a dataset suitable for training one of our networks, one only needs to wait approximately 150 years [43]! This is clearly unsuitable for generating training data with moderate computational resources.
2. **How to plug into an end-to-end framework?** If we wanted to insert split-step into a network, how would we propagate the gradient for the loss term? Following a gradient through the method would offer many challenges and, if even possible, slow the training tremendously. As a result, split-step cannot be used as an innate prior at present time.

From our perspective, a perfect simulator would be one which has generality, the ability to generate a dataset, easily allows gradient computation, and maintains accuracy. It is likely that one must compromise on at least one of these properties.

4.1.2 Diagnosing the Problem and a Solution

Is it possible to point at one aspect of split-step and blame it for the pain we are suffering on the applications side? The authors have considered this point in detail over the last 5 years and have come to the following conclusion: the numerical wave propagation is to blame!¹

Split-step requires each point source to be numerically propagated through a series of phase screens. This requires $M + 1$ 2D FFTs for each point source on an $N \times N$ grid. Then, we must model the PSF formation, requiring an additional FFT. This results in $N^2(M + 2)$ 2D FFTs *and* a spatially varying convolution. If we can avoid taking so many FFTs, this will save us a great deal of computation.

This puts us in a difficult predicament. We have identified the problem as numerical wave propagation. However, the problem we want to study is *wave propagation* through turbulence. How can we resolve these two seemingly conflicting desires?

This leads us to make the following problem statement: *if* we can generate a set of phase functions without numerical wave propagation

¹Our confidence in blaming numerical propagation now is much more certain than it was in 2018. We also say this in good humor, the reasoning for doing wave propagation is of course well justified!

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

then our problems would be overwhelmingly solved. As one may anticipate, this introduces a whole new set of issues, however, it will avoid our main issues with numerical wave propagation.

This “skipping” of numerical wave propagation is exactly what the simulator proposed by the authors and collaborators achieves. What is more is that the accuracy is still guaranteed to be high, though there are a few carefully considered approximations that are unavoidable at this time. Our model also does not incorporate amplitude effects such as scintillation, however, this is in line with the previous Chapter’s discussions. We are not alone in such an endeavor, as there are many other alternatives to split-step which in many ways go about solving the same or related problems. Some rely on the split-step methodology more than others, though overwhelmingly the idea is the same: the goal is to skip straight to the phase or image effects and still maintain accuracy. This leads us to define these methods as **propagation-free methods**, methods that do not require numerical wave propagation.

4.2 A Survey of Propagation-Free Methods

The source of propagation-free methods is a combination of the optics and computer vision communities. This is in stark contrast to split-step, which was primarily developed within the physics/optics community. A purely optics-inspired method may present a disadvantage to someone interested in computer vision due to speed. If one desires a large amount of training data for the purposes of machine learning, split-step will likely not be suitable for this task [43, 42]. Propagation-free methods are typically motivated by speed, though the end applications may vary. These different end goals inspire different varieties of propagation-free simulations.

4.2.1 Simplistic Models

A natural starting point for alternative simulations is what we shall refer to as **zero-order approximations**. These methods are often used as a proof-of-concept type simulation in place of a more sophisticated simulator. These methods are inspired by the fact that turbulent images have two characteristics: they are distorted geometrically and blurry. The standard way to do this is that given an image that represents $I_g(\mathbf{u})$, the pixels are shifted randomly (per pixel) by a 2D Gaussian

4.2. A SURVEY OF PROPAGATION-FREE METHODS



Figure 4.1: In these methods, a spatially variant pixel shifting and spatially invariant blur are applied to produce an image that shares some similarity with turbulent imagery. Source: [81]

vector. After this, a Gaussian blur is applied to the image. The methods by Chak et al. [81] and Lau et al. [82] both utilize such a model, though the contributions of these papers are more on the restoration side, with their simulation methodology being used more for proof-of-concept. We present a figure from Chak et al. [81] in Figure 4.1.

These methods may indeed produce images that are geometrically warped and blurred. However, the same statistics that arise in the Kolmogorov model are clearly not represented in this framework. For these reasons, we typically suggest other methods to be utilized for simulation. That being said, the concept of pixel shifting and blur is useful in turbulence restoration as in the case of [82, 81] and Zhu and Milanfar [83]. We should note that this concept of splitting the operators of pixel shifting and blurring in itself and keeping the operators to be general is valid, such as done by Zhu and Milanfar [83] and further elaborated on from the side of simulation in [84]. We'll return to the topic of restoration and the use of the geometric warping and blurring model in Chapter 5.

4.2.2 Direct-to-Image Methods

The simplistic model uses the concept of pixel shifting and blur, though the ways in which these two elements are modeled are minimally dependent on the underlying physics. There are, however, ways to introduce the physics into the image formation process *without* going to the phase domain. By this, we mean that our goal is not to model ϕ_x , but to model its resultant PSF $|h_x|^2$ directly. This may feel as if it contradicts some of the previous discussions in Chapter 3 – we said there was no straightforward model of the PSF! However, we do know certain things about it, such as the long and short exposure functions which are closely related to the average PSFs. For these reasons, we refer to

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

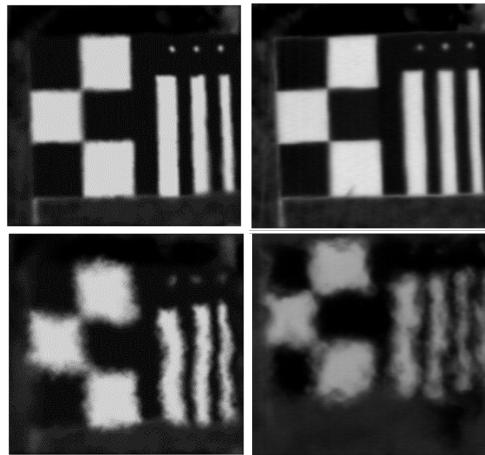


Figure 4.2: [Left] Two simulated images and [Right] real images with corresponding turbulence strength from the NATO dataset. Source: [85]

these methods as **direct-to-image methods**.

There exist some simulations within this space such as that by Repasi and Weiss [86, 85], Leonard et al. [87], or Potvin et al. [88] which use a blend of analytic and empirical properties (the empirical aspects partially based on the NATO RTG-40 dataset [89, 90]) to simulate PSFs and the subsequent images directly. The NATO dataset used various measurement devices, with one example being **scintillometers** to measure the effective turbulence level by providing an estimate for C_n^2 . This makes it an incredibly useful collection of data for the sake of simulator verification.

We present some images from Repasi and Weiss's paper that showcase some results from the NATO project (Figure 4.2) in which real and simulated images are compared (a digital ground truth is used for producing the simulated images). Leonard et al. [87] uses a similar approach that utilizes empirical results from the NATO dataset which we present in Figure 4.3.

These methods are similar in their approach, so we will speak a bit generally to include them under the same description. These methods use pixel displacements, similar to the simplistic models, however, the variance of the pixel displacement is given as a function of C_n^2 , wave-

4.2. A SURVEY OF PROPAGATION-FREE METHODS

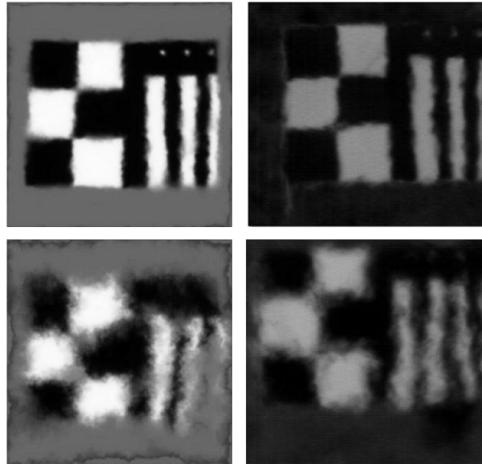


Figure 4.3: [Left] Simulated results for different turbulence strengths which correspond to [Right] field data. These results also utilize the NATO dataset. See [87] for more one-to-one comparisons and C_n^2 information. Source: [87].

length, distance, etc. The result is displacements that more closely match true turbulent cases. The pixel shifts are also drawn in a correlated fashion. With respect to the blur, both methods use the short exposure function as a mean width value for the blur size (i.e. the SE transfer function can be mapped to a SE PSF which is the average blur of centered PSFs) which is approximated as a Gaussian. The width is then allowed to vary in a patch or pixel-wise fashion. The result is a spatially varying blur that fluctuates similarly to turbulent cases.

These simulation methodologies have been revisited more recently by Miller et al. [91, 92]. We would point the interested reader to Figure 1 of Miller et al. [91] for a clear and concise visual description of their processing pipeline. The concept is similar, but we wish to emphasize a few key elements of these works. Firstly, a camera model is introduced in [91] which allows for the incorporation of temporal information in the imaging process (such as longer / shorter exposures) as well as sensor and post-processing effects, effectively modeling the entire image signal processing (ISP) pipeline. These papers also demonstrate the dependence of the image realization on a random seed, which produces three **random fields**, two for pixel shifts and one for the PSF blur width.

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

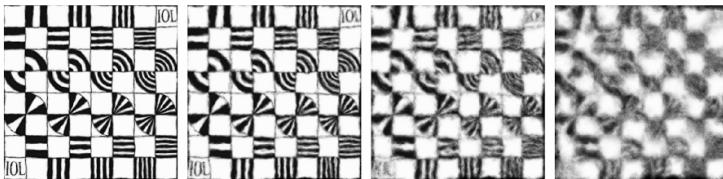


Figure 4.4: Results from the brightness function model with increasing D/r_0 (all simulated). Source [94].

This work also models temporal effects, which allows turbulence to vary over the acquisition time of an image which introduces additional blurring that is present in real imagery.

4.2.3 Ray Tracing Methods

There are other propagation-free methods which we refer to as ray-tracing, the first of which is known as the brightness function simulation [93, 94, 95]. The brightness function model is faster than split-step as a result of its lack of wave propagation evaluation, instead propagating “bundles” of rays through a perturbing medium. These bundles of rays are then distributed across the imaging plane for each pixel as a function of the medium, resulting in spatially varying effects as a function of the phase screens. Given certain choices in simulation parameters, the approach may be considerably faster than split-step.

We will not provide such a detailed account of this model as in the case of the previous discussion. The reason for this is that it is mostly similar to split-step, with the propagations being replaced with ray tracing. However, we would like to emphasize that this is more easily said than done with the main theoretical basis of this simulation with Vorontsov et al. [93] being a testament to this fact.

More “standard” ray tracing (i.e., those other than the brightness function) have been applied for the simulation and modeling of turbulent effects on an imaging system. Voelz et al. [96] provides an analysis of standard ray tracing approaches, with carefully performed ray tracing matching wave optics simulations to a suitable degree of accuracy for most applications. Additionally, a comprehensive work on a similar simulation modality is described by [97] and made publicly available (see paper for details).

4.2. A SURVEY OF PROPAGATION-FREE METHODS

4.2.4 Other Methods

There exist methods outside of these clear boundaries. For example, Hunt et al. [98] suggested an empirical PSF generation method by a dictionary of PSFs obtained by SVD. This particular approach is unique in that the PSFs decomposed by SVD were obtained in a real imaging scenario, thus incorporating real data into the generation step.

In recent years, deep learning methods are becoming popular. Generative models such as generative adversarial networks (GAN) have shown impressive results in synthesizing various image distortion effects, even within the field of atmospheric turbulence such as Feng et al. [99]. Our argument against GAN is that the GANs reported in the literature so far are still black box algorithms. Besides being not easily interpretable, these methods also lack robustness and generalization capability when handling unseen turbulence conditions. Furthermore, verifying the turbulence statistics remains a challenge because we cannot assess the validity of the steps inside the algorithm. The statistical model is in the wave space, while GAN models typically are designed to produce effects in the image space. Therefore, theoretical comparison is not feasible for a GAN of this variety as the turbulent model is given for a wave and not an image, with the exception of a few cases as a result of temporal averaging.

4.2.5 Phase-Based Simulation

In direct-to-image simulation, generated random fields described properties of the distortions in the image domain. **Phase-based simulation** is similar in this sense, however, instead the distortions model the **phase domain**. One may wonder why would this be preferred over the direct-to-image approach. The phase domain requires a conversion through the image formation equation, whereas direct-to-image requires no extra computation.

The answer is that the statistics are described not in the image domain, but in the phase domain! Therefore, if we wish to derive various properties and verify directly with the developments stemming from Tatarskii, it will be most easily done in the phase domain. These methods have been used for empirical study [40, 100], however, often only for a *single* point. This won't do us much good for simulating images under **anisoplanatic** conditions, i.e., conditions that are not isoplanatic.

One may suspect that if we know the Kolmogorov power spectral

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

density, the phase realizations can be drawn directly from the correlation matrix (or the structure function $\mathcal{D}_\phi(\xi, \xi')$ as we presented in Chapter 3). However, the structure function of the phase only describes the phase realization which emerges from a single point \mathbf{u} in the object plane. It does not tell us how the phase functions $\phi_{\mathbf{u}}(\mathbf{f})$ and $\phi_{\mathbf{u}'}(\mathbf{f})$, located at \mathbf{u} and \mathbf{u}' , are related to one another. Thus, moving towards simulating images will require additional considerations.

This is the exact problem that was considered by the authors and collaborators in a series of publications [41, 42, 43, 101]. As a result, we are intimately familiar with the ideas that generated this simulator, which we refer to as **Zernike-based simulation**. The rest of this Chapter will be dedicated largely to understanding this variety of simulation, partly due to our familiarity with it but more for another reason. This method is fundamentally motivated for use in image processing, therefore, the tricks and language used to describe it comes from this lens. Therefore, describing this simulation will serve as a tool for interpreting Chapter 3 through a more image processing/computer vision lens. Our coverage will be reasonably detailed, but not complete; if one requires more details, we would refer them to our previously mentioned publications.

Moving to the high-level details, Zernike-based simulation wishes to simulate the familiar image formation process

$$I_i(\mathbf{x}) = \left(|\mathfrak{F}\text{ourier}\{P(\xi)e^{j\phi_{\mathbf{u}}(\xi)}\}|^2 \overset{\mathbf{u}}{\circledast} I_g \right) (\mathbf{x}), \quad (4.1)$$

where $\phi_{\mathbf{u}}(\xi)$ is a spatially varying phase realization that arises from the propagation. It is important to note that a phase realization (or phase function) is different from a phase screen, therefore we will reserve the term “phase screen” to describe the phase screens used in split-step. The phase function is defined across the aperture, whereas a phase screen represents a section of the atmosphere along the propagation path.

These phase realizations have two important properties in the case of atmospheric turbulence: (1) Their structure function is described through the Kolmogorov model (or its related models); (2) They are correlated spatially. Our goal, as previously stated, is to maintain these two properties *without* numerical wave propagation. For each point \mathbf{u} in an image, there is an associated point spread function

$$|h_{\mathbf{u}}(\mathbf{x})|^2 = \left| \mathfrak{F}\text{ourier} \left\{ e^{j\phi_{\mathbf{u}}(\xi)} \right\} \right|^2, \quad (4.2)$$

4.2. A SURVEY OF PROPAGATION-FREE METHODS

where we have dropped some constants for brevity. This means for each point in the image, there is a corresponding phase function $\phi_{\mathbf{u}}(\xi)$ and point spread function $|h_{\mathbf{u}}(\mathbf{x})|^2$. Using Equation (4.2) to interpret our desired statistical properties, if we can generate the per-pixel phase functions $\phi_{\mathbf{u}}(\xi)$ and $\phi_{\mathbf{u}'}(\xi)$ from the specified distribution then we will have achieved our goal.

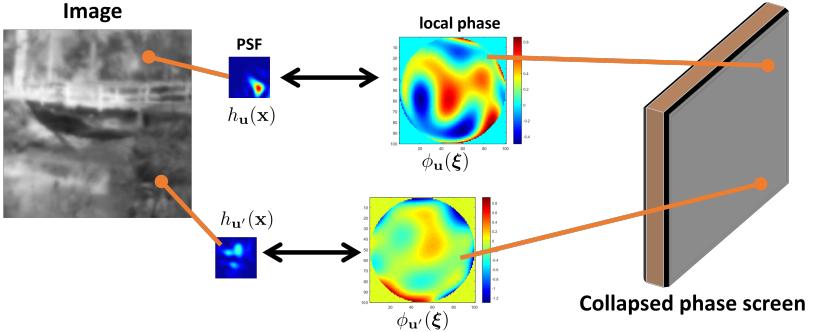


Figure 4.5: The collapsed phase-over-aperture model by Chimitt and Chan [41]. The idea is to compress the propagation path into a single abstract volume where each location has its own phase function and corresponding PSF. We note this differs conceptually from a phase screen.

The above argument leads to a conceptual diagram shown in Figure 4.5. In this figure, the information contained in the set of phase realizations is embedded into the **Zernike space** (referred to as a collapsed phase screen in [41]). For each location \mathbf{u} in the Zernike space there is a phase realization $\phi_{\mathbf{u}}(\mathbf{f})$, related to the PSFs $|h_{\mathbf{u}}(\mathbf{x})|^2$ by Equation (4.2). The resulting image can be formed by performing a spatially varying convolution:

$$I(\mathbf{x}) = |h_{\mathbf{u}}(\mathbf{x})|^2 \stackrel{\mathbf{u}}{\circledast} I_g(\mathbf{x}), \quad (4.3)$$

where $I_g(\mathbf{x})$ is the geometric image and $I(\mathbf{x})$ is the turbulence distorted image.

This Chapter serves to address the following questions which must be answered to enable this variant of simulation:

1. How do we generate the phase functions $\phi_{\mathbf{u}}(\xi)$ at each \mathbf{u} ? This is the premise of the Zernike space model. If we cannot gener-

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

ate the phase functions $\phi_{\mathbf{u}}(\xi)$, then there is no way that we can construct the PSFs $h_{\mathbf{u}}(\mathbf{x})$. We shall address this problem using a representation technique known as the **Zernike polynomials**.

2. How do we enforce the spatial correlation of two phase functions $\phi_{\mathbf{u}}(\xi)$ and $\phi_{\mathbf{u}'}(\xi)$? Again, this is a critical part of the method. If we cannot draw spatially correlated phase functions, we will defeat the purpose of skipping the wave propagation steps. To articulate this question, we will introduce few powerful results in the literature about the angle of arrival statistics, together with a mirroring technique, to develop a so-called **multi-aperture approach**.
3. Given that we may develop the model to represent turbulent distortions, how should we implement the method to be *fast*? Our main issue with split-step is its speed, therefore, we'd want Zernike-based simulation to give us a significant increase in speed.

Thus, Chapter 4.3 and Chapter 4.4 will be dedicated to these more theoretical considerations while Chapter 4.5 addresses the final question.

4.3 Representing Turbulent Phase Realizations

4.3.1 Phase as a Basis

The last Chapter presented two ways to represent the phase:

$$\begin{aligned}\phi(\xi) &= k \int_0^L n_1(\xi, z) dz && \text{(the definition),} \\ \phi(\xi) &= \mathfrak{Fresnel}[L/N] \{ e^{\phi_N} \mathfrak{Fresnel}[L/N] \{ \dots \} \} && \text{(via split-step),}\end{aligned}$$

with L as the propagation distance and N as the number of phase screens in split-step. Note that the split-step propagation expression is a sequential application of free-space propagation followed by phase imparting. Both of these representations were utilized in the last Chapter, though they present a few challenges: (1) lack of clarity for informing a prior model; (2) the statistical properties of each are a bit unfamiliar to the standard computer vision researcher, such as the structure function; (3) the extension to multiple phase realizations is unclear.

This leads us to instead formulate the phase in a different way through the concept of **basis representation** of a function. A basis

4.3. REPRESENTING TURBULENT PHASE REALIZATIONS

representation allows us to write $\phi_{\mathbf{u}}(\xi)$ as

$$\phi_{\mathbf{u}}(\xi) = \sum_{j=1}^N a_{\mathbf{u},j} Z_j(\xi), \quad (4.4)$$

for some basis functions $\{Z_j(\xi) | j = 1, \dots, N\}$ and some basis coefficients $\{a_{\mathbf{u},j} | j = 1, \dots, N\}$. Equations of this form are ubiquitous in computer vision. If we recall the classical Fourier expansion of a function, a continuously differentiable function $f(t)$ with bounded energy can be written as a linear combination of complex exponentials: $f(t) = \sum_{n=1}^N a_n Z_n(t)$, where $Z_n(t) = \exp\{j2\pi n t/N\}$ is the Fourier basis, and $a_j = \langle f(t), Z_n(t) \rangle$ is the Fourier coefficient. The benefit of expressing the function $f(t)$ in terms of the Fourier basis is that it allows us to describe the function using a low-dimensional representation. Thus, if we can generate the coefficients $\{a_n | n = 1, \dots, N\}$, we can represent the function $f(t)$.

Going back to the phase function $\phi_{\mathbf{u}}(\xi)$, the function $Z_j(\xi)$ is the basis representation of the phase. Unlike the phase function $\phi_{\mathbf{u}}(\xi)$ which is location dependent (hence spatially varying), the basis function $Z_j(\xi)$ is spatially *invariant*. Two phase functions $\phi_{\mathbf{u}}(\xi)$ and $\phi_{\mathbf{u}'}(\xi)$ differ in their basis representation only by the coefficients $a_{\mathbf{u},j}$ and $a_{\mathbf{u}',j}$. Addressing the notation $a_{\mathbf{u},j}$, the first index \mathbf{u} denotes the location whereas the second index j tells us which basis function are we using. We will refer to the vector $\mathbf{a}_{\mathbf{u}} = [a_{\mathbf{u},1}, \dots, a_{\mathbf{u},N}]^T$ to describe the vector of basis coefficients at location \mathbf{u} .

4.3.2 Zernike Basis

How do we pick an appropriate basis function $Z_j(\xi)$? The phase function $\phi_{\mathbf{u}}(\xi)$ is defined over the aperture, and the aperture is usually circular in shape. As such, the basis functions $Z_j(\xi)$ must also be a function defined over a circle.

Our choice of the basis function $Z_j(\xi)$ is the **Zernike polynomials**. The Zernike polynomials are two-dimensional orthogonal functions defined over the unit disk. The Zernike polynomials are mathematically well defined and there are tools to handle these mathematical objects. Furthermore, the work of Noll [39] utilized the Zernike polynomials to represent the phase with some important additional comments and small corrections by Roddier [40] and Wang et al. [102]. Noll's paper represents the starting point of the simulator developed by the authors and collaborators.

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

To work with Zernike polynomials, it is more convenient to switch from Cartesian coordinate $\xi = [\xi_x, \xi_y]^T$ to the polar coordinate $\rho = [r, \theta]^T$, where $\xi_x = r \cos \theta$ and $\xi_y = r \sin \theta$. Moreover, with Zernike polynomials, we prefer to work with unit norm vectors because the domain of the Zernike polynomials is the unit disk. To this end, we define $R \stackrel{\text{def}}{=} D/2$ as the aperture radius and scale r to $r = R\rho$ so that $0 \leq \rho \leq 1$. With these notations, the phase function $\phi(\xi)$ can be written as $\phi(R\rho)$ where $\rho = [\rho, \theta]^T$ and $R\rho = [R\rho, \theta]^T$. In terms of mathematical expressions, we can write

$$\underbrace{\phi_u(R\rho)}_{=\phi_u(\xi)} = \sum_{j=1}^N a_{u,j} \underbrace{Z_j(\rho)}_{=Z_j(\xi)}. \quad (4.5)$$

Before we look at the equations of the Zernike polynomials, we first show the shape in Figure 4.6. Among the different orders, the base $Z_0(\rho)$ is the constant offset which can be neglected for incoherent imaging. The first and the second Zernike polynomials $Z_1(\rho)$ and $Z_2(\rho)$ are known as the **x-tilt** and the **y-tilt** respectively. It is worth noting in some literature these are referred to as tip and tilt, respectively, though for clarity we prefer the former terminology. Other high order Zernike polynomials include astigmatism, coma, spherical aberrations, etc.

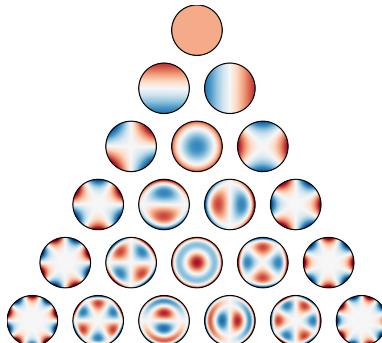


Table 1. Zernike polynomial up to and including the 4th radial order.

Order (n)	Freq (m)	$Z_n^m(\rho, \theta)$	Meaning
0	0	1	Constant term, or Piston
1	-1	$2\rho \sin(\theta)$	x-tilt, Distortion
1	1	$2\rho \cos(\theta)$	y-tilt, Distortion
2	-2	$\sqrt{5}\rho^2 \sin(2\theta)$	Astigmatism (-45°)
2	0	$\sqrt{3}(2\rho^2 - 1)$	Field curvature, Defocus
2	2	$\sqrt{6}\rho^2 \cos(2\theta)$	Astigmatism (0° or 90°)
3	-3	$\sqrt{8}\rho^3 \sin(3\theta)$	y-trefoil
3	-1	$\sqrt{8}(3\rho^3 - 2\rho) \sin(\theta)$	y-coma
3	1	$\sqrt{8}(3\rho^3 - 2\rho) \cos(\theta)$	x-coma
3	3	$\sqrt{8}\rho^3 \cos(3\theta)$	x-trefoil
4	-4	$\sqrt{10}(4\rho^4 - 3\rho^2) \sin(4\theta)$	y-quadrupole
4	-2	$\sqrt{10}(4\rho^4 - 3\rho^2) \sin(4\theta)$	y-secondary astigmatism
4	0	$\sqrt{5}(6\rho^4 - 6\rho^2 + 1) \sin(4\theta)$	Spherical, Defocus
4	2	$\sqrt{10}(4\rho^4 - 3\rho^2) \cos(4\theta)$	x-secondary astigmatism
4	4	$\sqrt{10}\rho^4 \cos(4\theta)$	x-quadrupole
...

Figure 4.6: The shape of the Zernike basis functions, and their corresponding equations. Source: https://en.wikipedia.org/wiki/Zernike_polynomials.

As indicated by the table in Figure 4.6, the Zernike polynomials are indexed by their **azimuthal** order m and their **radial** order n .

4.3. REPRESENTING TURBULENT PHASE REALIZATIONS

Thus, the pair (m, n) specifies the particular Zernike polynomial we are describing; m tells you how far out from the “center line” to go and n tells you how many rows down from the top. This leads to the notation $Z_n^m(\boldsymbol{\rho})$, however, it is often easier to work with a single index j with notation $Z_j(\boldsymbol{\rho})$ as we’ve written previously. In this book, we follow Noll’s paper [39] that maps $(m, n) \rightarrow j$ whose formula is skipped for brevity. With this mapping, the Zernike polynomial is defined as (recall that $\boldsymbol{\rho} = [\rho, \theta]$):

$$\begin{aligned} Z_{\text{even } j}(\boldsymbol{\rho}) &= \sqrt{2(n+1)} R_n^m(\rho) \cos(m\theta), & m \neq 0, \\ Z_{\text{odd } j}(\boldsymbol{\rho}) &= \sqrt{2(n+1)} R_n^m(\rho) \sin(m\theta), & m \neq 0, \\ Z_j(\boldsymbol{\rho}) &= \sqrt{n+1} R_n^m(\rho) \sin(m\theta), & m = 0, \end{aligned}$$

where $R_n^m(\rho)$ is

$$R_n^m(\rho) = \sum_{s=0}^{(n-m)/2} \frac{(-1)^s (n-s)!}{s! [(n+m)/2-s]! [(n-m)/2-s]!} \rho^{n-2s}.$$

There are many other such indexing, thus we do not feel it necessary to present exactly how the mapping is performed. Though, it is important to re-emphasize that in this book we indeed use Noll’s indexing scheme as is common in the atmospheric turbulence literature.

4.3.3 Zernike Coefficients

The Zernike polynomials are defined over the unit circle, therefore, the inner product must also be defined on the unit circle. To facilitate our discussions, we denote the circular aperture as

$$P(\boldsymbol{\rho}) = \begin{cases} 1, & |\boldsymbol{\rho}| \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

With this circular aperture defined, we can then consider the orthogonality principle over the unit circle. If we are given a phase function $\phi_{\mathbf{u}}(\boldsymbol{\rho})$, we can define the j th Zernike coefficient at location \mathbf{u} as

$$\begin{aligned} a_{\mathbf{u},j} &= \langle \phi_{\mathbf{u}}(R\boldsymbol{\rho}), Z_j(\boldsymbol{\rho}) \rangle_P \\ &\stackrel{\text{def}}{=} \int_0^{2\pi} \int_0^1 P(\boldsymbol{\rho}) \phi_{\mathbf{u}}(R\boldsymbol{\rho}) Z_j(\boldsymbol{\rho}) d\rho d\theta, \end{aligned} \quad (4.6)$$

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

where $\langle \cdot, \cdot \rangle_P$ denotes the inner product using $P(\rho)$ as a weight. Statistically, Tatarskii [22] argued that $\phi_{\mathbf{u}}(R\rho)$ is zero-mean Gaussian. This is partially a physics observation, and partially for mathematical convenience. If we agree with Tatarskii's assumption, then the Zernike coefficients $\{a_{\mathbf{u},j} | j = 1, \dots, N\}$ are zero-mean Gaussians because they are obtained through linear projections.

Generating the Zernike polynomials and subsequently the phase function can be done on a computer using customized libraries. In the example below, we use the Zernike polynomial MATLAB code developed by Gray [103].

Example. To make this example simple, we fix a location \mathbf{u} . We randomly draw 36 *independent* Gaussian coefficients $\{a_{\mathbf{u},j}\}_{j=1}^{36}$ from $a_{\mathbf{u},j} \sim \mathcal{N}(0, 1/\sqrt{36})$. The example here is merely for demonstration of how the coefficients can affect the shape of the phase.

The phase $\phi_{\mathbf{u}}(R\rho)$ is generated using `ZernikeCalc` [103]. Figure 4.7 illustrates one random realization of the phase distortion function $\phi_{\mathbf{u}}(R\rho)$ and the corresponding PSF $h_{\mathbf{u}}(\mathbf{x})$. The MATLAB code is shown below. Here, we over-sample the Fourier spectrum using $\text{fftK} = 8$ for anti-aliasing. The division $\text{ph}/2$ is due to the fact that the diameter is 2.

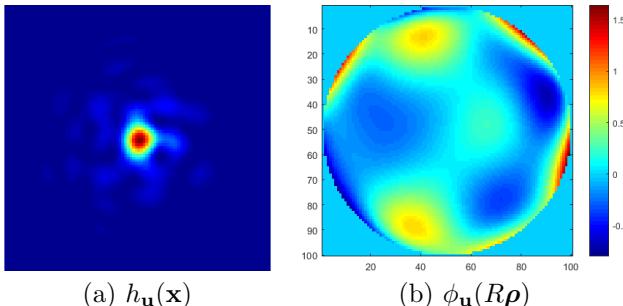


Figure 4.7: The random phase distortion (b) generated from the random Zernike polynomial in a grid of 100×100 (corresponding to a disk of diameter 2), and its associated PSF (a).

We emphasize that the phase function and the point spread function shown in 4.7 do not correspond to a wave propagated through

4.3. REPRESENTING TURBULENT PHASE REALIZATIONS

Kolmogorov turbulence. The reason is that the coefficients are *independently* sampled – we haven't put the Kolmogorov model anywhere! In reality, we shall sample the Zernike coefficients according to some known statistics which arise from the model of Chapter 3. Only then will the generated phase realizations and associated PSFs have the desired properties.

```
N = 100;
fftK = 8;
[P, ~] = ZernikeCalc(1,1);
[ph, ~] = ZernikeCalc(1:36, (1/6)*randn(36,1), ...
N, 'STANDARD');
U = exp(1i*ph/2).*P;
u = abs(fftshift(ifft2(U, N*fftK, N*fftK).^2));
```

4.3.4 Intermodal Correlation

As mentioned previously, Noll studied the use of the Zernike polynomials in representing a turbulent phase distortion [39]. Specifically, Noll asked the question: For a phase realization $\phi_{\mathbf{u}}(R\rho)$, what is the correlation $\mathbb{E}[a_{\mathbf{u},i}a_{\mathbf{u},j}]$? Due to the terminology of “modes”, this is referred to as **intermodal correlation**. This allows one to write the covariance matrix of the basis coefficients for a single phase realization, and thus we can draw phase realizations that *are* in accordance with Kolmogorov's theory. We additionally note that there exists another possible correlation, which is for two *distinct* points \mathbf{u} and \mathbf{u}' , we could ask the **spatial correlation** of $\mathbb{E}[a_{\mathbf{u},i}a_{\mathbf{u}',i}]$. We illustrate the difference in Figure 4.8.

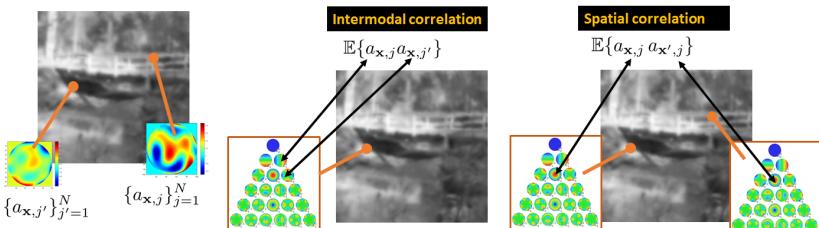


Figure 4.8: Intermodal and spatial correlation of the Zernike coefficients.

Recalling Equation (4.6), we can show that the joint expectation

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

$\mathbb{E}[a_{\mathbf{u},i} a_{\mathbf{u},j}]$ (at the same location \mathbf{u}) is:

$$\begin{aligned}\mathbb{E}[a_{\mathbf{u},i} a_{\mathbf{u},j}] &= \frac{1}{\pi^2} \iint P(\boldsymbol{\rho})P(\boldsymbol{\rho}')Z_i(\boldsymbol{\rho})Z_j(\boldsymbol{\rho}') \\ &\quad \times \mathbb{E}[\phi_{\mathbf{u}}(R\boldsymbol{\rho})\phi_{\mathbf{u}}(R\boldsymbol{\rho}')] d\boldsymbol{\rho}d\boldsymbol{\rho}'.\end{aligned}\quad (4.7)$$

The main result of Noll's paper is to evaluate Equation (4.7). Doing so through the Fourier Transform and usage of the Kolmogorov spectrum, the joint expectation can be explicitly evaluated as [40]:

$$\begin{aligned}\mathbb{E}[a_{\mathbf{u},i} a_{\mathbf{u},j}] &= 2.2698(-1)^{(n_i+n_j-2m)/2} \sqrt{(n_i+1)(n_j+1)} \mathbb{I}_{m_i m_j} \\ &\quad \times \frac{(D/r_0)^{5/3}}{\Gamma[(n_i+n_j+23/3)/2]} \\ &\quad \times \frac{\Gamma[(n_i+n_j-5/3)/2]}{\Gamma[(n_j-n_i+17/3)/2]\Gamma[(n_i-n_j+17/3)/2]},\end{aligned}\quad (4.8)$$

if $i - j = \text{even}$ and $\mathbb{E}[a_{\mathbf{u},i} a_{\mathbf{u},j}] = 0$ if $i - j = \text{odd}$. The indices m_i and n_i are the azimuthal and radial orders associated with the i th Zernike polynomial, and $\mathbb{I}_{m_i m_j}$ is an indicator function that gives $\mathbb{I}_{m_i m_j} = 1$ if $m_i = m_j$ and $\mathbb{I}_{m_i m_j} = 0$ if $m_i \neq m_j$. The normalized correlation matrix, which we will refer to as the **Noll matrix**, is shown in Figure 4.9, where we note most of the other entries are *sparse*.

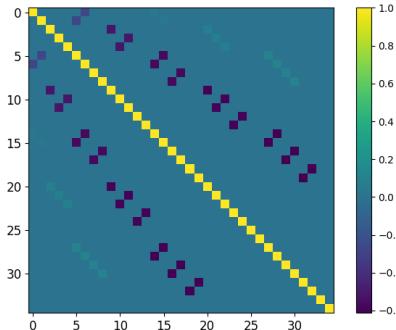


Figure 4.9: Inter-modal correlation of the Zernike coefficients $\mathbb{E}[a_{\mathbf{u},i} a_{\mathbf{u},j}]/\sqrt{\mathbb{E}[a_{\mathbf{u},i}^2]\mathbb{E}[a_{\mathbf{u},j}^2]}$ for a fixed spatial location \mathbf{u} .

Through standard random sampling methods, we can properly sample individual turbulent phase realizations. Consider a multivariate Gaussian random variable with mean $\mathbb{E}[\mathbf{y}] = \mathbf{0}$ and the correlation

4.3. REPRESENTING TURBULENT PHASE REALIZATIONS

matrix $\Sigma = \mathbb{E}[\mathbf{y}\mathbf{y}^T]$. In our case, Σ will represent the Noll matrix. To draw a random vector \mathbf{y} from this distribution, we decompose the correlation matrix $\Sigma = \mathbf{L}\mathbf{L}^T$ via **Cholesky decomposition**. Then, starting with a white noise vector $\mathbf{n} \sim \text{Gaussian}(\mathbf{0}, \mathbf{I})$, the transformed vector $\mathbf{y} = \mathbf{L}\mathbf{n}$ will satisfy the desired property that $\mathbb{E}[\mathbf{y}] = \mathbf{0}$ with covariance:

$$\begin{aligned}\mathbb{E}[\mathbf{y}\mathbf{y}^T] &= \mathbb{E}[\mathbf{L}\mathbf{n}\mathbf{n}^T\mathbf{L}], \\ &= \mathbf{L}\mathbb{E}[\mathbf{n}\mathbf{n}^T]\mathbf{L}, \\ &= \Sigma.\end{aligned}$$

The Cholesky decomposition is a standard method in various packages such as MATLAB or NumPy/PyTorch/etc., and thus is readily applied to the Noll matrix.

The Noll matrix contains off-diagonal entries. One may wonder if we may diagonalize the matrix by an eigen-decomposition. This is possible, however, the matrix is *paired* with the fact that it describes the coefficients for our Zernike basis. If we wish to diagonalize the matrix, we will require different basis functions. This type of decomposition is known as the **Karhunen–Loëve (KL)** decomposition where the bases are the eigenfunctions of the covariance function. This has been considered in papers such as Roddier [40]. However, this was also done earlier *by hand* for analyzing the probability of getting a lucky observation by Fried [27]. The KL basis functions are in non-closed form and accordingly more difficult to work with, therefore, we choose to continue with the Zernike polynomials.

4.3.5 What's Missing?

If we wish to describe or simulate the effects on an object observed through turbulence, it may be tempting to jump to the conclusion that we are ready to do so with Noll's result. However, we should recall that in the case of turbulent PSFs which span an image we require a spatially variant representation,

$$|h_{\mathbf{u}}(\mathbf{x})|^2 = \left| \mathfrak{F}\text{ourier}\{P(\rho)e^{j\phi_{\mathbf{u}}(R\rho)}\} \right|_{\mathbf{f}=\mathbf{x}/(\lambda z)}^2. \quad (4.9)$$

Here we emphasize the spatial dependency of the phase realizations. As a result, we must have *multiple* realizations of $\phi_{\mathbf{u}}$ due to the dependency on the source location. Noll's result does not allow such a possibility because the spatial correlation is missing!

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

The work of Noll is a great starting point, however, more work will be required to draw Zernike samples that are sampled properly in both intermodal and spatial axes. If we can find this representation, then we may successfully cut out the numerical wave propagation from the simulation procedure and allow us to generate samples much faster.

4.4 The Zernike Space

This leads us to formally introduce the Zernike space. The Zernike space generalizes the concept of a basis function for a single phase realization to a **random vector field** which is a collection of basis coefficient vectors. While that is easy to say, proper sampling will prove to be computationally infeasible at present [43]. We will set this issue aside for now and focus on the concept itself, concluding this discussion with these computational issues and an approximation that avoids them.

4.4.1 Overview

The Zernike space arises from the consideration that for each point in the object plane, there exists a phase realization $\phi_{\mathbf{u}}$, each with a Zernike representation $\mathbf{a}_{\mathbf{u}}$. Furthermore, each phase realization describes the PSF $|h_{\mathbf{u}}(\mathbf{x})|^2$. However, by our basis representation, the vector $\mathbf{a}_{\mathbf{u}}$ is also sufficient in describing the PSF. We refer to this set $\{\mathbf{a}_{\mathbf{u}}\}$ as the Zernike space.

We present a visualization of the Zernike space in Figure 4.10. For each point in the image, there exists a vector that is correlated intermodally and spatially to its neighbors. Thus, the Zernike space is a random vector field. The two axes of the random fields are spatial axes while the third axis is the Zernike coefficient axis. Thus, it is incorrect to say that the Zernike space is a volume, as the notion of locality is lost in the coefficient axis. It is this problem that will give rise to a majority of our issues when attempting to generate realizations of the Zernike space.

The Zernike space allows us to simply state the model that captures all of the turbulent statistics *without* having to describe the propagation. Because the turbulence is distributed as a random process, the Zernike space will accordingly be random. The way we will need to describe the Zernike space will be through its correlations. This will

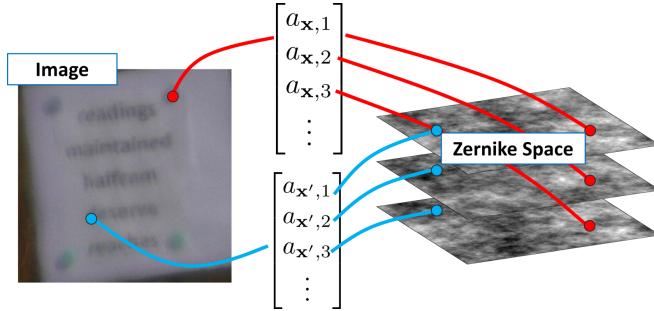


Figure 4.10: Visualization of the Zernike space. For each point in the image, there exists a Zernike vector, which further defines a vector field.

require work to “bake in” the required properties, though the gain will be eliminating numerical wave propagation in our computation.

4.4.2 Mirroring

A critical piece of information that would be needed to enable the Zernike-based approach is the **spatial correlation** of the basis coefficients. That is, for a fixed Zernike polynomial order i , we want to find the joint expectation $\mathbb{E}[a_{\mathbf{u},i} a_{\mathbf{u}',i}]$ for two different coordinates \mathbf{u} and \mathbf{u}' . The derivation here is through an approximate analysis by the authors [41] which is conceptually a bit easier to understand. We later present an overview of a more general solution.

To begin our analysis, we first generalize the structure function. To begin, let us recall the structure function, though slightly re-arranged:

$$\mathcal{D}_\phi(R\rho - R\rho') = 2.91k^2 \int_0^L C_n^2(z) \left| R(\rho - \rho') \left(1 - \frac{z}{L}\right) \right|^{5/3} dz. \quad (4.10)$$

Note Equation (4.10) is equivalent to our continuous spherical wave structure function definition from Equation (3.24). The structure function $\mathcal{D}_\phi(R\rho - R\rho')$ is defined just by the two coordinates $R\rho$ and $R\rho'$ which are two points of the *same* phase function $\phi_\mathbf{u}(R\rho)$. However, what we are interested in is the pair $\phi_\mathbf{u}(R\rho)$ and $\phi_{\mathbf{u}'}(R\rho)$. Therefore, a key step here is to redefine the structure function $\mathcal{D}_\phi(R\rho - R\rho')$ so that it depends on \mathbf{u} and \mathbf{u}' . This will allow us to model the case of two separate phase realizations.

To tackle the problem, we start by considering the geometry shown in Figure 4.11. The problem of interest is the one shown in Figure 4.11(a) where we use one aperture to observe two different points in

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

the scene. The separation of the two points in the scene is $\mathbf{u} - \mathbf{u}'$. If the distance from the aperture to the object plane is L , the normal vector pointing out of the image plane will form an angle with \mathbf{u} (and another angle with \mathbf{u}'). These two angles are the **angle-of-arrival**. Though the angles are typically of interest, we prefer to consider the separation in the object plane $\mathbf{u} - \mathbf{u}'$. Let us then extend the structure function to include a separation in the object plane, as well as the aperture plane. This leads us to define a more general structure function [104, 34],

$$\begin{aligned}\mathcal{D}(R\rho - R\rho', \mathbf{u} - \mathbf{u}') &= 2.91k^2 \int_0^L dz C_n^2(z) \\ &\times \left| R(\rho - \rho') \left(\frac{L-z}{L} \right) + z/L(\mathbf{u} - \mathbf{u}') \right|^{5/3}.\end{aligned}\quad (4.11)$$

The magnitude term within the integrand can be understood through simple geometry: it is an interpolation between the vector characterizing the separation in the object plane to the difference vector in the aperture plane.

Using geometric optics and symmetry, the roles of the object plane and the aperture can be flipped as shown in Figure 4.11(b). We refer to such a technique as **mirroring**. In this new configuration, the separation $\mathbf{u} - \mathbf{u}'$ is translated to the separation of two points in the aperture plane. Assuming that each aperture has a diameter D , we can define a vector \mathbf{s} such that

$$Ds = \mathbf{u} - \mathbf{u}', \quad (4.12)$$

according to the symmetry of the mirroring. The vector \mathbf{s} tells us how many units of D is needed for the separation of two points.

Readers at this point may wonder: Why construct such a mirror and map the original problem to the mirrored space? The short answer is that the spatial correlation in Figure 4.11(b) has been thoroughly studied whereas the original one in Figure 4.11(a) is not. In the prior work of Chanan [105] and Takato and Yamaguchi [106], it was shown that if there are two *different* imaging systems looking at one point in the space (i.e., Figure 4.11(b)), there exists closed-form expressions of the spatial correlation (of the Zernike coefficients). They were interested in this problem because for astronomy, large telescope cameras with **multiple apertures** are used to observe stars. Therefore, if we can find this mapping, we can simply leverage the existing results of Chanan [105] and Takato and Yamaguchi [106].

4.4. THE ZERNIKE SPACE

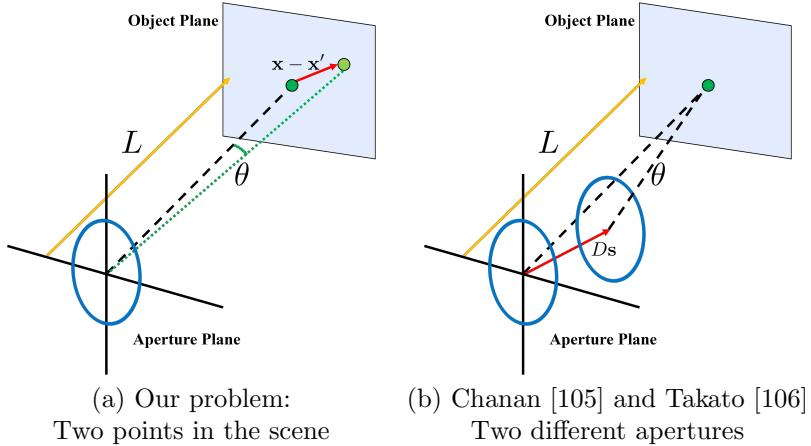


Figure 4.11: Visualization of the geometries of the two types of problems. The problem we are primarily concerned with is shown in (a), however, we would like to leverage the results of analysis of the geometry considered in (b).

Going back to the spatial correlation, our intention is to replace $\mathbb{E}[\phi_{\mathbf{u}}(R\rho)\phi_{\mathbf{u}'}(R\rho')]$ with $\mathbb{E}[\phi_{\mathbf{u}}(R\rho)\phi_{\mathbf{u}'}(R\rho')]$. We will then leverage the two aperture results to solve our problem. Beginning with the expectation, we may write

$$\begin{aligned}\mathbb{E}[a_{\mathbf{u},i} a_{\mathbf{u}',j}] &= \frac{1}{\pi^2} \iint P(\rho)P(\rho')Z_i(\rho)Z_j(\rho') \\ &\quad \times \mathbb{E}[\phi_{\mathbf{u}}(R\rho)\phi_{\mathbf{u}'}(R\rho')] d\rho d\rho'.\end{aligned}\quad (4.13)$$

Through the definition of the structure function, we may write this as

$$\begin{aligned}\mathbb{E}[a_{\mathbf{u},i} a_{\mathbf{u}',j}] &= \frac{-1}{2\pi^2} \iint d\rho d\rho' P(\rho)P(\rho')Z_i(\rho)Z_j(\rho') \\ &\quad \times \mathcal{D}(R\rho - R\rho', \mathbf{u} - \mathbf{u}').\end{aligned}\quad (4.14)$$

Interestingly, it was shown in [41] that if a first-order Taylor approximation was made upon the magnitude expression at $z = L/2$, combined with an assumption of a constant C_n^2 profile, then the expression may

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

be written through mirroring as

$$\begin{aligned}\mathbb{E}[a_{\mathbf{u},i}a_{\mathbf{u}',j}] &= \frac{L}{2^{5/3}\pi^2} \iint P(\boldsymbol{\rho})P(\boldsymbol{\rho}') \\ &\quad \times Z_i(\boldsymbol{\rho})Z_j(\boldsymbol{\rho}')\mathbb{E}[\phi_{\mathbf{u}}(R\boldsymbol{\rho})\phi_{\mathbf{u}}(R\boldsymbol{\rho}' + D\mathbf{s})] d\boldsymbol{\rho}d\boldsymbol{\rho}'.\end{aligned}\quad (4.15)$$

The impact of the mirroring is that the two phase functions $\phi_{\mathbf{u}}(R\boldsymbol{\rho})$ and $\phi_{\mathbf{u}'}(R\boldsymbol{\rho}')$ are now changed to $\phi_{\mathbf{u}}(R\boldsymbol{\rho})$ and $\phi_{\mathbf{u}}(R\boldsymbol{\rho}' + D\mathbf{s})$. The latter is due to the mirroring analysis.

The result of this is that the problem of angle-of-arrival correlation has been mapped to the multiple aperture problem. Therefore, the results of Chanan [105] and Takato and Yamaguchi [106] can be applied directly for our case. Using the Kolmogorov power spectral density results in an expression that we present in the following theorem:

Theorem 4.1 (Zernike Spatial Correlation via Multi-Aperture).

Let \mathbf{u} and \mathbf{u}' be two points in the object plane. Consider a two-aperture model with $D\mathbf{s} = \mathbf{u} - \mathbf{u}'$ where D is the aperture diameter, and \mathbf{s} is a vector defining the separation. The spatial correlation of two Zernike coefficients $a_{\mathbf{u},i}$ and $a_{\mathbf{u}',j}$ is given by

$$\begin{aligned}\mathbb{E}[a_{\mathbf{u},i}a_{\mathbf{u}',j}] &\approx \frac{-2.91k^2C_n^2L}{2\pi^22^{5/3}} \iint P(\boldsymbol{\rho})P(\boldsymbol{\rho}')Z_i(\boldsymbol{\rho})Z_j(\boldsymbol{\rho}') \\ &\quad \times |R(\boldsymbol{\rho} - \boldsymbol{\rho}') + D\mathbf{s}|^{5/3} d\boldsymbol{\rho}d\boldsymbol{\rho}',\end{aligned}\quad (4.16)$$

4.4.3 Spatial Correlation of the Tilts

Theorem 4.1 is a general result for *all* Zernike coefficients. To gain some insights into how this equation can be used to draw random turbulence samples, we focus on two Zernike coefficients: the horizontal tilt (characterized by $a_{\mathbf{u},2}$) and the vertical tilts (characterized by $a_{\mathbf{u},3}$). We are interested in these two coefficients because the tilts occupy the majority of the energy in the Zernike basis representation. In addition, the tilt statistics are slightly more manageable to track analytically.

When restricted to $a_{\mathbf{u},2}$ and $a_{\mathbf{u},3}$, the Zernike basis $Z_2(\boldsymbol{\rho})$ and $Z_3(\boldsymbol{\rho})$ will take a relatively simple form. Using Chanan's result [105, Eq. 11], we can show that for $j \in \{2, 3\}$, the spatial correlation is given as follows.

Theorem 4.2 (Spatial correlation of tilts). *The spatial correlation of the 2nd and the 3rd Zernike coefficients (the horizontal and the vertical tilts) are approximated by*

$$\mathbb{E}[a_{\mathbf{u},j} a_{\mathbf{u}',j}] \approx \frac{c_2}{2^{5/3}} \left(\frac{D}{r_0} \right)^{5/3} [I_0(s) \mp \cos 2\psi_0 I_2(s)], \quad j = 2, 3, \quad (4.17)$$

where $c_2 = 7.7554$, and the minus sign is for $a_{\mathbf{u},2}$ and the plus sign is for $a_{\mathbf{u},3}$. The coordinate $[s, \psi_0]$ is illustrated in Figure 4.12. The length of $D\mathbf{s} = \mathbf{u} - \mathbf{u}'$ is the scalar Ds , where $s = |\mathbf{s}|$ and the angle is ψ_0 .

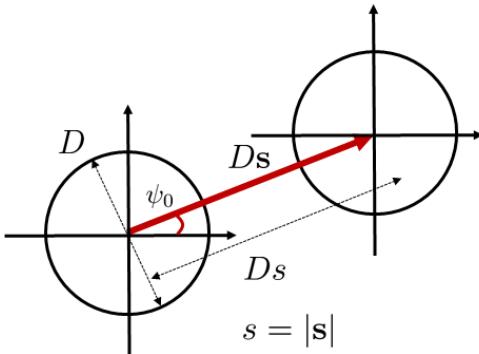


Figure 4.12: Notation of the coordinates. The magnitude of ξ is measured in the unit of the diameter D .

The integrals $I_0(s)$ and $I_2(s)$ in Equation (4.17) are defined through the Bessel functions of the first kind:

$$\mathcal{J}_0(s) = \int_0^\infty \zeta^{-14/3} J_0(2s\zeta) J_2^2(\zeta) d\zeta, \quad (4.18)$$

$$I_2(s) = \int_0^\infty \zeta^{-14/3} J_2(2s\zeta) J_2^2(\zeta) d\zeta, \quad (4.19)$$

where J_0 and J_2 are the Bessel functions of the first kind. On computers, these can be defined using built-in libraries. In MATLAB, for example, we can use commands `besselj` and `integral` as shown below.

```
% MATLAB code to evaluate integrals I_0 and I_2
s    = linspace(0,smax,N);
```

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

```

f = @(z) z^(-14/3)*besselj(0,2*s*z)*...
        besselj(2,z)^2;
I0 = integral(f, 1e-8, 1e3, 'ArrayValued', true);
g = @(z) z^(-14/3)*besselj(2,2*s*z)*...
        besselj(2,z)^2;
I2 = integral(g, 1e-8, 1e3, 'ArrayValued', true);

```

How do we draw the random tilts then? Considering Equation (4.17) again, we recognize that it defines the correlation matrix.

$$[\mathbf{C}_j]_{\mathbf{u}, \mathbf{u}'} = \frac{\mathbb{E}[a_{\mathbf{u}, j} a_{\mathbf{u}', j}]}{\mathbb{E}[a_{\mathbf{u}, j} a_{\mathbf{u}, j}]} = \frac{I_0(s) \mp \cos(2\psi_0) I_2(s)}{I_0(0)}, \quad j = 2, 3. \quad (4.20)$$

In this equation, $[\mathbf{C}_j]_{\mathbf{u}, \mathbf{u}'}$ denotes the correlation matrix \mathbf{C}_j for $j = 2, 3$. The subscript $(\mathbf{u}, \mathbf{u}')$ denotes that we are looking at the coordinate $(\mathbf{u}, \mathbf{u}')$. Since the right-hand side of the correlation matrix is described by the polar coordinate $[s, \psi_0]$ of the difference vector $\mathbf{s} = (\mathbf{u} - \mathbf{u}')/D$, the matrix \mathbf{C}_j is Toeplitz. This should not be a surprise because the correlation matrix is **homogeneous** (aka wide-sense stationary).

A visualization of the correlation matrices $[\mathbf{C}_2]_{\mathbf{u}, \mathbf{u}'}$ and $[\mathbf{C}_3]_{\mathbf{u}, \mathbf{u}'}$ is shown in Figure 4.13. The matrices demonstrate a Toeplitz block structure because the correlation is two-dimensional. Nevertheless, the Toeplitz structure is preserved due to its homogeneity.

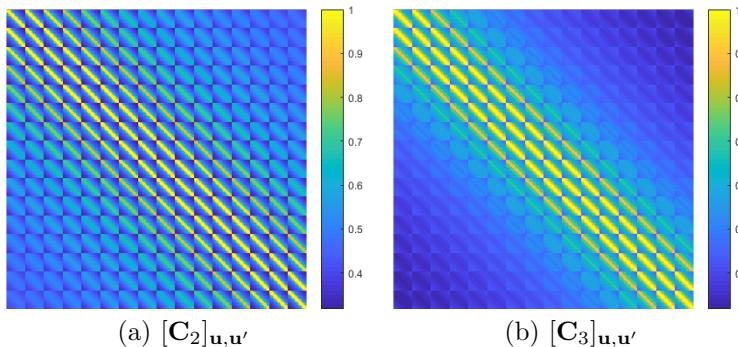


Figure 4.13: The covariance matrix \mathbf{C}_2 and \mathbf{C}_3 for an image of 16×16 pixels. The size of the matrices are $16^2 \times 16^2$.

Thus far, we have described how the Zernike coefficients $a_{\mathbf{u}, 2}$ and $a_{\mathbf{u}, 3}$ can be evaluated statistically. To finally draw the random tilt, we still need proper scaling so that the Zernike coefficient can correspond

4.4. THE ZERNIKE SPACE

to the number of pixels. As mentioned in [105, Eq. 2], the relationship between the tilt angle α_x and the Zernike coefficient is given by

$$\alpha_{\mathbf{u},2} = \frac{2\lambda}{\pi D} a_{\mathbf{u},2}, \quad \text{and} \quad \alpha_{\mathbf{u},3} = \frac{2\lambda}{\pi D} a_{\mathbf{u},3}. \quad (4.21)$$

Since the Nyquist sampling between two adjacent pixels in the object plane is $L\lambda/2D$, it holds that the tilt angles converted to the number of pixels in the object are

$$\alpha_{\mathbf{u},2} [\text{pixels}] = \frac{L \cdot \frac{2\lambda}{\pi D} a_2}{\frac{L\lambda}{2D}} = \frac{4}{\pi} a_{\mathbf{u},2}, \quad \text{and} \quad \alpha_{\mathbf{u},3} [\text{pixels}] = \frac{4}{\pi} a_{\mathbf{u},3}.$$

Putting everything together, we can show that the un-normalized correlation coefficient of the tilt angles is

$$\begin{aligned} \mathbb{E}[\alpha_{\mathbf{u},2} \alpha_{\mathbf{u}',2}] &= \frac{16}{\pi^2} \mathbb{E}[a_{\mathbf{u},2} a_{\mathbf{u}',2}] \\ &= \underbrace{\frac{16}{\pi^2} \frac{c_2}{2^{5/3}} I_0(0) [\mathbf{C}_2]_{\mathbf{u},\mathbf{u}'}}_{\kappa^2}, \end{aligned} \quad (4.22)$$

where we defined κ as the constant preceding the normalized correlation function $[\mathbf{C}_2]_{\mathbf{u},\mathbf{u}'}$. For vertical tilts, the relationship is $\mathbb{E}[\alpha_{\mathbf{u},3} \alpha_{\mathbf{u}',3}] = \kappa^2 [\mathbf{C}_3]_{\mathbf{u},\mathbf{u}'}$.

We are now in the position to draw random tilts. Since the correlation matrix represents a homogeneous process and so that matrix is block Toeplitz, drawing random samples can be done in the Fourier domain. The program below is a MATLAB implementation of drawing random horizontal tilts from the statistics.

```
kappa2 = I0(1)*7.7554*(D/r0)^(5/3)/(2^(5/3))*...
          (2*lambda/(pi*D))^(2*(2*D/lambda)^2;
Cx      = kappa2*C2;
Cxfft   = fft2(Cx);
S_half  = sqrt(abs(Cxfft));
b       = randn(N,N);
MVx = real(ifft2(S_half.*b))*sqrt(2)*N;
```

Two examples of the randomly sampled tilt vectors are shown in Figure 4.14. For the stronger turbulence condition $C_n^2 = 1 \times 10^{-15} \text{ m}^{-2/3}$, we can observe a strong correlation from one corner of the image to the other corner of the image. This explains the necessity of drawing the

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

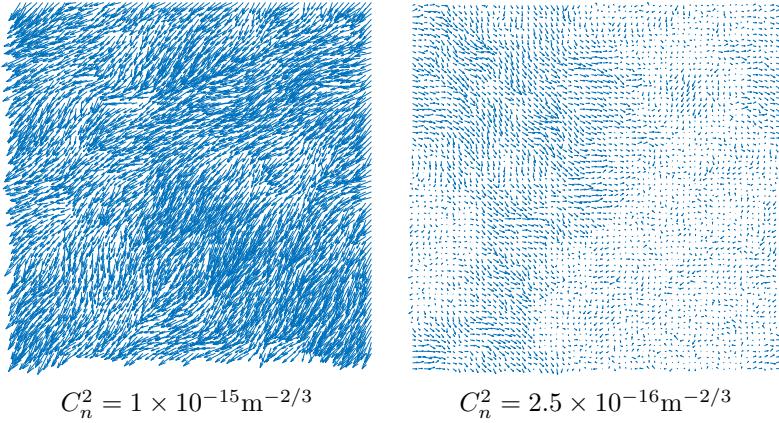


Figure 4.14: Simulated tilt maps for $C_n^2 = 1 \times 10^{-15} \text{m}^{-2/3}$, and $C_n^2 = 2.5 \times 10^{-16} \text{m}^{-2/3}$. The tilts are scaled by $2\times$ and skipped per every 8 pixels for display.

turbulence according to the theoretical model instead of a set of i.i.d. Gaussian random samples smoothed by heuristic operations.

A typical question for a numerical simulator is how do we know that the random samples we drew are correct? Since these random tilts are the particular realization of a two-dimensional random process, it is impossible to take a real image in the field and ask the simulator to match exactly. The best we can do is to argue from a statistical average point of view. If we take enough measurements, will the simulator produce something that would match the known theoretical statistics?

In the turbulence literature, the two commonly used statistics are the *Z-tilt* statistics and the differential-tilt statistics. The **Z-tilt** is defined as

$$\text{Z-tilt}(\mathbf{u}, \mathbf{u}') = \mathbb{E}[\alpha_{\mathbf{u},j} \alpha_{\mathbf{u}',j}], \quad j = 2, 3, \quad (4.23)$$

whereas the **differential tilt** is defined as

$$\text{D-tilt}(\mathbf{u}, \mathbf{u}') = \mathbb{E}[(\alpha_{\mathbf{u},j} - \alpha_{\mathbf{u}',j})^2], \quad j = 2, 3. \quad (4.24)$$

For the numerical simulation we just described, we can verify the validity of the simulator by comparing it against the theoretical statistics. In Figure 4.15 we show the comparison between the simulator (and its theoretical limit) and the one derived by Basu et al. [104]. Overall the

4.4. THE ZERNIKE SPACE

match between what the multi-aperture simulator can offer and what we expect it from the angle of arrival analysis in [104, 25]. Therefore, we can conclude that drawing the tilt without wave propagation has been successfully implemented.

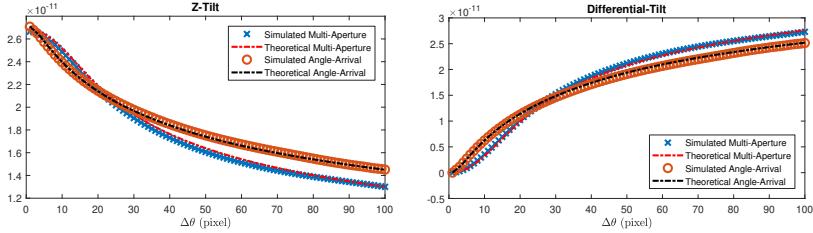


Figure 4.15: The theoretical and simulated tilt statistics at $C_n^2 = 1 \times 10^{-15} \text{ m}^{-2/3}$. The curves are plotted versus the number of Nyquist pixels. The angle-of-arrival is due to Basu et al. [104] as well as Fried [26], and the multi-aperture is our approximation based on Chanan [105]. Additionally, we note 20,000 random realizations were used in generating the simulated curves.

4.4.4 High-Order Correlations and a General Solution

The previous discussion focused on the tilt coefficients. This is for two reasons: (1) The tilt coefficients are easy to compare to existing results which are already due to the study of angle-of-arrival correlations; (2) the expression for tilt is a bit simpler to write down. To generalize the approximation, we want to include any pair $\mathbb{E}[a_{\mathbf{u},i} a_{\mathbf{u}',j}]$, leading us to present the following theorem which gives the correlations in a slightly more compact form:

Theorem 4.3 (Spatial correlation of all Zernike coefficients).

The spatial correlation of all Zernike coefficients $i, j \geq 1$ is approximated by

$$\mathbb{E}[a_{\mathbf{u},i} a_{\mathbf{u}',j}] \approx \mathcal{A} \sqrt{(n_i + 1)(n_j + 1)} \left(\frac{1}{2}\right)^{5/3} C_n^2 L f_{ij}(\mathbf{s}, k_0), \quad (4.25)$$

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

where $\mathcal{A} = 0.00969k^22^{14/3}\pi^{2/3}R^{5/3}$, n_i and n_j are the Noll indices, $\mathbf{s} = (\mathbf{u} - \mathbf{u}')/D$, and $f_{ij}(\mathbf{s}, k_0)$ is the integral defined in Takato and Yamaguchi [106].

This theorem represents the types of correlations utilized in the Zernike-based simulations [41, 42, 43], though we note that it is *still* an approximation.

Though we do not list f_{ij} directly, it is useful to observe a few properties of Equation (4.25). Firstly, the correlation depends on the difference $\mathbf{u} - \mathbf{u}'$, thus the field for a single coefficient is WSS. There is also an angular dependence, thus the correlation functions are anisotropic in nature. The aperture D also affects the length of correlation – a larger aperture will induce more correlations while a smaller aperture will create a more anisotropic effect. Finally, as expected, the propagation distance L and C_n^2 value influence the variance of the result, with a higher constant C_n^2 or longer propagation distance making the turbulence stronger.

To close our theoretical description of the correlations, the above problem of Zernike coefficient correlations has been solved without such approximation in [101] and is written in the following way:

$$\begin{aligned} \mathbb{E}[a_{\mathbf{u},i}a_{\mathbf{u}',j}] &= \mathcal{A}\sqrt{(n_i + 1)(n_j + 1)} \int_0^L \left(\frac{L-z}{L}\right)^{5/3} C_n^2(z) \\ &\quad \times f_{ij} \left(\left(\frac{z}{D(L-z)} \right) (\mathbf{u} - \mathbf{u}'), k_0 \right) dz. \end{aligned} \quad (4.26)$$

This represents a continuous integration over the path, which is a significant improvement over the approximation provided in [41]. It can be shown that Equation (4.25) is a special case of Equation (4.26) as described in Chimitt and Chan [101]. These results are similar to a paper by Whiteley et al. [107] thus we would suggest the interested reader to either paper for a discussion of the more general approach.

Fortunately, the correlation function one chooses to use (approximate or exact) doesn't affect *how* they are used in simulation.

4.4.5 Image Formation and the Zernike Space

The previous discussions highlight how we can describe correlations in the Zernike space, either through the approximation of (4.25) or (4.26). We now turn to answering a main question of this book: How can we understand image formation through the Zernike space?

4.5. ZERNIKE-BASED SIMULATION

To begin, we assume that phase distortions are sufficient in describing the effects. Recalling our familiar image formation equation,

$$I(\mathbf{x}) = |h_{\mathbf{u}}(\mathbf{x})|^2 \stackrel{\mathbf{u}}{\circledast} I_g(\mathbf{x}), \quad (4.27)$$

and PSF formation equation (utilizing the Zernike representation)

$$|h_{\mathbf{u}}(\mathbf{x})|^2 = \left| \mathfrak{Fourier} \left\{ P(\boldsymbol{\rho}) e^{j \sum_{j=1}^N a_{\mathbf{u},j} Z_j(\boldsymbol{\rho}/R)} \right\} \right|_{\mathbf{f}=\mathbf{x}/(\lambda z)}^2, \quad (4.28)$$

we can combine the two to write

$$I_i(\mathbf{x}) = \left| \mathfrak{Fourier} \left\{ P(\boldsymbol{\rho}) e^{j \sum_{j=1}^N a_{\mathbf{u},j} Z_j(\boldsymbol{\rho}/R)} \right\} \right|^2 \stackrel{\mathbf{u}}{\circledast} I_g(\mathbf{x}), \quad (4.29)$$

dropping the resizing terms. Equation (4.29) represents a compact form of image formation using the Zernike space. Thus, the entire process can be described with basis decomposition, random vectors, and spatially varying convolution.

While (4.29) is fairly compact in some sense, we can go a step further. That is, what if we were able to decompose the PSF into a set of basis functions, similar to what we did with the phase? If we were to write the PSF as

$$|h_{\mathbf{u}}(\mathbf{x})|^2 = \sum_{m=1}^M \beta_{\mathbf{u},m} \varphi_m(\mathbf{x}), \quad (4.30)$$

we could then approximate the spatially varying convolution as a sum of invariant convolutions. The reasons and methodology for doing so will be described shortly, but the result is the following:

$$I_i(\mathbf{u}) = \sum_{m=1}^M (\varphi_m \circledast (\beta_m \odot I_g))(\mathbf{u}), \quad (4.31)$$

where φ_m is our invariant basis representation of the PSF and β_m are the coefficients for the PSF basis (which, of course, are related to the Zernike space). While stating φ_m or the transform from the Zernike space to β_m will prove to be analytically intractable, we will discuss how it can be utilized in the context of reconstruction in Chapter 5.

4.5 Zernike-Based Simulation

Much like split-step simulation, it is a numerical interpretation of the true image formation process. In a similar fashion, the Zernike space is

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

a theoretical way of describing the turbulence distortions which we are presently interested in considering the numerical implementation of. Therefore, the following discussion is motivated by the computational difficulties one encounters when attempting to actualize this form of simulation.

4.5.1 Numerical Challenges

It may seem that we can use Cholesky-based generation to generate samples of the Zernike space as we did with a single coefficient vector. When the size of the Zernike spatial grid is small, its covariance is accordingly small and may therefore be generated using Cholesky decomposition. However, for a large grid of points, the size of the covariance matrix may result in the decomposition becoming infeasible. The size of the correlation matrix for an image of size $W \times H$ and 36 Zernike coefficients requires the construction of a matrix that is $36HW \times 36HW$. For a standard 256×256 image this results in a matrix that is over 2 million by 2 million entries in size! We would have to decompose and then multiply by this matrix for *one* realization, motivating us to look for an alternative.

One important case of numerical generation is that if the correlation structure is WSS, the matrix Σ is *circulant* and thus the eigen-decomposition is equivalent to the Fourier transform. In this case, generating the random vector \mathbf{y} can be implemented via

$$\mathbf{y} = \mathbf{U}\mathbf{S}^{\frac{1}{2}}\mathbf{U}^T\mathbf{e} = \mathcal{F}^{-1}(\mathbf{S}^{\frac{1}{2}}\mathcal{F}(\mathbf{e})),$$

where \mathcal{F} denotes the discrete-time Fourier transform, and the diagonal matrix \mathbf{S} is the Fourier spectrum of one row (or column) of the correlation matrix Σ . The significance of the homogeneity is that it allows us to speed up the sampling process by performing all computations in the Fourier space. In addition, the memory bottleneck is resolved because we don't need to construct the full correlation matrix Σ and run the Cholesky factorization. This is exactly the property we utilized for generating phase *screens* in Chapter 3.

This brings us to present two issues. For a single field, the correlation is WSS. Although Equation (4.25) is a function of the spatial separation $\mathbf{u} - \mathbf{u}'$ in the case of a single f_{ii} , the *entire* Zernike space is not WSS due to the fact that f_{ij} is not a function of index difference $i - j$. On top of this, we will still need to perform an FFT for converting the phase realizations into PSFs and spatially varying

convolution. Therefore, there are many numerical issues that have a non-trivial path forward to getting this simulation to where it needs to be for large dataset generation. We will tackle these one at a time, starting with this issue of a lack of WSS.

4.5.2 Fourier Sampling the Approximate Zernike Space

Although the Zernike space as a whole is non-WSS, there is an approximation that largely solves this issue [43]. This method utilizes the insight that for a single coefficient field, the correlation *is* WSS. This means that one may generate individual correlation fields correctly, however, they will be independent from one another. This alone will not solve our problem as this neglects the intermodal correlation.

To introduce the approximation, we will refer to the matrix which represents the correlational structure of the i th field with the j th field as $\mathbf{A}_{i,j}$. In other words, $\mathbf{A}_{i,j}$ corresponds to a matrix form of $f_{i,j}$. The entire Zernike space correlation, which is in the form of a **tensor**, can then be written as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \cdots & \mathbf{A}_{1,N} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{N,1} & \cdots & \mathbf{A}_{N,N} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathbf{s}_0} & \cdots & \mathbf{A}_{\mathbf{s}_p} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{\mathbf{s}_p} & \cdots & \mathbf{A}_{\mathbf{s}_0} \end{bmatrix},$$

where here we are showing two “projections” of the same tensor. To motivate our development, we pose the following question: Can we generate the fields independently, utilizing the single coefficient WSS property, then mix them according to *some* matrix? This question eventually led to the following scheme for sampling the Zernike space in an approximate manner:

1. Generate $i = \{1, 2, \dots, N\}$ unit-variance, spatially correlated random fields according to $\mathbf{A}_{i,i}$. Note this generation uses only autocovariance functions. At this stage, our N fields are independent, thus utilizing the FFT-based method based on their homogeneity;
2. Perform a point-wise mixing of the random fields according to the Noll matrix Σ . This mixing is done pixel-wise (across the coefficient index dimension) per pixel.

This generation process can be written mathematically in the following

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

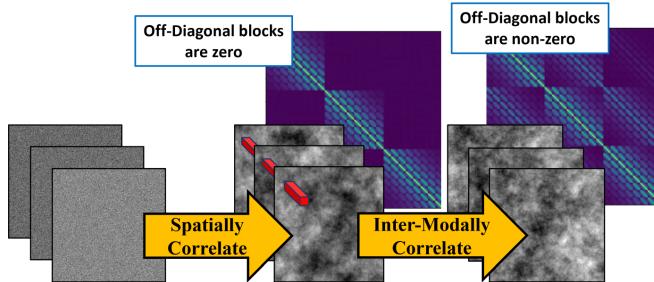


Figure 4.16: A visual representation of the realizations and covariance structure as it changes through the generation process. White noise is used to generate independent random fields which are then mixed according to the index axis covariance. Here, the covariance structure is of a smaller grid than the random field shown for ease of interpretation. Source: [43].

way

$$\tilde{\mathbf{A}} = \mathbf{L} \begin{bmatrix} \mathbf{A}_{1,1} & 0 & \dots & 0 \\ 0 & \mathbf{A}_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{A}_{N,N} \end{bmatrix} \mathbf{L}^T, \quad (4.32)$$

where $\mathbf{L}\mathbf{L}^T = \Sigma$ and $\tilde{\mathbf{A}}$ is regarded as the approximation to the Zernike space.

We note the resultant matrix is no longer diagonal, however, the off-diagonal entries will be approximations of the true correlation structure. This approximation and its impact have been analyzed in [43]. A visualization of the proposed generation process is provided in Figure 4.16 using a simplistic example in the case of an 8×8 image with 3 coefficient fields for visualizing the covariance structure. The same concept presented in this Figure can be extended to the case of an $W \times H$ image with N coefficient fields.

4.5.3 Approximating Spatially Varying Blur

Invariant convolution may be implemented through the FFT and is accordingly fast to compute. However, spatially variant convolution does *not* have such a simple mathematical relationship, and thus is accordingly far slower than its invariant counterpart. Therefore, we are

4.5. ZERNIKE-BASED SIMULATION

interested in representing spatially variant convolution and a summation of *invariant* convolutions. To this end, we follow the approach outlined in Mao et al. [42].

To begin, we recall that the PSF $|h_{\mathbf{u}}(\mathbf{x})|^2$ is defined per pixel. This is analogous to the phase, which is also defined per pixel. For the phase, we decomposed it via the Zernike polynomials, for the PSF there is no such easy decomposition. However, if it *were* true, we could write the following **PSF basis representation** via basis functions $\varphi_1(\mathbf{x}), \dots, \varphi_M(\mathbf{x})$ as

$$|h_{\mathbf{u}}(\mathbf{x})|^2 = \sum_{m=1}^M \beta_{\mathbf{u},m} \varphi_m(\mathbf{x}), \quad (4.33)$$

where $\beta_{\mathbf{u},m}$ is the m th basis coefficient at pixel location \mathbf{u} . Here, $\varphi_m(\mathbf{x})$ is the basis function that will be shared across *all* the pixel locations. The coordinate \mathbf{u} is embedded in the coefficient $\beta_{\mathbf{u},m}$ which is \mathbf{u} -dependent.

Since we are interested in spatially varying convolution, we are inevitably led back to our discussion of scattering versus gathering. As described in Chimitt et al. [15], gathering results in the following form:

$$I_i(\mathbf{x}) = \underbrace{\sum_{m=1}^M \beta_{\mathbf{x},m} \underbrace{(\varphi_m \circledast I_g)(\mathbf{u})}_{\text{invariant blur}}}_{\text{linear combination of invariant blurs}}, \quad (4.34)$$

while scattering results in

$$\underbrace{I_i(\mathbf{x}) = \sum_{m=1}^M \left(\varphi_m \circledast \underbrace{(\beta_m \odot I_g)}_{\text{pixelwise weighted input}} \right) (\mathbf{x})}_{\text{linear combination of invariant blurs}}. \quad (4.35)$$

The difference between these two is subtle, coming down to the order of operations that are not commutable. That is, do we multiply by the coefficients first, or do we convolve first? The answer to this question is given in [15] where scattering was found to be the proper choice for simulation.

Through (4.35), the convolutions within the summation are spatially invariant. Thus, we can multiply the image $I_g(\mathbf{x})$ by each random field β_m and convolve with the associated basis functions φ_m . After

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

this, summation gives us a spatially varying image, which we depict in Figure 4.17. The result is a reduction of the computation, now we only require M 2D FFTs where in Mao et al. [42] M was chosen to be 100.

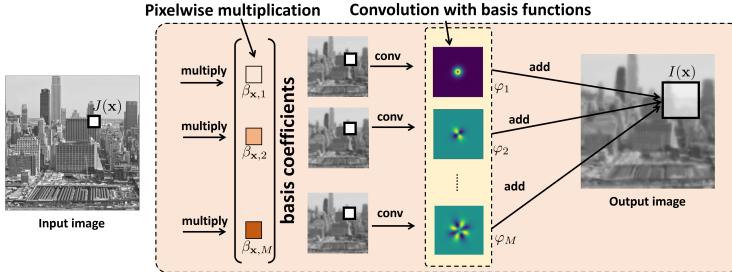


Figure 4.17: A block diagram that depicts how a pixel is multiplied by its coefficient vector and convolved with its corresponding basis functions. Their contribution is then accordingly distributed across the output image. Note that the “receptive field” is larger in the output image due to the interpretation of scattering.

This argument rests upon the assumption that (4.33) is possible. This leaves us with two fundamental questions in order to facilitate this approach: (1) How do we construct the PSF basis functions? (2) How do we determine the basis coefficients from the Zernike space?

For the PSF basis functions, we can construct them *offline* via a **supervised learning** approach as shown in Figure 4.18. Suppose that we have run the simulation offline to generate a large set of PSFs $|h_{\mathbf{u}_1}(\mathbf{x})|^2, \dots, |h_{\mathbf{u}_P}(\mathbf{x})|^2$ where P is a very large number. These PSFs can be drawn independently because the goal here is to learn the common structure of all of them. To ensure sufficient variability, when generating these PSFs, we can use a wide range of C_n^2 values so that the PSFs cover the turbulence conditions from weak to strong.

Given the simulated PSFs $|h_{\mathbf{u}_1}(\mathbf{x})|^2, \dots, |h_{\mathbf{u}_P}(\mathbf{x})|^2$, we can perform a standard **principal component analysis** (PCA) to determine the basis functions $\{\varphi_m(\mathbf{x}) \mid m = 1, \dots, M\}$. Implementations of the PCA are available in most of the modern computing libraries and so we skip it for brevity. However, one thing to note is that the PSFs are the results of both the *tilts* and the *high-order aberrations*. For tilts, using the PCA to encode the tilt is not cost-effective because the tilts are simple delta functions in the two-dimensional space. All they do is to point a pixel to the new location. For such a simple operation, we can just record them directly without wasting the principal components.

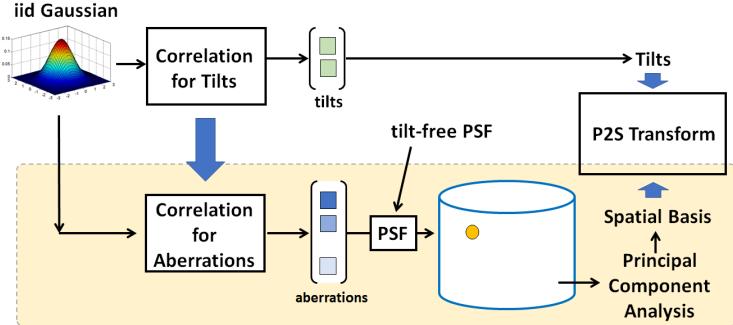


Figure 4.18: To generate the basis functions $\{\varphi_m(\mathbf{x}) \mid m = 1, \dots, M\}$, we first simulate a large set of PSFs $h_{\mathbf{u}_1}(\mathbf{x}), \dots, h_{\mathbf{u}_P}(\mathbf{x})$. By running the principal component analysis, we can determine the basis functions. Source: [42]

Decoupling the tilts from the high-order aberrations also decouples the training data from the tilts. This will then reduce the number of training samples for the PCA. The idea is that for the tilts, we save them and record their statistics. For the high-order aberrations, we send the training samples to the PCA to learn the PSF structure.

4.5.4 The Phase-to-Space Transform

With the basis functions $\varphi_m(\mathbf{x})$ able to be found computationally, we now discuss the basis coefficients $\{\beta_{\mathbf{u},m} \mid m = 1, \dots, M\}$. The definition of the basis coefficients is that

$$\beta_{\mathbf{u},m} = \langle |h_{\mathbf{u}}(\mathbf{x})|^2, \varphi_m(\mathbf{x}) \rangle = \int |h_{\mathbf{u}}(\mathbf{x})|^2 \varphi_m(\mathbf{x}) d\mathbf{x}. \quad (4.36)$$

However, in the absence of the PSF $|h_{\mathbf{u}}(\mathbf{x})|^2$, we need a mechanism to generate the coefficients $\beta_{\mathbf{u},m}$. If we knew the $|h_{\mathbf{u}}(\mathbf{x})|^2$ then there is no problem to solve because the goal here is to generate the PSF!

The correspondence between the Zernike coefficients and a set of PSF basis coefficients has been considered in the form of the extended Nijboer-Zernike diffraction theory [108, 109]. This theory is rather elegant, however, our PCA basis functions φ_m will be of no use if we adopt it. Furthermore, it is unclear whether or not the basis is optimal for turbulence. Therefore, in [42], Mao et al. produced a learning-based approach to provide the mapping from $\{a_{\mathbf{u},n} \mid n = 1, \dots, N\}$ to

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

$\{\beta_{u,m} | m = 1, \dots, M\}$. This network is called the **Phase-to-Space (P2S) Transform**. The name Phase-to-Space means that we are converting the phase coefficients to spatial PSF coefficients.

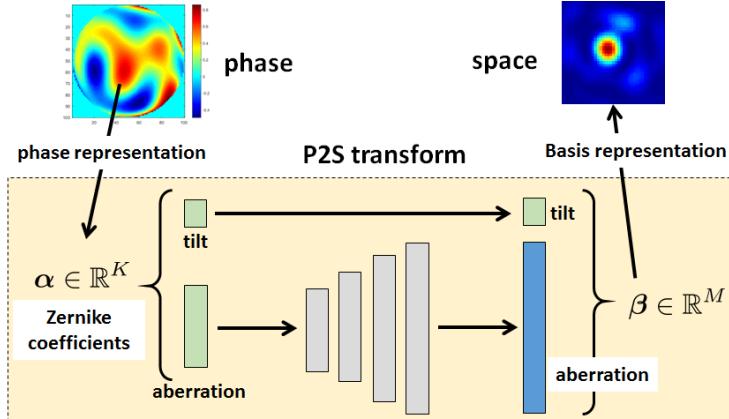


Figure 4.19: A visualization of the Phase-to-Space (P2S) Transform network. A shallow neural network maps the higher-order Zernike coefficients to a set of PSF coefficients. Source: [42].

The P2S framework is shown in Figure 4.19. The idea is train a *shallow* neural network that maps the vector $\mathbf{a} = [a_{u,1}, \dots, a_{u,N}]^T$ to another vector $\beta = [\beta_{u,1}, \dots, \beta_{u,M}]^T$. The dimension of the vector \mathbf{a} is typically in the order $N = 36$ whereas the dimension of the vector β is typically around $M = 100$. Since the dimension is so small, a simple fully-connected neural network would suffice for learning the mapping.

How can we train such a transform? If we had a large set of training data consisting of pairs of Zernike coefficient vectors and PSF basis coefficient vectors, we could simply train a network to find the mapping. Here we drop the subscript of position as we are interested in realizations that are completely independent. Each training sample will consist of the pair of vectors (\mathbf{a}, β) , with the Zernike basis coefficient vector \mathbf{a} as the input and the PSF basis coefficient vector β as the output. To prepare the vectors β , we first construct the PSF $|h_u(\mathbf{x})|^2$ associated with the vector \mathbf{a} . Then, we use Equation (4.36) to compute β . The entire process can be done offline. When the training samples are prepared, we train the shallow neural network until the convergence criteria are met. The resulting neural network will then be able to perform the mapping $\mathbf{a} \rightarrow \beta$. Concerning the exact network

4.5. ZERNIKE-BASED SIMULATION

architecture, Table 4.1 shows the network reported in [42].

input layer	layer 1	layer 2	output layer
	full conn.	full conn.	full conn.
34	34	100	100

Table 4.1: Network architecture of the P2S transform network.

The advantage of using a neural network for the Phase-to-Space transform is that it can be parallelized. For a high-resolution image with many pixels, the parallelism enables us to generate the β 's simultaneously across all pixels in the image in *a single pass*. Such a speed up can be significant for large images.

4.5.5 Summary

At this point, we have described various aspects of Zernike-based simulation, now we wish to put it all together. At a high level, Zernike-based simulation follows the pipeline of generating Zernike space samples and applying them to an image via the P2S transform (also utilizing (4.35)). A depiction of Zernike-based simulation is presented in Figure 4.20. Let us spend some effort in writing the recipe down to accompany this depiction which should also help to unify the topics we've discussed:

1. **Zernike space sampling.** Using the approximation of Chimitt et al. [43], we can generate realizations of the Zernike space quickly. To do so, we must have precomputed either of the correlation kernels (4.25) or (4.26). It is possible to compute these correlation kernels once and sample them as required – thus the integration need only be performed computationally one time. Thus, the correlation kernels shown are sampled from the pre-computed integrals and used to generate the realization (through FFT-based random signal generation) of the Zernike space \mathbf{a}_u .
2. **Image warping.** With a random realization of the Zernike space, we can first distort the input image via warping. To do this, we take the tilt Zernike coefficients and send them to a warping module which distorts the image accordingly. It is important to note that one must convert from the Zernike coefficient values to pixels (which requires knowledge of the sample spacing).

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

3. **P2S transform.** The P2S network then receives the higher order Zernike coefficients and converts them to β_u . This represents the β_m s in (4.35) that we will multiply the warped image by. Note that this step saves us a great deal of time – we don’t have to take an FFT to form the PSFs *and* we avoid a direct implementation of spatially variant convolution, instead opting for an approximation.
4. **Spatially variant convolution.** Finally, we wish to form the output image by the usage of the approximation to spatially variant convolution. The warped image is multiplied by each “plane” of coefficients β_u and is convolved by its corresponding basis function φ_m which we refer to as the P2S kernels. From here, we can sum and normalize them, forming the final output image.

We compare the results by Zernike-based simulation with split-step simulation along with simulation times in Figure 4.21. Although the results seem somewhat similar, the Zernike-based simulation finishes in under 1 second whereas a full dense-field split-step simulation (i.e. one complete split-step propagation per pixel) would take approximately 20 minutes.

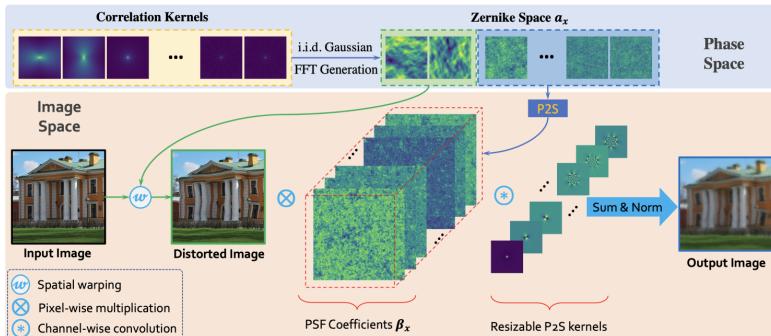


Figure 4.20: An overview of Zernike-based simulation as described in the series of papers [41, 42, 43].

To highlight the gain in speed offered by Zernike-based simulation, this process can be done for a 4k image in 1 minute [43] with a basis vector for *every* pixel. While this may seem slow to some (one minute for a single image!) using split-step with 10 phase screens in the same fashion for a 4k image (that is, using a propagation for every pixel in the image) would take approximately 13 hours! We show a 4k image

4.5. ZERNIKE-BASED SIMULATION

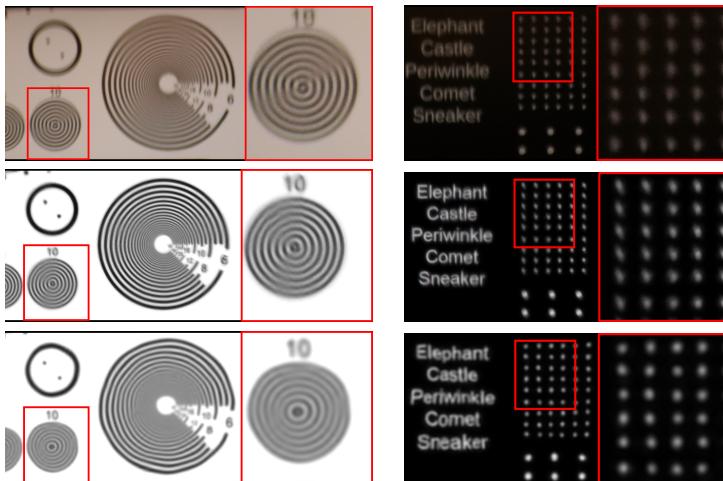


Figure 4.21: A comparison of [Top] Ground truth data [Middle] Zernike-based simulation and [Bottom] Split-step simulation. Source: [43].

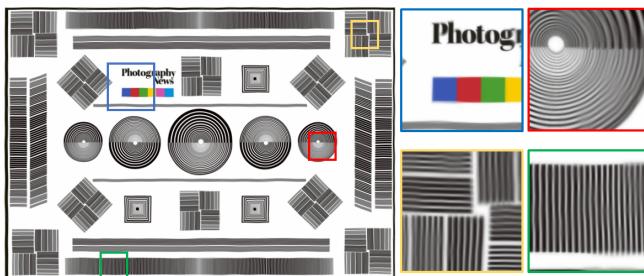


Figure 4.22: A 4k image simulated using the Zernike-based approach (here at a slightly lower resolution to save on PDF rendering time). Source: [43].

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

simulated in this fashion in Figure 4.22 (at a lower resolution here to save on PDF rendering time).

To mirror the presentation of split-step, we also provide a pseudo-code version of Zernike-based simulation as follows:

Algorithm 2 Zernike-based simulation

- 1: Determine the size of each Zernike field (equivalent to the size of an image in pixels)
 - 2: Generate M independent Zernike fields
 - 3: Mix along index-dimension according to the Noll matrix
 - 4: Apply warping transformation to input image I_g
 - 5: Perform P2S transform
 - 6: Multiple I_g by the various basis coefficient fields, producing $\beta_m I_g$ for $m \in 1, \dots, M$.
 - 7: Mix according to scattering convolution approximation
-

We note that the spatially varying convolution approximation step may be utilized in split-step, however, the phase realization $\phi_{\mathbf{u}}$ would need to be decomposed by the Zernike polynomials, thus requiring an additional step of inner products. However, there would still likely be a gain in speed due to the vast advantage of Fourier-based convolution. With the main recipe of this variety of simulation presented, we turn to a few additional considerations we wish to highlight.

4.6 Additional Topics

To not distract us from the critical details, we have avoided a few additional considerations. To close the loop on these, we present them now to be applied in hindsight. These do not dramatically impact the understanding of the model or simulation, however, they are important considerations when doing simulation. We additionally present a real-time application of this variety of simulation.

4.6.1 Wavelength

In principle, the spectral response of the turbulent medium is wavelength dependent, and the distortion must be simulated for a dense set of wavelengths. However, as we discussed in the beginning of Chapter 3, the index of refraction can be decoupled into the static part $n_0(\mathbf{r}, \lambda)$

and the varying part $n_1(\mathbf{r}, t)$. The static part $n_0(\mathbf{r}, \lambda)$ has some dependencies on the wavelength but it does not contribute to the turbulence distortion. It is mostly the refractive index in the ambient medium. The varying part $n_1(\mathbf{r}, t)$ changes with respect to the location \mathbf{r} and time t . For the typical visible spectrum (roughly 400nm to 700nm), the influence due to the wavelength is insignificant compared to the time and location. Therefore, it is sometimes justifiable to use one wavelength for all three color channels (RGB) when constructing the PSFs.

To illustrate the situation, Mao et al. [42] showed an example (re-shown in Figure 4.23) of the individual PSFs for several wavelength from 400nm (blue) to 700nm (red). By observing the figure, we notice that the shape of the PSFs barely changes from one wavelength to another. In the same figure, we simulate two color images. The first image is simulated by using a single PSF (525nm) for the color channels (and displayed as an RGB image). The second image is simulated by considering 3 PSFs with wavelengths 450nm, 540nm, and 670nm. We note that (c) is a more realistic simulation but requires $3 \times$ computation. However, the similar PSFs across the color make the difference visually indistinguishable, as seen in (d).

The small gap demonstrated in Figure 4.23 suggests that we can simulate the RGB channels identically. In Mao et al. [42], the authors acknowledge that this approximation does not hold in the worst-case scenarios when the optical system demonstrates substantial chromatic aberrations. This in itself is not a *proof* that regardless of the situation one is simulating, one is justified to simulate one wavelength and apply it to an image as if it were valid for all wavelengths. We do not intend the previous discussion as a suggestion that this applies in all cases *or* that it is even a proof for the particular case considered. It is always important to consider whether or not this approximation is valid for the situation one is modeling. We refer the reader to Hardie et al. [80] for a detailed discussion of multi-spectral simulation, which addresses this problem in the context of split-step simulation.

4.6.2 Elements of Realistic Simulation

With publicly available code for the simulator by Mao et al. [42], it seems natural to download it, vary the parameters, and receive a large amount of perfectly accurate training data. We emphasize that, unfortunately, this is *not* the case.

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

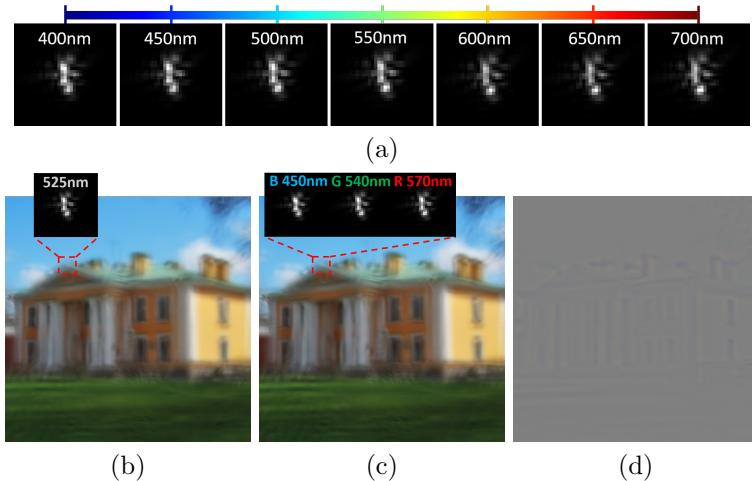


Figure 4.23: (a) PSFs across the visible spectrum. (b) The same distortion applied to three channels using the center wavelength of the visible spectrum. (c) Wavelength-dependent distortions applied to three channels. (d) Difference map between (b) and (d). Source: [42].

The careful application of an imaging simulation is not as simple as it may appear. Consider an image I_g passed to the simulator. Firstly, it must be noted that *no* image, in reality, is truly equal to I_g . A perfectly in-focus image, however, may serve as a suitable approximation. A more significant difficulty arises when we consider the sampling of I_g , which is potentially unknown. Our image formation equation depends on knowing the spatial parameters so that the PSF may be properly applied.

To elaborate further, let us consider an image that is chosen out of a dataset. Applying the simulator naively will almost certainly result in a mismatch between the generated PSFs and the image. In other words, our $|h_u|^2$ will be either too big or too small. How do we account for this? We must know or assume the sampling of the image to be simulated as it will impact the perceived strength of turbulence, correlation distance, pixel shifting, and so on. It cannot be ignored.

As a closing note, we find it to be permissible to perform this “naive” application of a simulator to a dataset if the parameters are chosen somewhat reasonably. This will allow one to train a network for some downstream task which, in our experience, can be used on

real data reasonably well. However, to say that any image out of the dataset is representative of turbulence for the chosen parameters is simply wrong *unless* the user took the care of ensuring the sampling measurements were correct, justifying any assumption of wavelength-invariance, and so on. As a rule of thumb, naive application to a dataset is acceptable if a downstream task stands to benefit from it; it is not acceptable in the case of any analysis of the nature of turbulence.

4.6.3 Real-Time Turbulence Generation

We conclude the Chapter with a graphical user interface (GUI) for real-time turbulence simulation. Figure 4.24 shows a GUI built by Purdue University. In this GUI, we can feed the simulator with a camera source such as a webcam. The resolution of the camera feed for this particular GUI is limited to 512×512 . The simulation is run on a host machine with an nVidia RTX 3080Ti GPU. The throughput of the simulator is approximately 7 frames per second.

As we can see in the demonstration, the GUI will generate in real time the turbulence-distorted image. Simultaneously, it displays the real-time point spread functions as well as the real-time tilt map. The subfigures on the bottom left are the real-time tilt statistics, including the Z-tilt and the D-Tilt. Of course, they don't match as they are individual realizations, on expectation they do indeed match!

4.7 Summary

This Chapter has described a more computer vision and computational imaging view of the simulation and modeling of the phase distortions caused by atmospheric turbulence. The ideas discussed frame the image formation process as a sampling problem, with the Zernike basis coefficients forming a field from which the image is formed.

Part 1 Split-step alternatives: We began by introducing the problems which arise from split-step when we wish to use it for generating training data. This led us to survey the existing alternatives to split-step, ranging from ray tracing to GANs. This brought us to introduce phase-based simulation to which the described Zernike-based simulation modality belongs.

Part 2 The phase as a basis: The phase-over-aperture model and Zernike representation introduce an alternative way of viewing the process of modeling the turbulent image formation process. Specifically,

CHAPTER 4. PROPAGATION-FREE MODELING AND SIMULATION

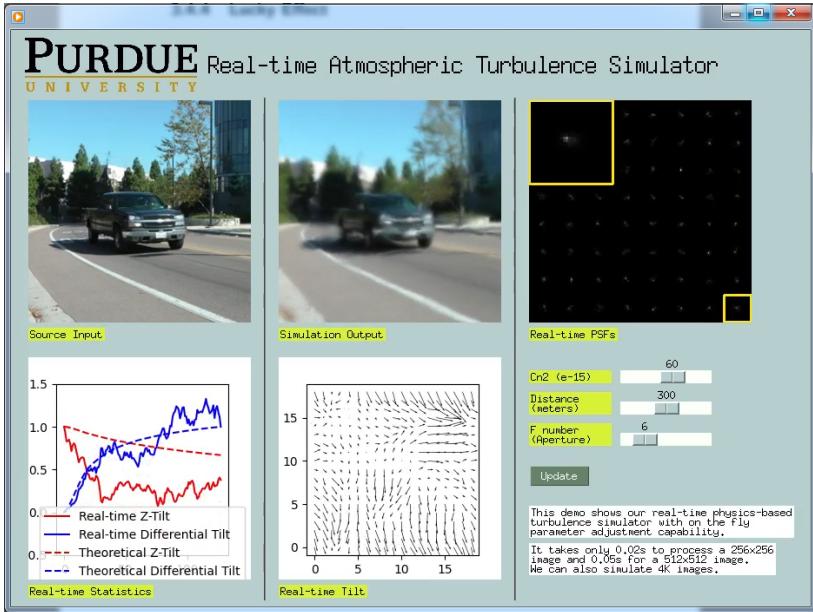


Figure 4.24: The GUI for turbulence simulation developed by Purdue University, with tunable knobs for turbulence parameters and real-time turbulent realization graphics.

the distortions may be viewed as arising from a basis coefficients field, which we refer to as the Zernike space $\{\mathbf{a}_u\}$.

Part 3 The Zernike space: The concept of the Zernike space and its application to simulation was introduced by the development of the multi-aperture simulation. Utilizing classical results from astronomical telescope correlations, the multi-aperture simulation extends these concepts to simulating spatially correlated ground-to-ground images.

Part 4 Zernike-based simulation: The P2S approach [42] combines the theoretical insight of the multi-aperture simulation with a forward transformation which is a neural network-based approach. This allows for the multi-aperture simulation to be applied to a broader class of problems, such as generating data for training machine learning restoration methods.

5. Image Restoration

In long-range imaging, one of the biggest challenges is to recover distorted images. There are hardware and software solutions, but they each have pros and cons. Hardware solutions are often faster, but they are constrained by size, weight, and power. Software solutions are not as customized although they have a larger degree of freedom. In this Chapter, we will focus on the software (algorithm) solutions.

Developing image restoration algorithms for atmospheric turbulence is not an easy task. For the most part, the challenges are associated with the uncertainty and complexity of the forward model which is random. This is different from classical inverse problems such as **deconvolution** where the blur is unknown but structured and deterministic.

Notations

Discussing the inverse problems requires some changes in notations. Since all inverse problems are solved on a computer, any continuous function has to be discretized. On the object plane, the continuous function $J(\mathbf{x})$ is now represented by a N -dimensional vector

$$(\text{object plane}) \quad \mathbf{J} = [J(\mathbf{x}_1), J(\mathbf{x}_2), \dots, J(\mathbf{x}_N)]^T \in \mathbb{R}^N,$$

where $\mathbf{x}_1, \dots, \mathbf{x}_N$ are the N coordinates the digital image is defined upon. Similarly, on the image plane, the function $I(\mathbf{x})$ is now represented as

$$(\text{image plane}) \quad \mathbf{I} = [I(\mathbf{x}_1), I(\mathbf{x}_2), \dots, I(\mathbf{x}_N)]^T \in \mathbb{R}^N.$$

The discretized PSFs $h_{\mathbf{u}}(\mathbf{x})$ will be written as

$$(\text{image plane}) \quad \mathbf{h}_{\mathbf{u}} = [h_{\mathbf{u}}(\mathbf{x}_1), h_{\mathbf{u}}(\mathbf{x}_2), \dots, h_{\mathbf{u}}(\mathbf{x}_N)]^T \in \mathbb{R}^N,$$

CHAPTER 5. IMAGE RESTORATION

where we note for simplicity in notation we will opt to use $h_{\mathbf{u}}(\mathbf{x}_N)$ and $\mathbf{h}_{\mathbf{u}}$ for the PSFs in this Chapter.

The relationship between the clean image and the distorted image is given by a nonlinear mapping $\mathcal{H}_{\boldsymbol{\theta}} : \mathbb{R}^N \rightarrow \mathbb{R}^N$:

$$\mathbf{I} = \mathcal{H}_{\boldsymbol{\theta}}(\mathbf{J}), \quad (5.1)$$

where $\boldsymbol{\theta}$ denotes some underlying hyper-parameters. Putting this in the context of atmospheric turbulence, $\mathcal{H}_{\boldsymbol{\theta}}$ is the wave propagation equations and $\boldsymbol{\theta}$ represents the turbulence parameters. Or, if we use the Zernike space model, then $\boldsymbol{\theta}$ would be the Zernike coefficients defined by the atmospheric turbulence. In the simplest case where $\mathcal{H}_{\boldsymbol{\theta}}$ represents an isoplanatic turbulence so that the blur is spatially invariant, the equation can be simplified to $\mathbf{I} = \mathbf{h} \circledast \mathbf{J}$ using a blur kernel \mathbf{h} and the convolution operator \circledast . If this happens, it resembles the classical deconvolution problem.

In non-local image processing algorithms, we sometimes need to process *patches* instead of the whole image. An image patch is a small 2D neighborhood around a certain pixel. Suppose we are looking at coordinate \mathbf{x}_i , we define a p -dimensional patch as

$$(\text{patch at pixel } \mathbf{x}_i) \quad \underline{\mathbf{I}}(\mathbf{x}_i) = [I(\mathbf{x}_i + \Delta\mathbf{x}_1), \dots, I(\mathbf{x}_i + \Delta\mathbf{x}_p)]^T,$$

where $\Delta\mathbf{x}_1, \dots, \Delta\mathbf{x}_p$ define a neighborhood of p pixels surrounding \mathbf{x}_i . The neighborhood is typically square.

If we need to extend the definitions to videos instead of a single image, we can associate the time axis to either the image or a patch. Using the image as an illustration, we say that we are looking at a frame located at time t as

$$(\text{image at time } t) \quad \mathbf{I}(t) = [I(\mathbf{x}_1, t), I(\mathbf{x}_2, t), \dots, I(\mathbf{x}_N, t)]^T.$$

To specify a patch located at pixel \mathbf{x} and time t , we write $\underline{\mathbf{I}}(\mathbf{x}, t)$.

Inverse Problems

As we shall elaborate in detail later in this Chapter, the inverse problem associated with atmospheric turbulence is often formulated as

$$\hat{\mathbf{J}} = \underset{\mathbf{J}}{\operatorname{argmin}} \|\mathbf{I} - \mathcal{H}_{\boldsymbol{\theta}}(\mathbf{J})\|^2 + \lambda g(\mathbf{J}), \quad (5.2)$$

where \mathbf{I} is the observed distorted image, \mathbf{J} is the latent clean image, $\mathcal{H}_{\boldsymbol{\theta}}$ is the turbulence forward degradation model, and $g(\mathbf{J})$ is a **regularization function** constraining the search space of the solution. This

5.1. UNDERSTANDING THE FORWARD MODEL

least-squares fitting problem is *not* necessarily the only way to formulate the problem. In fact, many deep learning algorithms would directly aim to recover \mathbf{J} without even worrying about if we are using the least-square function or which regularization function g to use. The purpose of introducing Equation (5.2) is to help readers who are familiar with the classical optimization-based inverse problems (such as total variation) to appreciate the connection. Even for readers coming from a deep learning perspective, we often find Equation (5.2) useful in terms of elaborating insights.

The difficulty of Equation (5.2) is when \mathcal{H}_θ is complicated, e.g., \mathcal{H}_θ is the turbulence model of which the instantaneous random realization is not known. For this problem, we need to simultaneously estimate \mathcal{H}_θ while recovering \mathbf{J} . This is a strongly ill-posed optimization because the number of parameters needed to describe \mathcal{H}_θ can be even larger than the number of pixels in \mathbf{J} . Thus, unless the structures of the image content and turbulence statistics are properly utilized, recovering \mathbf{J} from the above optimization can be extremely challenging if not impossible.

Plan for this Chapter

Our goal in this Chapter is to highlight the key principles behind atmospheric turbulence restoration algorithms. We acknowledge the exponential growth of papers in the field over the past three years thanks to the promising results of deep learning. However, given the sheer volume of the algorithms, instead of listing every single method, we believe it is more meaningful to focus on a few main concepts.

5.1 Understanding the Forward Model

Since the core difficulty in solving the inverse problem is the complexity of the forward model, in this Chapter, we take a closer look at the operator that defines the turbulence forward model. For simplicity of the discussion, we shall focus on the spatial domain.

5.1.1 Turbulence Operator \mathcal{H}_θ

The distortion caused by turbulence is spatially varying. Therefore, any convolution that defines the blur needs to be spatially varying too. Let's start by considering the spatially varying convolution. If the

CHAPTER 5. IMAGE RESTORATION

ground truth clean image intensity (at an input pixel coordinate \mathbf{u}) is $J(\mathbf{u})$ and the spatially varying point spread function (PSF) is defined by $h_{\mathbf{u}}(\mathbf{x})$ (where \mathbf{u} specifies the source location), the observed image intensity (at an output coordinate \mathbf{x}), $I(\mathbf{x})$, will be defined according to

$$I(\mathbf{x}) = \int h_{\mathbf{u}}(\mathbf{x} - \mathbf{u}) J(\mathbf{u}) d\mathbf{u} \quad (5.3)$$

where \circledast denotes convolution.

Although the model in Equation (5.3) is spatially varying, it is linear due to the linearity of convolution. This can be shown by noting that if there are two clean images $J_1(\mathbf{x})$ and $J_2(\mathbf{x})$ (correspondingly two distorted images $I_1(\mathbf{x})$ and $I_2(\mathbf{x})$), then for any constants a and b , it holds that

$$\begin{aligned} I(\mathbf{x}) &= \int h_{\mathbf{u}}(\mathbf{x} - \mathbf{u}) (aJ_1(\mathbf{u}) + bJ_2(\mathbf{u})) d\mathbf{u} \\ &= a \underbrace{\int h_{\mathbf{u}}(\mathbf{x} - \mathbf{u}) J_1(\mathbf{u}) d\mathbf{u}}_{I_1(\mathbf{x})} + b \underbrace{\int h_{\mathbf{u}}(\mathbf{x} - \mathbf{u}) J_2(\mathbf{u}) d\mathbf{u}}_{I_2(\mathbf{x})}. \end{aligned}$$

Therefore, we have shown the linearity of the operator.

In matrix and vector form, the convolution equation for each of the coordinates $\mathbf{x}_1, \dots, \mathbf{x}_N$ can be expressed in terms of a linear combination of the input pixels

$$I(\mathbf{x}_i) = \sum_{j=1}^N h_{\mathbf{u}_j}(\mathbf{x}_i - \mathbf{u}_j) J(\mathbf{u}_j), \quad i = 1, \dots, N, \quad (5.4)$$

where $h_{\mathbf{u}_j}(\mathbf{x}_i)$ denotes the \mathbf{x}_i th pixel of the point spread function from pixel \mathbf{u}_j , and $I(\mathbf{x}_i)$ and $J(\mathbf{u}_j)$ are the pixel values of the observed and the clean images, respectively. This leads us to write

$$\mathbf{I} = \mathcal{H}_{\boldsymbol{\theta}}(\mathbf{J}), \quad (5.5)$$

where the operator $\mathcal{H}_{\boldsymbol{\theta}}$ captures the convolutions imparted by the spatially varying point spread functions $h_{\mathbf{u}_1}, \dots, h_{\mathbf{u}_N}$. If we consider all

5.1. UNDERSTANDING THE FORWARD MODEL

the N pixels in Equation (5.4), \mathcal{H}_{θ} takes on a familiar form:

$$\mathbf{I} = \underbrace{\begin{bmatrix} h_{\mathbf{u}_1}(\mathbf{x}_1 - \mathbf{u}_1) & h_{\mathbf{u}_2}(\mathbf{x}_1 - \mathbf{u}_2) & \dots & h_{\mathbf{u}_N}(\mathbf{x}_1 - \mathbf{u}_N) \\ h_{\mathbf{u}_1}(\mathbf{x}_2 - \mathbf{u}_1) & h_{\mathbf{u}_2}(\mathbf{x}_2 - \mathbf{u}_2) & \dots & h_{\mathbf{u}_N}(\mathbf{x}_2 - \mathbf{u}_N) \\ \vdots & \vdots & \ddots & \vdots \\ h_{\mathbf{u}_1}(\mathbf{x}_N - \mathbf{u}_1) & h_{\mathbf{u}_2}(\mathbf{x}_N - \mathbf{u}_2) & \dots & h_{\mathbf{u}_N}(\mathbf{x}_N - \mathbf{u}_N) \end{bmatrix}}_{\mathcal{H}_{\theta}} \mathbf{J}. \quad (5.6)$$

A pictorial illustration is shown in Figure 5.1.

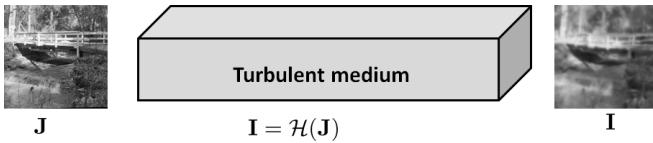


Figure 5.1: The forward model of the atmospheric turbulence can be written as a general operator \mathcal{H}_{θ} that takes a clean image \mathbf{J} and maps it to a distorted image \mathbf{I} .

As a special case, if the point spread function is spatially invariant, we have $h_{\mathbf{u}_j}(\mathbf{x}_i - \mathbf{u}_j) = h(\mathbf{x}_i - \mathbf{u}_j)$. This will recover the familiar spatially invariant convolution we see in the image deconvolution literature, i.e.,

$$I(\mathbf{x}_i) = \sum_{j=1}^N h(\mathbf{x}_i - \mathbf{u}_j) J(\mathbf{u}_j), \quad i = 1, \dots, N. \quad (5.7)$$

The resulting convolution matrix is then a *circulant* matrix that is diagonalizable by the discrete Fourier transform matrix.

Inspecting the operator \mathcal{H}_{θ} , we recognize that any *inverse problem* has to involve some kind of inverse mapping $\mathcal{H}_{\theta}^{-1}$. However, directly inverting \mathcal{H}_{θ} can be difficult because \mathcal{H}_{θ} consists of not only blurs but also tilts. The question we want to address here is how to decouple the blur and tilt as we model the turbulence.

5.1.2 Decoupling Tilt and Blur

There are two options we can consider when decoupling \mathcal{H}_{θ} . The first one is the **blur-then-tilt** model, where we first blur the image using

CHAPTER 5. IMAGE RESTORATION

spatially varying blurs and then add tilts to the blurred image:

$$\mathbf{I} = \mathcal{H}_\theta(\mathbf{J}) = [\mathcal{T} \circ \mathcal{B}](\mathbf{J}) = \mathcal{T}(\mathcal{B}(\mathbf{J})).$$

Here, “ \circ ” denotes the function composition of two operators.

The second one is the **tilt-then-blur** model, where we first add tilts to the image pixels and then blur the image using spatially varying blurs:

$$\mathbf{I} = \mathcal{H}_\theta(\mathbf{J}) = [\mathcal{B} \circ \mathcal{T}](\mathbf{J}) = \mathcal{B}(\mathcal{T}(\mathbf{J})).$$

The difference between the **tilt operator** \mathcal{T} and the **blur operator** \mathcal{B} is the Zernike modes we use to construct the phase distortion:

- Tilt \mathcal{T} : The phase is distorted by the first two Zernike coefficients via

$$\phi_{\mathbf{u}}(\boldsymbol{\rho}) = \sum_{m=2}^3 a_{\mathbf{u},m} Z_m(\boldsymbol{\rho}).$$

Recall that the first two Zernike modes (ignoring the DC term) represent the horizontal and the vertical displacements, i.e., $Z_2(\boldsymbol{\rho}) = 2\rho_x$ and $Z_3(\boldsymbol{\rho}) = 2\rho_y$ where $\boldsymbol{\rho} = [\rho_x, \rho_y]^T$ is the coordinate. Thus, if we define $\boldsymbol{\alpha}_{\mathbf{u}} = [\lambda a_{\mathbf{u},2}/R, \lambda a_{\mathbf{u},3}/R]^T$, then $\phi_{\mathbf{u}}(\boldsymbol{\rho})$ can be written as $\phi_{\mathbf{u}}(\boldsymbol{\rho}) = \boldsymbol{\alpha}_{\mathbf{u}}^T \boldsymbol{\rho}$.

- Blur \mathcal{B} : The phase is distorted by all Zernike coefficients except the first two:

$$\phi_{\mathbf{u}}(\boldsymbol{\rho}) = \sum_{m=4}^{\infty} a_{\mathbf{u},m} Z_m(\boldsymbol{\rho}).$$

If we subtract the overall phase from the tilt, we can obtain the following decomposition:

$$\phi_{\mathbf{u}}(\boldsymbol{\rho}) = \underbrace{\sum_{m=2}^3 a_{\mathbf{u},m} Z_m(\boldsymbol{\rho})}_{\boldsymbol{\alpha}_{\mathbf{u}}^T \boldsymbol{\rho}} + \underbrace{\sum_{m=4}^{\infty} a_{\mathbf{u},m} Z_m(\boldsymbol{\rho})}_{\varphi_{\mathbf{u}}(\boldsymbol{\rho})}$$

In other words, the overall phase can be written as a linear term plus a tilt-free phase term.

Once the phase distortions are defined, the resulting point spread function is

$$h_{\mathbf{u}}(\mathbf{x}) = |\mathfrak{Fourier}(e^{-j\phi_{\mathbf{u}}(\boldsymbol{\rho})})|^2. \quad (5.8)$$

5.1. UNDERSTANDING THE FORWARD MODEL

Since the Zernike coefficients used to define the phase distortion are random, the blur and tilt operators \mathcal{B} and \mathcal{T} are both spatially varying and random.

Inspecting the decomposition of the phase, we notice that the exponential functions have their respective Fourier transform pairs

$$e^{-j\phi_{\mathbf{u}}(\rho)} = \underbrace{e^{-j\alpha_{\mathbf{u}}^T \rho}}_{\longleftrightarrow \delta(\mathbf{u} - \alpha_{\mathbf{u}})} \cdot \underbrace{e^{-j\varphi_{\mathbf{u}}(\rho)}}_{\longleftrightarrow b_{\mathbf{x}}(\mathbf{u})},$$

where $\delta(\mathbf{u} - \alpha_{\mathbf{u}})$ is the delta function representing the tilt and $b_{\mathbf{x}}(\mathbf{u})$ is the blur. Since multiplication in the Fourier space corresponds to convolution in the spatial domain, the resulting PSF is

$$h_{\mathbf{u}}(\mathbf{x}) = \delta(\mathbf{x} - \alpha_{\mathbf{u}}) \circledast b_{\mathbf{u}}(\mathbf{x}) = b_{\mathbf{u}}(\mathbf{x} - \alpha_{\mathbf{u}}). \quad (5.9)$$

Therefore, the PSF is a shifted version of the blur.

Now, let's take a closer look at the tilt-then-blur and blur-then-tilt operations. In terms of matrix-vector notation, the operator \mathcal{T} can be written as a shifting matrix $\mathbf{T} \in \mathbb{R}^{N \times N}$ with

$$\mathbf{T} = \begin{bmatrix} t_{\mathbf{u}_1}(\mathbf{x}_1) & t_{\mathbf{u}_2}(\mathbf{x}_1) & \dots & t_{\mathbf{u}_N}(\mathbf{x}_1) \\ t_{\mathbf{u}_1}(\mathbf{x}_2) & t_{\mathbf{u}_2}(\mathbf{x}_2) & \dots & t_{\mathbf{u}_N}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ t_{\mathbf{u}_1}(\mathbf{x}_N) & t_{\mathbf{u}_2}(\mathbf{x}_N) & \dots & t_{\mathbf{u}_N}(\mathbf{x}_N) \end{bmatrix}. \quad (5.10)$$

The (i, j) th entry of \mathbf{T} is $t_{\mathbf{u}_j}(\mathbf{x}_i)$. If a pixel located at \mathbf{u}_j is relocated to location \mathbf{x}_i , then $t_{\mathbf{u}_j}(\mathbf{x}_i) = 1$. Otherwise $t_{\mathbf{u}_j}(\mathbf{x}_i) = 0$. For example, if \mathbf{T} shifts the entire image by one pixel, then it is

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \quad (5.11)$$

Remark: If the shifting operator needs to include subpixel shifts, we can allow each row to be a local kernel (e.g. a Gaussian kernel) such that the values sum to one.

The operator \mathcal{B} is a collection of tilt-free but spatially varying blurs. In the matrix notation, we can define a matrix \mathbf{B} where $[\mathbf{B}]_{ij} = b_{\mathbf{u}_j}(\mathbf{x}_i)$ where $b_{\mathbf{u}_j}$ is the tilt-free blur located at \mathbf{u}_j . As before, $b_{\mathbf{u}_j}$ is generated by the phase distortion at \mathbf{u}_j using high-order Zernike

coefficients. The value $b_{\mathbf{u}_j}(\mathbf{x}_i)$ is the tilt-free blur $b_{\mathbf{u}_j}$ evaluated at pixel \mathbf{x}_i . The overall structure of the matrix \mathbf{B} is

$$\mathbf{B} = \begin{bmatrix} b_{\mathbf{u}_1}(\mathbf{x}_1) & b_{\mathbf{u}_2}(\mathbf{x}_1) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_1) \\ b_{\mathbf{u}_1}(\mathbf{x}_2) & b_{\mathbf{u}_2}(\mathbf{x}_2) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ b_{\mathbf{u}_1}(\mathbf{x}_N) & b_{\mathbf{u}_2}(\mathbf{x}_N) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_N) \end{bmatrix}. \quad (5.12)$$

The difference between $b_{\mathbf{u}_j}(\mathbf{x}_i)$ and $h_{\mathbf{u}_j}(\mathbf{x}_i)$ is the presence of the shifting action by the tilt \mathbf{T} . In fact, the source of tension is about whether $\mathcal{H}_{\boldsymbol{\theta}} = \mathcal{T} \circ \mathcal{B}$ (i.e. $\mathbf{H} = \mathbf{T}\mathbf{B}$?) or $\mathcal{H}_{\boldsymbol{\theta}} = \mathcal{B} \circ \mathcal{T}$ (i.e. $\mathbf{H} = \mathbf{B}\mathbf{T}$?).

5.1.3 Tilt-then-Blur or Blur-then-Tilt?

To understand the difference between $\mathcal{T} \circ \mathcal{B}$ and $\mathcal{B} \circ \mathcal{T}$, we consider a grid of point sources as shown in Figure 5.2. The tilt map is synthesized using the first two Zernike coefficients with some spatial correlation (aka the angle of arrival correlation). The blur in this example is a spatially invariant Gaussian blur so that the visualization is clearer.

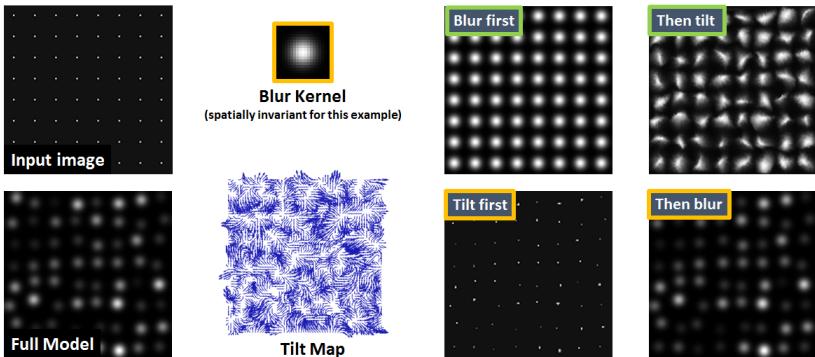


Figure 5.2: Comparing blur-then-tilt $\mathcal{T} \circ \mathcal{B}$ and tilt-then-blur $\mathcal{B} \circ \mathcal{T}$. Given a grid of point sources, a spatially invariant blur, and a dense tilt map, the result of $\mathcal{T} \circ \mathcal{B}$ (the green case) shows a grid of destroyed blurs whereas the result of $\mathcal{B} \circ \mathcal{T}$ (the yellow case) shows a grid of shifted blurs. We remark that $\mathcal{T} \circ \mathcal{B}$ is incorrect whereas $\mathcal{B} \circ \mathcal{T}$ is correct.

Ideally, if we apply $\mathcal{H}_{\boldsymbol{\theta}}$ to this grid of points, we should expect that the output is a grid of shifted blurs. The reason is that if a single

5.1. UNDERSTANDING THE FORWARD MODEL

point source (a delta function) is convolved with a shifted blur (a point spread function constructed using *all* Zernike coefficients), then the output must be the tilt-free blur shifted to the new location specified by the tilt.

Inspecting the case of *tilt-then-blur* $\mathcal{B} \circ \mathcal{T}$, we see a perfect match with the full operator \mathcal{H}_θ . This can be understood as first starting with an unshifted delta function. When we apply the tilt to the delta function, the delta function is moved to a new position specified by the amount of the tilt. The result of this tilting operation to the grid of points is then a grid of shifted delta functions. If we apply a blur, then the blur will change the deltas to blurs. This is exactly the same as the original operator \mathcal{H}_θ .

Mathematically, we can show that the original operator \mathcal{H}_θ is a set of shifted blurs. For example, if we assume that the tilt is globally shifting *all* the tilt-free blurs $\{b_{\mathbf{x}_i}(\mathbf{u}) | i = 1, \dots, N\}$ by one pixel, then each row of \mathbf{H} will be a shifted blur with a one-pixel shift. The i th blur is still located at the i th row, but the entries of the row are $h_{\mathbf{u}_j} = [b_{\mathbf{x}_1}(\mathbf{u}_2), b_{\mathbf{x}_1}(\mathbf{u}_3), \dots, b_{\mathbf{x}_1}(\mathbf{u}_N), 0]$. Putting all the entries together, we can show that \mathbf{H} is the matrix:

$$\mathbf{H} = \begin{bmatrix} b_{\mathbf{u}_2}(\mathbf{x}_1) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_1) & 0 \\ b_{\mathbf{u}_2}(\mathbf{x}_2) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_2) & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_{\mathbf{u}_2}(\mathbf{x}_N) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_N) & 0 \end{bmatrix}. \quad (5.13)$$

With a simple calculation, we can show that this matrix is identical to \mathbf{BT} :

$$\begin{aligned} \mathbf{BT} &= \begin{bmatrix} b_{\mathbf{u}_1}(\mathbf{x}_1) & b_{\mathbf{u}_2}(\mathbf{x}_1) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_1) \\ b_{\mathbf{u}_1}(\mathbf{x}_2) & b_{\mathbf{u}_2}(\mathbf{x}_2) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ b_{\mathbf{u}_1}(\mathbf{x}_N) & b_{\mathbf{u}_2}(\mathbf{x}_N) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \\ &= \begin{bmatrix} b_{\mathbf{u}_2}(\mathbf{x}_1) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_1) & 0 \\ b_{\mathbf{u}_2}(\mathbf{x}_2) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_2) & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_{\mathbf{u}_2}(\mathbf{x}_N) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_N) & 0 \end{bmatrix} = \mathbf{H}. \end{aligned} \quad (5.14)$$

Therefore, in this example, we have shown that $\mathbf{H} = \mathbf{BT}$.

If we use the *blur-then-tilt* model $\mathcal{T} \circ \mathcal{B}$, the result in Figure 5.2 shows that the blurs are not shifted but destroyed. The reason is that

CHAPTER 5. IMAGE RESTORATION

if we first blur the grid of points, the result is a grid of blurs. When we apply the tilt map to the grid of blurs, each tilt vector will move its corresponding pixel value to a different location. In other words, since the tilt map is dense, every pixel has a tilt vector. These include pixels that are not the delta functions; we refer to the tilt vectors at these locations as the non-centered tilts. In the full model \mathcal{H}_θ and the blur-after-tilt $\mathcal{B} \circ \mathcal{T}$ model, these off-centered tilts are *not* used because the tilt vectors are only valid for the centered tilts. In the blur-then-tilt model $\mathcal{T} \circ \mathcal{B}$, the grid of blurred points will have non-zero values at those non-centered tilts. As a result, the non-centered tilts will move the blurred pixel values to a different location. From the image formation perspective, this is a grid of destroyed point spread functions.

In terms of matrices and vectors, we can show that

$$\begin{aligned} \mathbf{TB} &= \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} b_{\mathbf{u}_1}(\mathbf{x}_1) & b_{\mathbf{u}_2}(\mathbf{x}_1) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_1) \\ b_{\mathbf{u}_1}(\mathbf{x}_2) & b_{\mathbf{u}_2}(\mathbf{x}_2) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ b_{\mathbf{u}_1}(\mathbf{x}_N) & b_{\mathbf{u}_2}(\mathbf{x}_N) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_N) \end{bmatrix} \\ &= \begin{bmatrix} b_{\mathbf{u}_1}(\mathbf{x}_2) & b_{\mathbf{u}_2}(\mathbf{x}_2) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ b_{\mathbf{u}_1}(\mathbf{x}_N) & b_{\mathbf{u}_2}(\mathbf{x}_N) & \dots & b_{\mathbf{u}_N}(\mathbf{x}_N) \\ 0 & 0 & 0 & 0 \end{bmatrix} \neq \mathbf{H}. \end{aligned} \quad (5.15)$$

To summarize, we conclude the difference between blur-then-tilt and tilt-then-blur in the following theorem.

Theorem 5.1 (Decomposition of Tilt and Blur). *If we decompose the turbulence operator \mathcal{H}_θ as the product of the tilt \mathcal{T} and the blur \mathcal{B} , then*

$$\mathcal{H}_\theta = \mathcal{B} \circ \mathcal{T} \quad \text{and} \quad \mathcal{H}_\theta \neq \mathcal{T} \circ \mathcal{B}. \quad (5.16)$$

5.1.4 Geometric Analysis

The above intuition can be formalized by a simple analysis. We consider a simplified situation where the turbulence is the combination of a dense field of tilts and a spatially *invariant* blur. We want to show that under this simplified condition, the two operators $\mathcal{T} \circ \mathcal{B}$ and $\mathcal{B} \circ \mathcal{T}$ are different.

5.1. UNDERSTANDING THE FORWARD MODEL

Blur-then-tilt $\mathcal{T} \circ \mathcal{B}$

Let's start by considering the spatially invariant blur. We define a template function $g(\mathbf{x})$. For example, we can think of this template function as a Gaussian blur kernel. For simplicity, we assume that this Gaussian blur kernel is spatially invariant. This spatially invariant blur will give us the blur $b_{\mathbf{u}_j}(\mathbf{x})$ centered at pixel \mathbf{u}_j as a shifted version of the template function:

$$b_{\mathbf{u}_j}(\mathbf{x}) = g(\mathbf{x} - \mathbf{u}_j). \quad (5.17)$$

When this tilt-free blur is applied to the clean image $J(\mathbf{x})$, we obtain the tilt-free blurred image

$$I_{\mathcal{B}}(\mathbf{x}_i) = \sum_{j=1}^N b_{\mathbf{u}_j}(\mathbf{x}_i) J(\mathbf{u}_j) = \sum_{j=1}^N g(\mathbf{x}_i - \mathbf{u}_j) J(\mathbf{u}_j). \quad (5.18)$$

For the blur-then-tilt model $\mathcal{T} \circ \mathcal{B}$, the tilt \mathbf{t}_i assigns $I_{\mathcal{B}}(\mathbf{x}_i)$ to a new pixel location $\mathbf{x}_i + \mathbf{t}_i$ of the final image $I_{\mathcal{T} \circ \mathcal{B}}$. That is,

$$I_{\mathcal{T} \circ \mathcal{B}}(\mathbf{x}_i + \mathbf{t}_i) = I_{\mathcal{B}}(\mathbf{x}_i).$$

Letting $\mathbf{v}_i = \mathbf{x}_i + \mathbf{t}_i$, this becomes $I_{\mathcal{T} \circ \mathcal{B}}(\mathbf{v}_i) = I_{\mathcal{B}}(\mathbf{v}_i - \mathbf{t}_i)$. Since \mathbf{v}_i is a dummy variable, we can replace \mathbf{v}_i with \mathbf{x}_i . Thus,

$$I_{\mathcal{T} \circ \mathcal{B}}(\mathbf{x}_i) = \sum_{j=1}^N g(\mathbf{x}_i - \mathbf{t}_i - \mathbf{u}_j) J(\mathbf{u}_j). \quad (5.19)$$

Tilt-then-blur $\mathcal{B} \circ \mathcal{T}$

The tilt-then-blur model behaves differently. We first tilt the image by defining

$$I_{\mathcal{T}}(\mathbf{u}_j + \mathbf{t}_j) = J(\mathbf{u}_j). \quad (5.20)$$

Letting $\mathbf{v}_j = \mathbf{u}_j + \mathbf{t}_j$, it follows that $I_{\mathcal{T}}(\mathbf{v}_j) = J(\mathbf{v}_j - \mathbf{t}_j)$. Now, if we blur this shifted image, we will have

$$I_{\mathcal{B} \circ \mathcal{T}}(\mathbf{x}_i) = \sum_{j=1}^N b_{\mathbf{v}_j}(\mathbf{x}_i) I_{\mathcal{T}}(\mathbf{v}_j) = \sum_{j=1}^N g(\mathbf{x}_i - \mathbf{v}_j) J(\mathbf{v}_j - \mathbf{t}_j).$$

Replacing $\mathbf{u}_j = \mathbf{v}_j - \mathbf{t}_j$, it follow that

$$I_{\mathcal{B} \circ \mathcal{T}}(\mathbf{x}_i) = \sum_{j=1}^N g(\mathbf{x}_i - \mathbf{t}_j - \mathbf{u}_j) J(\mathbf{u}_j). \quad (5.21)$$

The Roles of \mathbf{t}_i and \mathbf{t}_j

Comparing Equation (5.19) and Equation (5.21), the difference between the two equations is that for $\mathcal{T} \circ \mathcal{B}$, the tilt is \mathbf{t}_i whereas for $\mathcal{B} \circ \mathcal{T}$ the tilt is \mathbf{t}_j . In the case of $\mathcal{T} \circ \mathcal{B}$, the tilt \mathbf{t}_i is applied to the *output*. That is, we blur the image (drawn as a circle in Figure 5.3) and move the output by \mathbf{t}_i . Since each output pixel experiences a different \mathbf{t}_i , the blur is destroyed. For $\mathcal{B} \circ \mathcal{T}$, although the situation is more complicated because there are N tilts $\{\mathbf{t}_j \mid j = 1, \dots, N\}$, they perturb the location of the *input*. Therefore, while each \mathbf{x}_i sees N tilts, the N tilts are common for every \mathbf{x}_i . The shape of the blur is thus preserved.

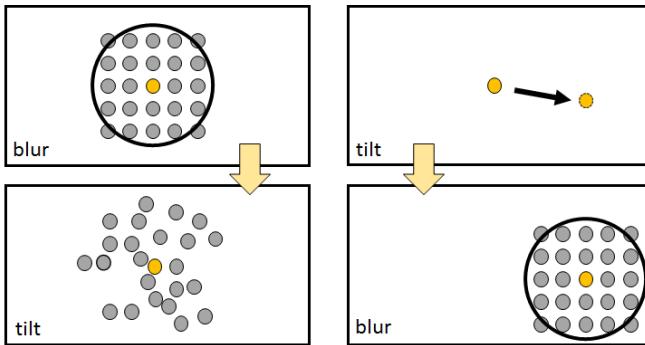


Figure 5.3: $\mathcal{T} \circ \mathcal{B}$ moves the center to a new location, whereas $\mathcal{B} \circ \mathcal{T}$ moves the points of the blur to different locations.

5.1.5 Impact to Real Images

Although the two operators $\mathcal{T} \circ \mathcal{B}$ and $\mathcal{B} \circ \mathcal{T}$ are theoretically different, their impacts on real images are less so. This is particularly true when the underlying image is a natural image.

To see why this is the case, rewrite Equation (5.19) and Equation (5.21) as

$$\begin{aligned} I_{\mathcal{T} \circ \mathcal{B}}(\mathbf{x}_i) &= \sum_{j=1}^N g(\mathbf{x}_i - \mathbf{u}_j) J(\mathbf{u}_j - \mathbf{t}_i), \\ I_{\mathcal{B} \circ \mathcal{T}}(\mathbf{x}_i) &= \sum_{j=1}^N g(\mathbf{x}_i - \mathbf{u}_j) J(\mathbf{u}_j - \mathbf{t}_j). \end{aligned}$$

5.1. UNDERSTANDING THE FORWARD MODEL

Then, by approximating $J(\mathbf{u}_j - \mathbf{t}_i)$ and $J(\mathbf{u}_j - \mathbf{t}_j)$ to the first order, the pointwise difference between $I_{\mathcal{T} \circ \mathcal{B}}(\mathbf{x}_i)$ and $I_{\mathcal{B} \circ \mathcal{T}}(\mathbf{x}_i)$ can be evaluated as

$$\begin{aligned} & I_{\mathcal{T} \circ \mathcal{B}}(\mathbf{x}_i) - I_{\mathcal{B} \circ \mathcal{T}}(\mathbf{x}_i) \\ &= \sum_{j=1}^N g(\mathbf{x}_i - \mathbf{u}_j) \left[J(\mathbf{u}_j - \mathbf{t}_i) - J(\mathbf{u}_j - \mathbf{t}_j) \right] \\ &\approx \sum_{j=1}^N \underbrace{g(\mathbf{x}_i - \mathbf{u}_j)}_{\text{convolution}} \underbrace{\nabla J(\mathbf{u}_j)^T}_{\text{image gradient}} \underbrace{(\mathbf{t}_i - \mathbf{t}_j)}_{\text{random tilt}}. \end{aligned} \quad (5.22)$$

distorted image gradient

An intuitive argument here is that $\mathbf{t}_i - \mathbf{t}_j$ is the difference between two tilt vectors. Since each tilt is a zero-mean Gaussian random vector, the difference remains a zero-mean Gaussian random vector. Although they are not white Gaussian, they are nevertheless *noise*. If there is a large ensemble average of these noise vectors, the result will be close to zero.

So, where does the average come from? There is a convolution by $g(\mathbf{x} - \mathbf{u})$. If the support of this blur kernel is large, then many of the noise vectors will be added and this will result in a small value. However, for an image with a large field of view, the relative size of the blur kernel g is usually not big (at most 30×30 for a 256×256 image). So, there must be another source that makes the error small.

The main reason why natural images tend to show a less difference between $\mathcal{T} \circ \mathcal{B}$ and $\mathcal{B} \circ \mathcal{T}$ is that the image gradient $\nabla J(\mathbf{u}_j)$ is typically *sparse*. For most parts, the gradient is zero except for edges. (Textures are less of a problem because they will be smoothed by the blur.) When $\nabla J(\mathbf{u}_j)$ is multiplied with the noise vector $\mathbf{t}_i - \mathbf{t}_j$, the result is an edge map with noise multiplied by every pixel. Convolving $\nabla J(\mathbf{u}_j)^T(\mathbf{t}_i - \mathbf{t}_j)$ with a blur kernel g will further smooth out the variations. Figure 5.4 shows a typical example with some standard optical configurations. The difference is not noticeable.

5.1.6 From Forward Model to Inverse Problems

So far we have shown that if we agree upon the phase decomposition in terms of the Zernike representation, then the correct turbulence forward model is tilt-then-blur. As a result, we need to be careful about statements such as

CHAPTER 5. IMAGE RESTORATION

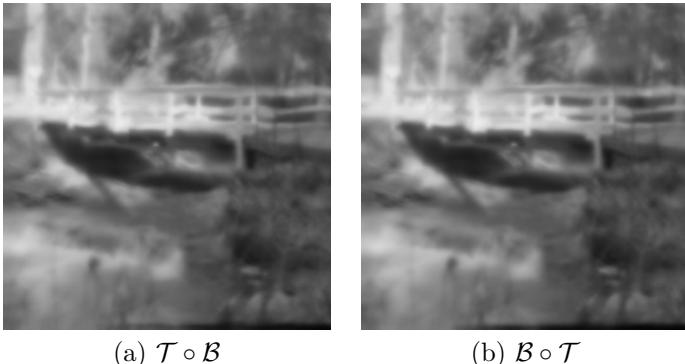


Figure 5.4: Simulated turbulence using $\mathcal{T} \circ \mathcal{B}$ and $\mathcal{B} \circ \mathcal{T}$. We notice that although theoretically the two are different, visually the two are nearly indistinguishable.

“... The image formation of atmospheric turbulence follows the equation $\mathbf{I} = \mathcal{T}(\mathcal{B}(\mathbf{J}))$... ”

It is more appropriate to instead state that the image formation model is $\mathbf{I} = \mathcal{B}(\mathcal{T}(\mathbf{J}))$ and comment that for natural images it can be approximated by $\mathbf{I} = \mathcal{T}(\mathcal{B}(\mathbf{J}))$.

From the standpoint of solving **inverse problems**, the issue becomes more interesting. The inverse problem, expressed in terms of the maximum-a-posteriori estimation, is [110, 111, 83, 112, 113]

$$\hat{\mathbf{J}} = \underset{\mathbf{J}}{\operatorname{argmin}} \quad \|\mathbf{I} - \mathcal{H}_{\boldsymbol{\theta}}(\mathbf{J})\|^2 + \lambda g(\mathbf{J}), \quad (5.23)$$

where $g(\mathbf{J})$ is the regularization function, also known as the image prior, that confines the search space of the solution \mathbf{J} .

The optimization in Equation (5.23) should look familiar to readers who know deconvolution. In the special case where $\mathcal{H}_{\boldsymbol{\theta}}$ is a spatially invariant blur, Equation (5.23) resembles the classical deconvolution problem. Therefore, any classical/modern solution that has been invented to solve deconvolution can become a candidate for Equation (5.23). For turbulence image restoration, approaches following this line of thoughts are abundant, e.g., [114, 115, 116, 117, 110, 118, 119, 120].

In the presence of turbulence, solving Equation (5.23) is more involved because we do not know $\mathcal{H}_{\boldsymbol{\theta}}$ which is the instantaneous turbu-

5.1. UNDERSTANDING THE FORWARD MODEL

lence operator. It is a random realization drawn from a certain probability distribution. Since we only have one observation of the distorted image \mathbf{I} , recovering \mathcal{H}_θ can be as hard as recovering \mathbf{J} .

There are two options that are worth discussing. Firstly, we can assume that instead of only one observation, we have T observations of a static scene. In this case, the optimization objective can be written as a sum of individual turbulence terms:

$$\widehat{\mathbf{J}} = \operatorname{argmin}_{\mathbf{J}} \sum_{t=1}^T \|\mathbf{I}(t) - \mathcal{H}_\theta^{(t)}(\mathbf{J})\|^2 + \lambda g(\mathbf{J}), \quad (5.24)$$

where $\mathcal{H}_\theta^{(t)}$ is the random realization of the turbulence distortion at time t . Solving the new equation is slightly easier because if we knew $\mathcal{H}_\theta^{(t)}$, then we will have more degrees of freedom. However, if we do not know $\mathcal{H}_\theta^{(t)}$, the new problem is as difficult as the original problem.

The other option is to decouple \mathcal{H} into \mathcal{T} and \mathcal{B} as follows.

$$\widehat{\mathbf{J}} = \operatorname{argmin}_{\mathbf{J}} \|\mathbf{I} - \mathcal{B}(\mathcal{T}(\mathbf{J}))\|^2 + \lambda g(\mathbf{J}). \quad (5.25)$$

This optimization can further be written as

$$\widehat{\mathbf{U}} = \operatorname{argmin}_{\mathbf{U}} \|\mathbf{I} - \mathcal{B}(\mathbf{U})\|^2 + \lambda g(\mathbf{U}), \quad (5.26)$$

$$\widehat{\mathbf{J}} = \operatorname{argmin}_{\mathbf{J}} \|\widehat{\mathbf{U}} - \mathcal{T}(\mathbf{J})\|^2 + \lambda g(\mathbf{J}). \quad (5.27)$$

This is a two-step approach. After we have recovered the intermediate variable $\widehat{\mathbf{U}}$, we can recover $\widehat{\mathbf{J}}$. The difficulty of this approach is that when solving for \mathbf{U} , we need to estimate the spatially varying blur \mathcal{B} . This task is as difficult as the original problem because we need to recover the blur kernel at every pixel.

Today's turbulence mitigation algorithms are mostly inspired by the **lucky imaging** concept which will be discussed in the next subsection. The core idea is to *flip* the order of the blur and the tilt so that the optimization is

$$\widehat{\mathbf{J}} = \operatorname{argmin}_{\mathbf{J}} \|\mathbf{I} - \mathcal{T}(\mathcal{B}(\mathbf{J}))\|^2 + \lambda g(\mathbf{J}). \quad (5.28)$$

As discussed earlier, tilt-then-blur and blur-then-tilt are different. However, for natural images (i.e., not point sources), the visual difference may not be too obvious. The benefit of flipping the order of tilt and

blur is that estimating the tilt from a stack of images is feasible. Moreover, after the lucky imaging step, a significant portion of the spatially varying blur will be removed because we will be fusing sharp regions of the images. As a result, \mathcal{T} can be inverted to a reasonable degree of accuracy, and \mathcal{B} will become easier because the spatially varying part of the blur will be suppressed. In the following Chapter, we will discuss these two steps one by one.

Tilt-Blur $\mathcal{T} \circ \mathcal{B}$ or Blur-Tilt $\mathcal{B} \circ \mathcal{T}$?

- The correct model is $\mathcal{B} \circ \mathcal{T}$. So, for simulation, we must use $\mathcal{B} \circ \mathcal{T}$.
- For solving inverse problems, it does not matter. It does not mean that we shouldn't use $\mathcal{T} \circ \mathcal{B}$; it's just that there are other considerations. For example, inverting the blur is often much harder than inverting the tilt.

5.2 Lucky Imaging

After discussing the forward imaging model, we now discuss methods to solve the inverse problem. Chapter 5.2 is dedicated to the concept of lucky imaging. The goal of lucky imaging is to handle the tilt operator \mathcal{T} and remove some of the spatially varying blur \mathcal{B} in Equation (5.28).

5.2.1 Lucky Probability via Zernike

Atmospheric turbulence is a random process caused by random phase distortion. If we model the phase using the Zernike representation, the randomness is encoded by the Zernike coefficients which are sampled from a probability distribution. Among many properties, the probability distribution is a high dimensional *zero-mean* Gaussian according to Tatarskii [22]. Thus, if we observe a Zernike coefficient over a long period of time, the ensemble average will be zero. What this means is that, from time to time, and/or across the spatial locations, we will see instants where the Zernike coefficients are approximately zero. The probability of getting a zero Zernike vector is not high, but in theory, it is possible. When this happens, we say that there is a **lucky event**. The resulting image is called a **lucky image** [27].

5.2. LUCKY IMAGING

To gain more intuition of the lucky image, we can conduct an experiment by putting a heat source in the optical path close to the camera. Since the heat is localized, the turbulence is **isoplanatic**, i.e., the entire field of view will experience the same turbulence. If we use a high-speed shutter to capture the frames, we will see that occasionally the image is sharp. Figure 5.5 shows an example, where we see that an unlucky frame suffers from severe distortions whereas a lucky frame is distortion free.

In a truly long-range situation, the turbulent effects become **anisoplanatic**. Across different regions of the image, we can still observe the lucky effect but it appears locally. Some regions of the image will experience a lucky effect while another region of the same image may not. In this case, we say that there is a local lucky effect, and the corresponding lucky region is known as a lucky *patch*.



Figure 5.5: An lucky event can be observed by placing a local heat source near the camera. This will create isoplanatic turbulence where the image is globally distorted. Over a period of observations, we will see distorted and distortion-free frames. A key for this experiment is to ensure that the exposure is short enough that motion blur is not accumulated.

The lucky event can be described mathematically by considering the PSFs. Figure 5.6 shows a grid of PSFs resulting from passing point sources through a turbulence simulator. For this grid of PSFs, some PSFs are nearly a delta function, e.g., $h_{\mathbf{u}}(\mathbf{x}) \approx \delta(\mathbf{x} - \mathbf{u})$.¹ If a PSF is a delta function, the convolution with an image will be distortion free. Such a phenomenon can happen in space across a large field of view. It can also happen in time over a long period of observation. If we use short exposures to capture a large number of frames over time, then

¹Technically speaking, the PSF will never become a delta function because the finite aperture of the camera will at best give us a diffraction-limited blur. So, instead of having a pure delta function, we will obtain a diffraction-limited PSF.

CHAPTER 5. IMAGE RESTORATION

occasionally we will see a delta function for the PSF.

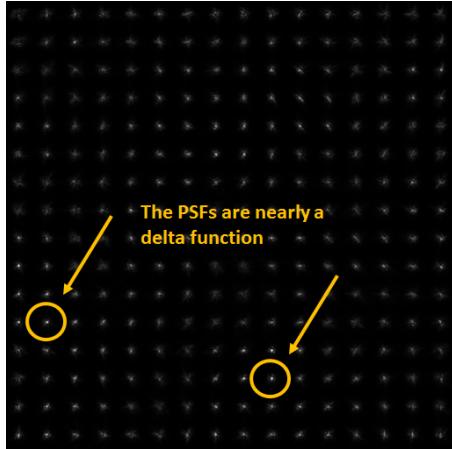


Figure 5.6: A grid of point sources after going through turbulence simulation. From time to time, and from location to location, the PSF can be close to a delta function. We say that when this happens, there is a lucky event.

We can define the lucky event via the Zernike coefficients.

Definition 5.1 (Lucky event). Let $\mathbf{a}_\mathbf{u} = [a_{\mathbf{u},m} \mid m = 2, \dots, M]$ be the Zernike coefficients for a pixel coordinate \mathbf{u} , a lucky event happens when

$$\text{lucky event at pixel } \mathbf{u} = \left\{ \|\mathbf{a}_\mathbf{u}\|_2^2 \leq \tau \right\}, \quad (5.29)$$

where τ is some user-defined threshold.

In plain words, a lucky event happens when the magnitude of the Zernike coefficients is small enough. The magnitude of the Zernike coefficients measures the amount of turbulence energy. If the scene is turbulence-free, then the phase distortion is $\phi_\mathbf{u}(\rho) = 0$, which is equivalent to requiring all Zernike coefficients to be zero, i.e., $\|\mathbf{a}_\mathbf{u}\|_2 = 0$. As turbulence becomes stronger, $\|\mathbf{a}_\mathbf{u}\|_2$ will grow.

The analysis of the lucky probability was first documented by David Fried in his 1978 paper [27]. Fried's approach is similar to what

5.2. LUCKY IMAGING

we described above where he uses tilt-corrected phase functions. However, instead of using the Zernike functions as the basis, Fried considered the Karhunen–Loeve (KL) expansion. The reason is that the Zernike basis functions, although they are orthogonal, don't produce statistically independent coefficients. Therefore, if we consider a pair of coefficients $a_{\mathbf{u},i}$ and $a_{\mathbf{u},j}$, the joint expectation cannot be factorized as $\mathbb{E}[a_{\mathbf{u},i}a_{\mathbf{u},j}] \neq \mathbb{E}[a_{\mathbf{u},i}]\mathbb{E}[a_{\mathbf{u},j}]$, and hence we cannot ensure $\mathbb{E}[a_{\mathbf{u},i}a_{\mathbf{u},j}] = 0$ even if we know $\mathbb{E}[a_{\mathbf{u},i}] = 0$ for any i . This will cause difficulties when considering the norm $\mathbb{E}[\|\mathbf{a}_{\mathbf{u}}\|_2^2]$. The KL representation alleviates the difficulty because the basis functions are computed via the principal component analysis and so they are statistically independent.²

The definition of the lucky event by using the KL representation is the same as that using the Zernike representation. Let $\beta = [\beta_1, \dots, \beta_M]$ be the KL expansion coefficients, the lucky event can be defined as

$$\text{lucky event at pixel } \mathbf{u} = \left\{ \|\beta_{\mathbf{u}}\|_2^2 \leq \tau \right\}, \quad (5.30)$$

for some user-defined parameters τ . With the KL expansion, Fried showed the probability of obtaining a lucky event.

Theorem 5.2 (Probability of a Lucky Event). *Let $\beta_{\mathbf{u}}$ be the Karhunen–Loeve expansion coefficients of the phase function at pixel \mathbf{u} . The probability of obtaining a lucky event is*

$$\underbrace{\mathbb{P}\left[\|\beta_{\mathbf{u}}\|_2^2 \leq 1\right]}_{\text{probability of a lucky event}} \approx 5.6e^{-0.1557(D/r_0)^2}, \quad (5.31)$$

for $D/r_0 \geq 3.5$, where D is the aperture diameter and r_0 is the Fried parameter.

The prediction by Fried is useful in some ways but also limited in other ways. On the positive side, it does tell us the probability of obtaining a lucky event. The equation is interpretable because it is written in terms of D/r_0 . For stronger turbulence, D/r_0 increases and so Fried's formula predicts that the probability will drop.

The downside of the prediction can be seen in several ways. First, $D/r_0 \geq 3.5$ is quite big for many ground-to-ground imaging systems.

²The caveat is that the Karhunen–Loeve (KL) expansion is not as physically interpretable as the Zernike coefficients. Mathematically, the KL decomposition is in a non-closed form determined by the phase function's covariance equation.

CHAPTER 5. IMAGE RESTORATION

Typically, for ground-to-ground imaging (especially if we consider a passive imaging system that is incoherent), any $D/r_0 \geq 3.5$ is considered difficult for image restoration algorithms. For astronomical imaging, we might be able to handle a larger D/r_0 because we have access to more advanced tools such as adaptive optics.

Another reason why Fried's result has limited applicability for computer vision is that the lucky event Fried considered is within the isoplanatic angle. Isoplanatism means that the entire frame is distorted homogeneously by the same turbulent effect. For astronomical imaging, isoplanatic turbulence happens because the field of view is small. For ground-to-ground applications, especially those associated with computer vision tasks, the turbulence is anisoplanatic, implying that the distortions are spatially varying. Therefore, instead of seeing a lucky frame, we can only see lucky *patches*. To compute the probability of getting an algorithmically fused lucky frame, we need to compute the correlation across different patches. This remains an open problem, to our knowledge.

5.2.2 Optical Lucky Imaging

Lucky imaging can be achieved optically and digitally. The technique on the optical side is known as **adaptive optics**. This technique is mostly used for astronomical imaging because the star does not move (as in the sense of a car moving in the scene). Figure 5.7 shows a schematic diagram of an adaptive optics system [121, 38, 122].

An adaptive optics system requires a **reference star** in addition to the **target star** [124]. Usually, astronomers will look for a bright star near to the target star. Since we know the distance of the reference star, waves originating from the source can be measured and tracked. In adaptive optics, we use a wavefront sensor to record the incident wavefront, hence determining the instantaneous phase of the wave. If atmospheric turbulence is present, the measured phase will deviate from the theoretical prediction. A feedback signal measuring such a deviation will be sent to the control system which is connected to a set of adaptive mirrors. The adaptive mirrors rotate their angles in order to correct the phase lead and lag measured by the wavefront sensor. Once the phase lead and lag are compensated by the adaptive mirror, the optics will be used to capture the target. Since the optics have already been configured to compensate for the turbulence, the target star can be captured without any part of the phase distortions.

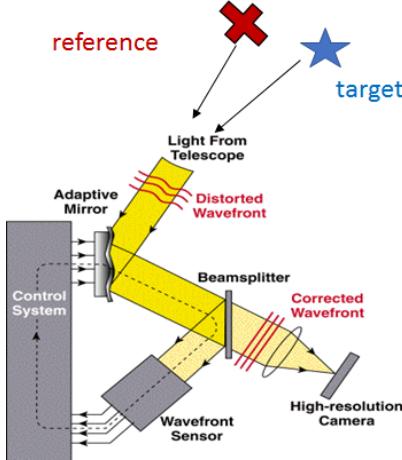


Figure 5.7: An adaptive optics system implements the concept of lucky imaging using optical instrumentation. With the help of deformable mirrors, the adaptive optics system measures the phase of the reference star and compensates for the phase distortion by moving the deformable mirrors [123]. In terms of turbulence correction, the random tilts are removed but the residue blur remains. Therefore, some deblurring algorithms are still needed to produce the result. Source: https://lyot.org/background/adaptive_optics.html.

The mirrors of the adaptive optics system have limited precision of the phase it can correct. In most of cases, the adaptive optics system can correct for the first-order phase deviation. Using our terminologies, this is exactly the first two Zernike coefficients representing the *tilt*. Mathematically, what adaptive optics seeks to find is the best linear fit so that we can decompose the phase as [23, 24]

$$\phi_u(\rho) = \underbrace{\alpha^T \rho}_{\text{linear term: tilt}} + \underbrace{\varphi_u(\rho)}_{\text{high order aberrations}} . \quad (5.32)$$

In this equation, the vector $\alpha = [\alpha_1, \alpha_2]$ represents the normal vector of the best fitted linear plane. An intuitive way of thinking about α is to treat them as the first two Zernike coefficients (which are the tilts). The adaptive optics system can compensate for this term, thus leaving us just the high order aberration term $\varphi_u(\rho)$. However, α is not exactly the Zernike coefficients because the Zernike basis functions

CHAPTER 5. IMAGE RESTORATION

are not statistically independent. Some high-order aberration terms can also have the tilting effect. Therefore, by performing the adaptive optics, we can sometimes compensate for more than just the first two Zernike coefficients.

The other limitation of the adaptive optics system is that its spatial resolution is limited by the number of deformable mirrors that can be installed and controlled. We should remind ourselves that the deformable mirror is adjusted mechanically. While advanced MEMS can be used to reduce the power and space, precise control of the angles of the mirrors is still challenging. The speed of the deformable mirrors is also an issue. We are not able to adjust the mirrors too quickly for moving scenes. Finally, adaptive optics often require a reference star. For many ground-to-ground imaging applications, this may not be available.

5.2.3 Digital Lucky Imaging: Sharpness Metric

Digital lucky imaging is based on capturing a stack of short-exposure images and then processing them using algorithms. The overall idea is to identify the distortion-free frames by finding which frames have the sharpest image content [125]. These frames will then be aggregated to construct a lucky image [126].

The idea of identifying the sharpest frame can be realized in multiple ways. In this Chapter, we mention a few of them:

Intensity Variance. In turbulence literature, the Strehl ratio is often used as a metric to predict the quality of a frame if a reference star is available. The Strehl ratio is the ratio between the peak of the intensity in the aberrated PSF and that of a diffraction-limited PSF. In [83], it was mentioned that the Strehl ratio has a relationship to the *variance* of the pixel intensity in a local neighborhood. More specifically, for a fixed coordinate \mathbf{x} , we consider a local neighborhood $\Omega_{\mathbf{u}}$ and compute the variance as

$$\mu_{\mathbf{x}} = \frac{1}{|\Omega_{\mathbf{u}}|} \sum_{\mathbf{s} \in \Omega_{\mathbf{u}}} I(\mathbf{s}),$$

$$V_{\mathbf{x}} = \frac{1}{|\Omega_{\mathbf{u}}|} \sum_{\mathbf{s} \in \Omega_{\mathbf{u}}} (I(\mathbf{s}) - \mu_{\mathbf{x}})^2.$$

If we plot the variance as a function of time, we can obtain a plot similar to Figure 5.8. A lucky frame is then identified if the variance is

5.2. LUCKY IMAGING

an *outlier* of the plot. There are different ways to determine if a pixel is an outlier. We refer readers to [83] for a hypothesis testing approach.

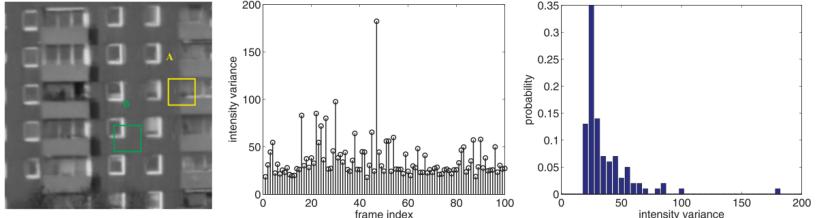


Figure 5.8: One way to identify sharp frames from a stack of turbulence images is to compute the local intensity and evaluate its variance. [Left] An image. [Middle] The variance of the patch as a function of time. [Right] Histogram of the variance. The idea here is to identify the outliers from the histogram. Image courtesy: [83].

Gradient. Another commonly used algorithmic strategy to identify a sharp image is by means of checking the gradient. The idea was reported in Joshi and Cohen [127] and later refined in [128]. To compute the gradient of a region, we calculate the horizontal and vertical finite differences:

$$\begin{aligned}\mathbf{g}_x &= \mathbf{I} \circledast [1, -1], \\ \mathbf{g}_y &= \mathbf{I} \circledast [1; -1].\end{aligned}$$

In this pair of equations, the filters $[1, -1]$ and $[1; -1]$ are the first-order horizontal and vertical difference operators, respectively. When we convolve (i.e., the \circledast operation) the input image \mathbf{I} with these filters, we effectively extract the edges of the image and store them as \mathbf{g}_x and \mathbf{g}_y , respectively. A compact notation for this operation is the gradient operator $\nabla : \mathbb{R}^N \rightarrow \mathbb{R}^{N \times 2}$:

$$[\mathbf{g}_x, \mathbf{g}_y] = \nabla \mathbf{I}.$$

The sharpness of the image can be computed via the norm of the gradient. For example, we can consider the **isotropic total variation** norm

$$\|\mathbf{I}\|_{\text{TV}_2} = \sqrt{\sum_{n=1}^N ([\mathbf{g}_x]_n^2 + [\mathbf{g}_y]_n^2)},$$

where $[\cdot]_n$ denotes the n -th element of the vector. Another approach is to consider the **anisotropic total variation norm**, defined as

$$\|\mathbf{I}\|_{\text{TV}_1} = \sum_{n=1}^N |[\mathbf{g}_x]_n| + |[\mathbf{g}_y]_n|.$$

If we want to compute the *local* gradient, we can replace the image \mathbf{I} by a patch $\underline{\mathbf{I}}(\mathbf{x})$ located at coordinate \mathbf{x} . If we further want to include the time axis, we can consider the patch located at \mathbf{x} at time t , i.e., $\underline{\mathbf{I}}(\mathbf{x}, t)$. Then, a sharpness metric δ^S can be defined as

$$\delta^S(\mathbf{x}, t) = \|\underline{\mathbf{I}}(\mathbf{x}, t)\|_{\text{TV}_1}. \quad (5.33)$$

Notice that the choice of the anisotropic or the isotropic total variation is not very important, because the two behave similarly.

Variations of the gradient are abundant. For example, instead of considering the horizontal or vertical gradient, we can consider the Laplacian of a local patch. In the age of deep learning, we can train a network to extract features; and from the features, we can identify the sharp patches.

5.2.4 Digital Lucky Imaging: Fusion Step

Suppose that we have a stack of T frames. How should we construct the lucky image? Shall we just use one of the sharpness metrics and pick *the sharpest*? The answer is no, because the so-called “sharpest” frame may not even be a good frame. For example, if for some reason an object in a frame is over-exposed, then there will be a surge in the gradient around the object boundary and background noise. If we simply just pick according to the gradient, then this poorly exposed frame will be picked. So in digital lucky imaging, we almost never pick the sharpest frame but a set of similarly sharp frames.

The way to accomplish the frame selection can be a hard decision (i.e., by setting a threshold) or a soft decision (which will be discussed shortly). The benefit of the hard decision is that we are more aggressive in terms of completely ignoring frames that are below the threshold, so their influence on the lucky image is eliminated. The downside, however, is that the quality of the lucky image will depend heavily on the threshold. Since there are no universal rules for choosing the threshold (besides a few statistical hypothesis techniques), it is sometimes better to consider the soft decision approach.

5.2. LUCKY IMAGING

In a soft decision approach, we **fuse** multiple images to construct one lucky image. Using the sharpness metric as an example, we can consider a weight:

$$w_{\mathbf{x},t}(\Delta t) = \exp \{ \alpha \|\underline{\mathbf{I}}(\mathbf{x}, t + \Delta t)\|_{\text{TV}_1} \}. \quad (5.34)$$

This definition involves two parts. The first part is the exponential function. We take the exponential function to scale the total variation norm. If a patch is sharp, $\|\underline{\mathbf{I}}(\mathbf{x}, t + \Delta t)\|_{\text{TV}_1}$ will be large and so the exponential function further amplifies the magnitude. The second part is the coordinate offset Δt . This offset is a running index that defines the weight, which is the weight for a reference pixel located at (\mathbf{x}, t) , and the weight is computed with respect to the temporal neighbors of (\mathbf{x}, t) . In other words, we are computing the weight along the time axis and finding out which frame contains the sharpest content.

Given the weight, we can now fuse the lucky patch, at location (\mathbf{x}, t) , as

$$\underline{\mathbf{I}}_{\text{lucky}}(\mathbf{x}, t) = \frac{\sum_{\Delta t \in \Omega_T} w_{\mathbf{x},t}(\Delta t) \underline{\mathbf{I}}(\mathbf{x}, t + \Delta t)}{\sum_{\Delta t \in \Omega_T} w_{\mathbf{x},t}(\Delta t)}. \quad (5.35)$$

This equation is reminiscent of the classical non-local means [129] where we construct weights to form a linear combination of pixels (or patches). The slight difference here is that we are forming the linear combination purely along the temporal axis because we are looking for a lucky patch in time.

In [128], it was commented that if we only do the above sharpness-based non-local averaging, we still have not overcome the limitation of the sharpness metric that it cannot differentiate a reliable sharp patch versus a poor-quality but a sharp patch. As such, it was proposed to include a **geometric consistency** metric:

$$\delta^G(\mathbf{x}, t) = \|\underline{\mathbf{I}}(\mathbf{x}, t) - \underline{\mathbf{I}}_{\text{ref}}(\mathbf{x}, t)\|^2. \quad (5.36)$$

Here, $\underline{\mathbf{I}}_{\text{ref}}$ is the **reference patch** constructed from the stack of frames. The reference patch is typically the temporal average of all the frames if the scene is static. As it is called, the reference patch is turbulence-jittering free so that it can serve the purpose of being a reference. Physically, this temporal average can also be considered a long-exposure image.

The meaning of the geometric consistency term is that we want to tell whether the current patch $\underline{\mathbf{I}}(\mathbf{x}, t)$ is a faithful sample in the

CHAPTER 5. IMAGE RESTORATION

stack of frames. If it is a good sample, then geometrically it should not deviate too much from the reference. The norm measuring the deviation between the two terms is therefore capturing the consistency.

When geometric consistency is added, the weight will become

$$w_{\mathbf{x},t}(\Delta t) = \underbrace{\exp \left\{ -\alpha_1 \|\underline{\mathbf{I}}(\mathbf{x}, t + \Delta t) - \underline{\mathbf{I}}_{\text{ref}}(\mathbf{x}, t + \Delta t)\|^2 \right\}}_{\text{geometric consistency weight}} \\ \times \underbrace{\exp \left\{ \alpha_2 \|\underline{\mathbf{I}}(\mathbf{x}, t + \Delta t)\|_{\text{TV}_1} \right\}}_{\text{sharpness weight}}. \quad (5.37)$$

We can now replace the weighted sum using this new weight. Note that in this definition, the exponential function has a negative exponent with respect to the geometric consistency. So, if there is a large deviation between the current and the reference, the weight will be small so that we can skip the geometrically inconsistent patch.

Figure 5.9 is a pictorial illustration of the impact of the geometric consistency term and the sharpness term. This figure is adapted from [128].

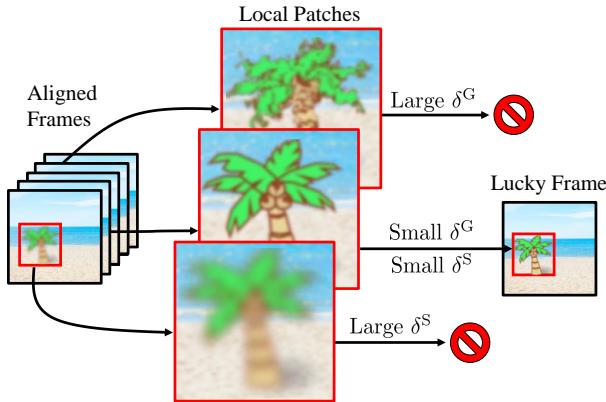


Figure 5.9: Lucky imaging patch selection requires the patch to be geometrically consistent with the reference so that it is no longer distorted by the pixel jittering of the turbulence. The geometrical consistency score is measured by δ^G . The other metric which is the sharpness metric is used to ensure that the selected patch is sufficiently sharp so that it is less affected by the atmospheric blur. Source: [128].

5.2. LUCKY IMAGING

From Lucky Patches to Lucky Image. When we apply any of the above lucky fusion methods, the result is a patch instead of an image. To fuse an image, we can stitch multiple patches together. If there are overlaps among the patches, we take the average of the overlapped regions. The approach is similar to how BM3D performs its aggregation step. Readers interested in this can consult [130].

Connecting to Inverse Problems. Once the lucky image is formed, the inverse problem becomes

$$\hat{\mathbf{J}} = \underset{\mathbf{J}}{\operatorname{argmin}} \|\mathbf{I}_{\text{lucky}} - \mathbf{h} \circledast \mathbf{J}\|^2 + \lambda g(\mathbf{J}). \quad (5.38)$$

In this equation, \mathbf{h} represents the residual blur remaining in the lucky image $\mathbf{I}_{\text{lucky}}$. We will come back to this point in the next subsection. But at the high level, we have converted Equation (5.28) into a much simpler optimization problem where we only need to focus on the deconvolution task. Once the deconvolution is completed, the overall restoration problem is solved.

Fourier Burst Accumulation. The method we presented here broadly belongs to the family of block matching. There are alternative approaches in the literature such as Fourier burst accumulation [131, 132]. **Fourier burst accumulation** operates in the frequency domain. If the input images are distorted by turbulence, the blur associated with the turbulence will make the Fourier spectrum of the image narrower (due to the lowpass effects). Thus, given an image stack, we look at the Fourier spectra of the individual frames. For each frequency in the spectrum, we assign a weight so that the processed spectrum is a weighted average of the input spectra. The weights are designed such that the widest spectrum (i.e., those suffering the least due to blur) will be given a higher weight. If the object does not move, a weighted average of the Fourier spectrum may be suitable to remove the blur.

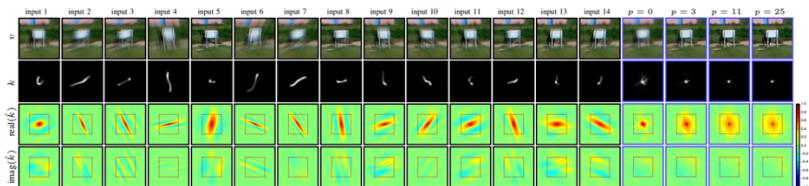


Figure 5.10: Fourier burst photography constructs the weights by evaluating high-frequency components. Source: [131].

5.3 Image Registration

In this Chapter, we discuss another important element in turbulence mitigation: pixel registration. There are two objectives of pixel registration. Firstly, we need to compensate for the jittering effects caused by the turbulence. These effects are typically small but random. Secondly, if the scene contains moving objects, then the registration algorithm needs to compensate for the motion.

5.3.1 Registration for Non-Rigid Deformation

Consider a reference frame $I_{\text{ref}}(\mathbf{x}, t)$ and a frame of interest $I(\mathbf{x}, t)$. The problem of image registration is to find the displacement vector $\Delta\mathbf{x}$ such that after reversing the motion, we can minimize any error between the prediction and the reference:

$$\widehat{\Delta\mathbf{x}} = \underset{\Delta\mathbf{x}}{\operatorname{argmin}} \left(I_{\text{ref}}(\mathbf{x}, t) - I(\mathbf{x} + \Delta\mathbf{x}, t) \right)^2, \quad (5.39)$$

where the ℓ_2 norm is just one of the many possible distance functions we can choose.

If we assume that the motion is simple and small, we can use the first-order approximation to write

$$I(\mathbf{x} + \Delta\mathbf{x}, t) = I(\mathbf{x}, t) + \nabla_{\mathbf{x}} I(\mathbf{x}, t)^T \Delta\mathbf{x},$$

where $\nabla_{\mathbf{x}}$ denotes the spatial gradient of the image. Assuming that our goal is to set $I_{\text{ref}}(\mathbf{x}, t) \approx I(\mathbf{x} + \Delta\mathbf{x}, t)$, we will reach the equation

$$\nabla_{\mathbf{x}} I(\mathbf{x}, t)^T \Delta\mathbf{x} = I_{\text{ref}}(\mathbf{x}, t) - I(\mathbf{x}, t). \quad (5.40)$$

The equation is sometimes called the **optical flow** equation. Since it is a system of linear equations, we can solve it using standard linear algebra methods. One of those is the **Lucas-Kanade** algorithm. Because of the sheer volume of literature on this subject, we shall not repeat the implementation in this book. Generally speaking, optical flow today is very mature. There are plenty of Python / MATLAB packages one can use, including OpenCV [133, 134, 135, 136].

For atmospheric turbulence, since the turbulent effect is significantly less structured than object motion, running optical flow on a per-pixel basis is deemed impossible in the early days. Thus, during the

5.3. IMAGE REGISTRATION

late 00s when optimization algorithms were beginning to become popular, various motion models were proposed to improve the regularity of the problem. Among these models, the **non-rigid image registration** was the most widely adopted model for its robustness and simplicity.

The main argument behind the non-rigid image registration methods is that in a natural image, not all pixels need to be compensated for. For example, if we have a flat region, there is no need to run any optical flow for pixels there even if we know the turbulence distortion is not static. Since there are a limited number of interest points in the image, the image registration algorithm can mainly focus on them. In addition, turbulence is correlated – if one pixel is shifting to the right, then its adjacent pixel should also shift to the right. This allows us to use the concept of **control points** and **spline interpolation** to identify the locations of these carefully chosen control points, as shown in Figure 5.11.

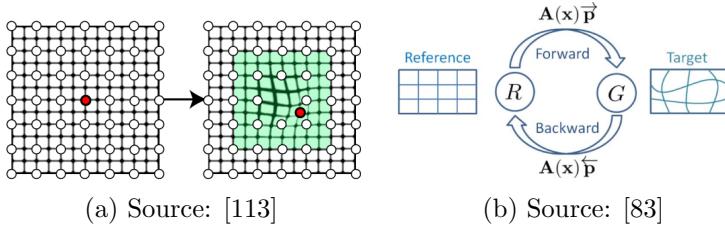


Figure 5.11: Non-rigid motion estimation is commonly used to handle the turbulence image registration problem.

Non-rigid image registration has the advantage that it reduces the number of variables to a small set of control points. This makes the estimation significantly easier. The downside, however, is that the method is largely agnostic to turbulence physics. In fact, turbulence never follows a spline model. Even if some correlations can be approximated by a spline, the hyper-parameters defining the spline model need to be determined too. As algorithms evolve, we see that in the early 2020s, people have gone back to optical flow for a more precise estimation of the motion. The efforts are spent on improving the reference images and using deep neural networks to pull better features.

5.3.2 Reference Frame

For any image registration algorithm to work, we must have a **reference frame** so that the current frame has a reference point when mapping the pixels. For example, if we want to align the $(t + 1)$ -th frame with respect to the t -th frame, we would call the t -th frame a reference frame. In turbulence, the difficulty is that all frames in the image stack are distorted. Thus, aligning the $(t + 1)$ -th distorted frame to the t -th distorted frame does not do anything useful. As such, we need a method to first recover a reasonable reference frame.

There are multiple ways to construct a reference frame. We highlight a few methods here:

Temporal Average. The simplest solution is to construct a temporal average, as done in [137, 83, 138, 139]. The idea is to compute

$$I_{\text{ref}}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T I(\mathbf{x}, t), \quad (5.41)$$

where T is the number of frames we have in the stack. If the scene is static, the temporal average will give us the **long-exposure** image [59]. A long-exposure image is tilt-free because according to Tatarskii the tilts follow a zero-mean Gaussian process [22]. However, as we presented in the turbulence physics Chapter, the long-exposure image is very blurry. Furthermore, if the scene contains a moving object, it is possible that the moving object will be washed out.

Robust PCA. Another approach that can construct a reference frame is the robust principal component analysis (RPCA) technique, where we refer readers to [111] for more detailed discussions. The idea is to consider the observed video $\mathcal{I} = [\mathbf{I}(t_1), \dots, \mathbf{I}(t_T)] \in \mathbb{R}^{N \times T}$ that consists of T frames with each frame containing N pixels. RPCA solves the optimization problem

$$\begin{aligned} & \underset{\mathcal{L}, \mathcal{S}}{\text{minimize}} \quad \|\mathcal{L}\|_* + \lambda \|\mathcal{S}\|_0 \\ & \text{subject to } \mathcal{I} = \mathcal{L} + \mathcal{S}. \end{aligned} \quad (5.42)$$

In this minimization problem, the norm $\|\cdot\|_*$ is the nuclear norm which is the sum of the eigenvalues of the matrix. The norm $\|\cdot\|_0$ is the sparsity norm that measures the number of non-zeros in the matrix. The minimization problem says that the video \mathcal{I} can be decomposed into a **low rank** part \mathcal{L} and a **sparse** part \mathcal{S} . Using our language, we can treat the low rank part as something that does not move, i.e.,

5.3. IMAGE REGISTRATION

the reference frame(s). The moving part is encoded by \mathcal{S} because we assume that the movement is mostly sparse.

RPCA is computationally heavy even though the minimization problem is convex. We need to resort to the most advanced convex optimization solvers to solve the rank minimization. Generally speaking, although many papers have been published for RPCA (beyond the context of turbulence), the actual usage in practice is not common. Besides the computational issues, the solution provided by RPCA is also not guaranteed to be good because minimizing the nuclear norm does not mean that we can extract a good reference image. If the scene contains a moving object, it is likely that RPCA will wash out it too.

Non-local Averaging. One of the more robust approaches is to perform a space-time nonlocal average instead of a simple temporal average [128]. The idea is to construct a reference patch $\mathbf{I}_{\text{ref}}(\mathbf{x}, t)$ via a linear combination of nearby patches:

$$\mathbf{I}_{\text{ref}}(\mathbf{x}, t) = \frac{\sum_{\Delta t \in \Omega_T} w_{\mathbf{x}, t}(\Delta t) \underline{\mathbf{I}}(\mathbf{x}, t + \Delta t)}{\sum_{\Delta t \in \Omega_T} w_{\mathbf{x}, t}(\Delta t)}. \quad (5.43)$$

In this equation, $w_{\mathbf{x}, t}(\Delta t)$ is a weight that will be defined shortly. The weight is a function of the time stamps Δt in a temporal neighborhood $\Omega_T = \{\Delta t_1, \dots, \Delta t_T\}$. Thus, over the time axis, we construct the weighted average using patches $\underline{\mathbf{I}}(\mathbf{x}, t + \Delta t)$ that are determined to be similar to $\underline{\mathbf{I}}(\mathbf{x}, t)$.

Why do we ignore the spatial neighborhood? Isn't $\underline{\mathbf{I}}(\mathbf{x}, t)$ also a distorted patch? Both can be answered by looking at how the weight $w_{\mathbf{x}, t}(\Delta t)$ is defined. We consider a standard ℓ_2 distance between the patch $\underline{\mathbf{I}}(\mathbf{x}, t)$ and $\underline{\mathbf{I}}(\mathbf{x} + \Delta \mathbf{x}, t + \Delta t)$:

$$\delta_{\mathbf{x}, t}(\Delta \mathbf{x}, \Delta t) = \|\underline{\mathbf{I}}(\mathbf{x}, t) - \underline{\mathbf{I}}(\mathbf{x} + \Delta \mathbf{x}, t + \Delta t)\|^2. \quad (5.44)$$

This is nothing but a pairwise comparison between the current patch $\underline{\mathbf{I}}(\mathbf{x}, t)$ and *everyone* in the space-time neighborhood. Among these pairwise distances, we search along the spatial axis and pick the smallest distance, for every Δt :

$$\tilde{\delta}_{\mathbf{x}, t}(\Delta t) = \min_{\Delta \mathbf{x} \in \Omega_x} \delta_{\mathbf{x}, t}(\Delta \mathbf{x}, \Delta t), \quad (5.45)$$

where $\Omega_x = \{\Delta x_1, \dots, \Delta x_K\}$ is the spatial neighborhood. Intuitively, what this equation does is look at each adjacent frame. If there is *one*

CHAPTER 5. IMAGE RESTORATION

patch in the adjacent frame that is similar to $I(\mathbf{x}, t)$, we will consider it a useful candidate for the weighted average. This can be an effective way of differentiating between two situations: (i) If there is object motion, then $\delta_{\mathbf{x},t}(\Delta\mathbf{x}, \Delta t)$ will be large for every Δt because we cannot find a similar patch within a spatial neighborhood as the object moves to another location. So, we can effectively preserve the moving patches by assigning a smaller weight so that the weighted average has less temporal effect. (ii) If there is turbulence, then the patches will jitter but not move away. Thus, for every Δt , it is likely that $\delta_{\mathbf{x},t}(\Delta\mathbf{x}, \Delta t)$ will be small for some $\Delta\mathbf{x}$. If we pick those patches, group them together, and take an average, the turbulence effect will be mitigated. Therefore, $\tilde{\delta}_{\mathbf{x},t}(\Delta t)$ has a unique capability of handling both the object motion and the turbulence.

The weight is defined by taking the exponential function of the distance:

$$w_{\mathbf{x},t}(\Delta t) = \exp\{-\beta\tilde{\delta}_{\mathbf{x},t}(\Delta t)\}, \quad (5.46)$$

where β is a hyper-parameter that can be tuned to adjust the decay rate of the weight. Typically, β is a function of the turbulence strength. If D/r_0 is large, β will be small so that we can use more patches to form the weighted average. In contrast, if D/r_0 is small so that turbulence is weak, β can be large so that we only use the least jittered patches for constructing the weighted average.

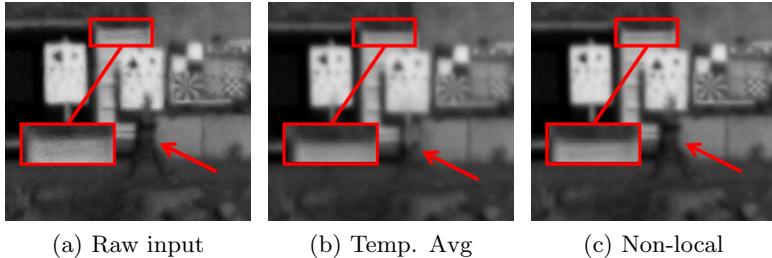


Figure 5.12: Evaluating the reference frame method using real dynamic data. (a) Raw input. (b) Temporal averaging. Observe that the man is blurred over 100 frames. (c) Non-local averaging. Observe that the man remains in the image while the background is stabilized. Source: [128].

A comparison of the reference frame generation method can be seen in Figure 5.12, where we show the results of the temporal aver-

aging method and the non-local method. The scene contains a static background with a moving foreground. As we can see, the temporal averaging washes out the moving foreground whereas the non-local method can preserve it.

5.3.3 Other Image Registration Methods

As far as pre-deep-learning methods are concerned, scenes containing both turbulence and moving objects are very difficult to align. People have considered approaches that extract the region of interest using segmentation methods [140, 141, 139, 142]. The problems with this approach are: (1) the object boundaries are often blurred due to motion. So unless we perform some kind of alpha matting algorithm to extract the transparency level, there is really no good methods to handle the boundaries. Needless to say, alpha matting itself is extremely complex and only works for a certain class of studio-level images. (2) After segmenting the object, most turbulence algorithms will proceed to mitigate the turbulence in the *background*, e.g., [143]. While recovering the background is mathematically more well-posed, it often has little to no practical value because the object of interest is typically in the *foreground*. As a result, segmentation-based methods are seldom used.

In the turbulence literature, a practically useful approach is the block matching concept [144, 145]. The idea is similar to the non-local averaging approach we presented above. Instead of performing a soft decision as in the weighted averaging, we select patches according to some kind of similarity scores and perform frequency domain filtering. The complexity of the method is comparable to the non-local averaging. With graphics processing units (GPUs), these operations can be done in parallel and may be computed reasonably quickly.

5.4 Image Deconvolution

Turbulence mitigation algorithms in the pre-deep-learning era are often implemented in a sequential manner where we first align the images and perform lucky fusion. Then from the lucky image, we perform another step of image deconvolution to remove the residue blur. This type of step-by-step inversion is both a reflection of the physics and a realization of the need for regularizing any optimization problem. In

CHAPTER 5. IMAGE RESTORATION

Figure 5.13 we show a figure taken from [146] that is “representative” of the typical procedures a turbulence mitigation algorithm takes.

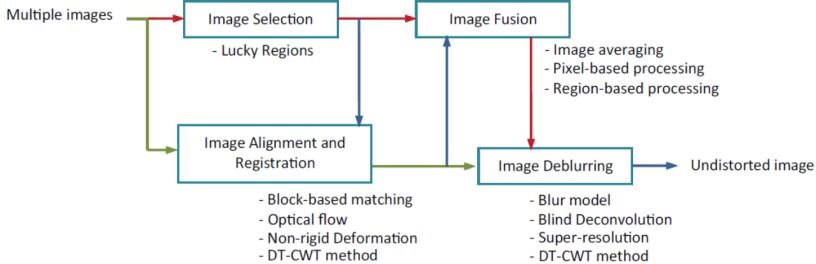


Figure 5.13: Image restoration based on lucky imaging often requires image registration, lucky frame construction, and deblurring. Source: [146].

5.4.1 Tilt-then-Blur in Inverse Problems

As we examine this pipeline, we recognize that the order of mitigating turbulence is often:

$$\hat{\mathbf{J}} = \text{Deblur}(\text{Lucky}(\mathbf{I})) \approx \mathcal{B}^{-1}\mathcal{T}^{-1}(\mathbf{I}),$$

where we recall the tilt \mathcal{T} and \mathcal{B} blur operator. The interesting observation here is that while the true forward model is $\mathcal{B}(\mathcal{T}(\cdot))$, i.e., tilt-then-blur, the inversion does not follow the order of deblurring-then-detilting. Instead, the inversion is detilting-deblurring.

The discrepancy between the forward model and the inverse method needs to be justified from a practical point of view. While deblurring *should* be performed first, it is also *much harder* to perform first. Since the blur in turbulence is spatially varying, deblurring means that we need to estimate all PSFs, one per pixel, in order to recover the image. Without further elaborating on the deblurring algorithms, this PSF estimation alone is enough to convince us that it is nearly an impossible task. Moreover, we need to make sure that the deblurred image (which is often low-passed) still contains the meaningful tilt (which is high-frequency content) for the de-tilting algorithm to work. So, although we want to deblur before de-tilt, practically it is impossible.

Tilt removal (including image registration and lucky fusion) has an immense advantage in that after these two steps, the blur in the lucky

image is more or less spatially invariant. The reason is that during the lucky selection process, we only pick the sharpest patches from the image stack to construct the lucky image. Since we are selecting patches over many frames, we are effectively eliminating bad PSFs and aiming for a delta-function PSF. If all the patches are sharp, then all the corresponding PSFs will be more or less a delta function. So, we have converted the spatially varying blur problem to an invariant blur problem. The merit of the computational efficiency improvement clearly outweighs the mismatch with the forward model. Therefore, although the forward model is tilt-then-blur, the inverse algorithm is often de-tilting and then deblurring.

5.4.2 Why Image Deconvolution?

As the final step of the turbulence mitigation pipeline, image deconvolution aims to boost the image resolution by removing any residue blur in the image. But where does the residue blur come from?

Figure 5.14 is an excerpt from Zhu and Milanfar [83]. In this figure, the authors commented that after the lucky fusion step, the images are not completely sharp because the optical system is limited by diffraction. Therefore, even in the absence of turbulence, the acquired image will still experience some amount of blur due to the Airy disc of a finite aperture. An image deconvolution is thus needed to remove the diffraction-limited blur.

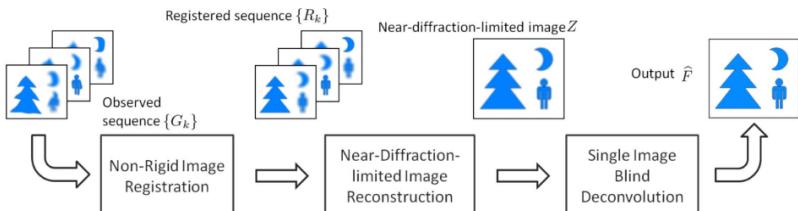


Figure 5.14: A visualization of the deconvolution process after registration. Source: [83].

Readers at this point may ask: if the blur is purely caused by the optical system, then it should be fixed. It seems a fairly straightforward problem to measure the point spread function (PSF) of the optical system and perform a standard deconvolution step using any off-the-shelf algorithm such as Lucy-Richardson. The problem, however,

that although the lucky image is constructed from the sharpest patches from the image stack, there is no guarantee that the sharpest patches are blur-free. Since the digital lucky imaging step is not perfect, there is always some residual blur. These residual blurs are induced by the algorithm and are data-dependent, we cannot say let's measure the PSF and perform Lucy-Richardson. A critical component here is to *estimate* the blur. This is why our deconvolution must be a **blind deconvolution**.

5.4.3 Alternating Minimization

In what follows, we discuss a framework for handling the blur in turbulence mitigation. We emphasize that this framework is not the only way of removing the blur. It is based on the classical alternating minimization, which is relatively easy to understand the intuition behind. We shall focus on *spatially invariant* blur, as we explained at the beginning of this deblurring Chapter.

Simultaneously estimating the image and the blur kernel is known as blind deconvolution [147, 148]. In digital image restoration, the associated inverse problem is often written in terms of matrices and vectors. Denoting $\mathbf{I}_{\text{lucky}} \in \mathbb{R}^N$ as the input blurry image (i.e., the whole lucky fused image instead of a lucky patch), $\mathbf{h} \in \mathbb{R}^K$ as the blur kernel, $\mathbf{J} \in \mathbb{R}^N$ as the latent clean image, the optimization for the blind deconvolution is

$$(\hat{\mathbf{J}}, \hat{\mathbf{h}}) = \underset{\mathbf{J}, \mathbf{h}}{\operatorname{argmin}} \|\mathbf{I}_{\text{lucky}} - \mathbf{h} \circledast \mathbf{J}\|^2 + \lambda g(\mathbf{J}) + \gamma r(\mathbf{h}). \quad (5.47)$$

In this equation, the operator \circledast denotes the spatially invariant convolution. The two functions $g(\mathbf{J})$ and $r(\mathbf{h})$ are the regularization functions that constrain the solution space of the optimization. The two scalars λ and γ are used to control the relative emphasis between the forward model and the regularizations.

Since simultaneously optimizing for \mathbf{J} and \mathbf{h} is a non-convex problem, the standard approach to solve the problem is to consider an alternating minimization strategy by fixing one variable and minimizing over the other variable [149, 150, 151, 152]:

$$\begin{aligned} \mathbf{J}^{k+1} &= \underset{\mathbf{J}}{\operatorname{argmin}} \|\mathbf{I}_{\text{lucky}} - \mathbf{h}^k \circledast \mathbf{J}\|^2 + \lambda g(\mathbf{J}), \\ \mathbf{h}^{k+1} &= \underset{\mathbf{h}}{\operatorname{argmin}} \|\mathbf{I}_{\text{lucky}} - \mathbf{h} \circledast \mathbf{J}^{k+1}\|^2 + \gamma r(\mathbf{h}). \end{aligned}$$

In this pair of equations, the superscript $k = 1, 2, \dots$ denotes the iteration number. As the algorithm makes progress, we expect the estimate of the blur kernel and the image to improve.

The success/failure of the alternating minimization is largely driven by the choice of the regularization functions $g(\mathbf{J})$ and $r(\mathbf{h})$. The regularization function $g(\mathbf{J})$ encapsulates the image prior which is usually learned through data. One relatively mature procedure to solve the \mathbf{J} -subproblem is the **plug-and-play ADMM** (PnP) algorithm [153, 154, 155, 156]. The idea is to split the \mathbf{J} -subproblem into two steps where one step inverts the forward imaging model using the current blur kernel \mathbf{h}^k and the other step uses a pre-trained deep neural network image denoiser to handle the noise. The neural network denoiser provides an implicit modeling of the image prior $g(\mathbf{J})$. We skip the details of the PnP algorithm and refer the readers to monographs such as [157, 158].

For the other regularization $r(\mathbf{h})$, since we are working on turbulence-related blur, we should leverage this piece of information. Generic blur models such as those for motion blur or out-of-focus blur are less relevant although they are popular in the literature. The approach we present here is derived from the phase-to-space (P2S) transform. In some sense, we are re-using the outcomes of the P2S framework for the deblurring purpose.

5.4.4 Modeling the Blur

Recall that when we train the P2S transform, we have constructed a dataset of PSFs. We denote these PSFs as $\mathbf{h}_1, \dots, \mathbf{h}_M$, where M is a large number presenting the number of example PSFs we have simulated. The PSFs are tilt-free, meaning that they are generated from the turbulence simulator with the tilts removed. The PSFs are sent to a principal component analysis (PCA) to extract the principal components. Denote these principal components as $\varphi_1, \dots, \varphi_L$, which are constructed from

$$\{\varphi_\ell\}_{\ell=1}^L = \text{PCA}(\mathbf{h}_1, \dots, \mathbf{h}_M), \quad (5.48)$$

where PCA stands for the principal component analysis.

Assuming that the principal components are determined, the PSFs can then be represented via

$$\mathbf{h} = \sum_{\ell=1}^L w_\ell \varphi_\ell. \quad (5.49)$$

CHAPTER 5. IMAGE RESTORATION

for some representation coefficients $\{w_\ell \mid \ell = 1, \dots, L\}$. Substituting this into the alternating minimization, we obtain a different optimization from an optimization in \mathbf{h}

$$\mathbf{h}^{k+1} = \operatorname{argmin}_{\mathbf{h}} \|\mathbf{I}_{\text{lucky}} - \mathbf{h} \circledast \mathbf{J}^{k+1}\|^2 + \gamma r(\mathbf{h}),$$

to an optimization in $\mathbf{w} = \{w_\ell \mid \ell = 1, \dots, L\}$:

$$\begin{aligned} \mathbf{w}^{k+1} &= \operatorname{argmin}_{\mathbf{w}} \left\| \mathbf{I}_{\text{lucky}} - \left(\sum_{\ell=1}^L w_\ell \boldsymbol{\varphi}_\ell \right) \circledast \mathbf{J}^{k+1} \right\|^2 + \gamma r(\mathbf{w}) \quad (5.50) \\ \mathbf{h}^{k+1} &= \sum_{\ell=1}^L w_\ell^{k+1} \boldsymbol{\varphi}_\ell. \end{aligned}$$

The difference between the first and the second equation is the change of optimization variable from \mathbf{h} to \mathbf{w} . In the first equation, we directly estimate the blur kernel \mathbf{h} without any constraint. In the second equation, we use a parametric model to confine the search space of the blur kernel. All our kernels are generated from the linear combination of the basis functions $\boldsymbol{\varphi}_\ell$. As long as we can specify the basis coefficients w_ℓ , we can construct the blur kernel. Here we have written the PSF as spatially invariant (an approximation that may be justified by the use of lucky fusion beforehand), though works such as Novak et al. [159] have performed similar decomposition and a spatially varying deconvolution or spatially varying pixel-shift estimation as in the case of water distortions [160].

The advantage of the optimization of \mathbf{w} over the optimization of \mathbf{h} is that the regularization function $r(\mathbf{h})$ is difficult to formulate whereas $r(\mathbf{w})$ can be statistically determined. In particular, by analyzing the distribution of the coefficients for a large number of example PSFs, it was found that a reasonable choice of the regularization function is

$$r(\mathbf{w}) = \sum_{\ell=1}^L \frac{|w_\ell|}{\sigma_\ell}, \quad (5.51)$$

where σ_ℓ is a learnable parameter, in this case the standard deviation of w_ℓ learned from data. This $r(\mathbf{w})$ is a mixture of weighted ℓ_1 regularization. With combined with a standard ℓ_2 loss in Equation (5.50), the problem can be solved with convex solvers such as CVX or the ADMM algorithm.

5.4. IMAGE DECONVOLUTION

Empirically, we can plot the distribution of the actual coefficients (by numerically calculating them from the raw PSFs). This will tell us how good Equation (5.51) is and how to estimate the parameters σ_ℓ . As shown in Figure 5.15, we observe a Laplacian type of distribution for the weights w_ℓ using a fixed PCA basis. The parameters σ_ℓ are estimated from this plot to maximize the fit of Equation (5.51) to the data. This also explains why if we naively assume that $r(\mathbf{w}) = \|\mathbf{w}\|_1$, we will not be able to capture the weights σ_ℓ as demonstrated in Figure 5.15.

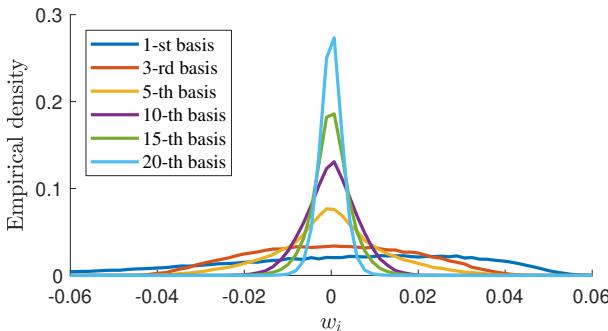


Figure 5.15: Statistical distribution of the basis coefficients w_ℓ . Except for the first few bases, the other basis coefficients demonstrate an exponentially decaying distribution. We can empirically fit this distribution with a parametric distribution. It was found that this is the exponential of the weighted sum of absolute values. Source: [128].

In Figure 5.16 we compared a few state-of-the-art turbulence mitigation methods (in the pre-deep learning era). The methods we study include an optimization-based approach by Lou et al. [110], a classical lucky-imaging-based approach by Zhu and Milanfar [83], a complex wavelet fusion technique CLEAR by Anantrasirichai et al [137], and an improved lucky imaging technique using all the concepts mentioned above [128]. As we can see in these examples, the overall deblurring result of [128] is convincingly better than the competitors. Especially for small objects such as those line patterns in the image, the deblurring algorithm performs well.

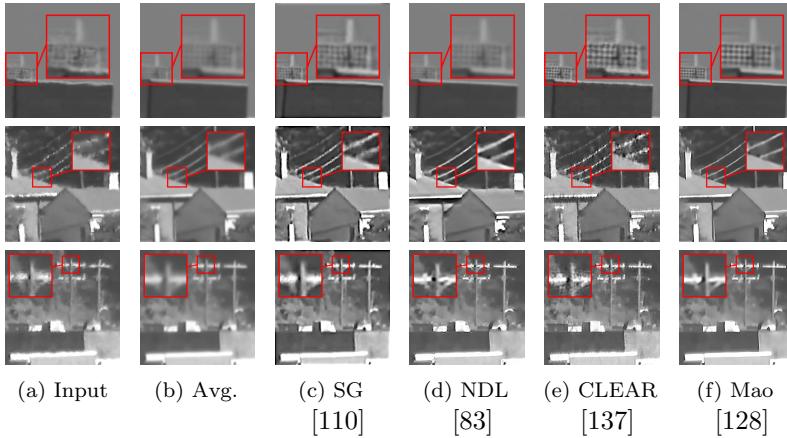


Figure 5.16: Overall comparisons using real long-range static sequences. The first sequence was obtained from Youtube. The rest were captured by Panasonic Full HD Camcorder HC-V180K (aperture diameter 24mm and focal length 103mm), at a distance of 4km and temperature of 30°C.

5.4.5 Beyond Deconvolution

Turbulence mitigation does not always stop at image deconvolution. As early as Shimizu's work [113], it was already mentioned that one can try to super-resolve the image. In a nutshell, we can assume that in addition to the blur and tilt problem, there is also a downsampling operation happening to the image. Thus, when solving the inverse problem, we can attempt to solve

$$\hat{\mathbf{J}} = \underset{\mathbf{J}}{\operatorname{argmin}} \quad \|\mathbf{I} - \mathbf{D}\mathcal{H}_{\theta}(\mathbf{J})\|^2 + \lambda g(\mathbf{J}), \quad (5.52)$$

where \mathbf{D} is a down-sampling operator, basically the identity matrix with alternating rows skipped. Our experience working on super-resolution is that deblurring is already hard enough. To further recover the lost high-frequency content of the image, adding a down-sampling operator is not going to help. In fact, the down-sampling operator makes the inverse problem even more ill-posed. If not regularized properly, it would be very difficult to solve the minimization. But if we regularize too much, then there is a danger of hallucinating the content instead of recovering it.

5.5 Deep Learning Methods

Significant progresses in image restoration algorithms to mitigate atmospheric turbulence are made since the early 2020s. One of the biggest contributing factors is the promise of the deep learning methods demonstrated in a large plethora of restoration tasks such as image/video denoising/deblurring. We will not dive into the development of generic image restoration tasks because it simply evolves too rapidly. Instead, the goal of Chapter 5.5 is to highlight a few relevant concepts in the context of atmospheric turbulence.

5.5.1 The Importance of Data

Deep learning methods, as the name suggests, are a collection of algorithmic (software) solutions that use deep neural networks to *learn* how to restore an image from *data*. The three key components of a deep learning method are

- (i) **Model:** Whether we have a carefully designed neural network to achieve the restoration goals.
- (ii) **Computing resources:** Whether we have a powerful computer with a graphics processing unit (GPU) computer accomplish the training.
- (iii) **Data:** Whether we have enough data to train the model.

Among the three factors, the data is the most important one for a few reasons. Firstly, since the deep neural network is trained based on the data, feeding the model with a poorly designed dataset will lead to poor performance of the network. For example, when handling atmospheric turbulence, if the training dataset contains only one scenario captured on a specific day in a specific place (also at just one temperature and one optical path), then no matter how good the architecture is, it will only learn how to handle that particular imaging condition. When it is presented with a different imaging condition, the network will fail. Secondly, network architectures today are more or less based on similar concepts, e.g., convolutional layers, attention layers, multi-scale, etc. While some configurations are better fit for atmospheric turbulence, their differences are not game changers if the dataset is poorly chosen.

While datasets are important, collecting one is unfortunately extremely challenging. Unlike standard problems such as image deblurring where we can easily synthesize the blur or shake the camera to

CHAPTER 5. IMAGE RESTORATION

collect the blurred image, in atmospheric turbulence we cannot easily “turn off” the turbulence as in the case of camera motion. We can capture the turbulence-distorted image on a hot day, but we will not be able to collect the ground truth on the same day. Some readers may think that if the target pattern is known *a priori*, we can use that as the ground truth. However, since the spectral response of the object to the camera changes from site to site (and day to day), the real ground truth is never known.

Datasets Today (Overview): As of today, data collection for atmospheric turbulence is mainly divided into two categories:

- Characterizing the turbulence, typically done by the defense/-physics community, and
- Evaluating algorithms, typically done by the image processing community.

Dataset	Publication	Approach	videos	range	Public?
NATO	[90], 2007	Real	n/a		✗
OTIS	[161], 2016	Real	16		✓
CLEAR	[137], 2012	Hot air Synthetic	3 8	1km —	✓ ✓
AFRL	[162], 2019	Real	n/a	2.5km	✗
Nature	[163], 2021	Synthetic			✓
Mao	[128], 2020	Real	5	4km	✓
Army	[164], 2021	Real, face			✗
UG2	[165] [166], 2022	Real, text	500	1km	✓
TMT	[167], 2022	Synthetic			✓

Table 5.1: A partial list of atmospheric turbulence datasets reported in the literature. Most of these datasets are developed for testing/evaluating algorithms. Their volumes are usually very small compared to training data used to train deep neural networks.

The focus and scale of the datasets vary significantly as one can see from Table 5.1. For most military efforts, the data usually comes with C_n^2 measurements via a scintillometer. This can be useful for high-energy physics and optical communication through the atmosphere where we need to carefully characterize the received laser beam. However, because of the sensitivity of the data (and the nature of the acquisition process), the data is often classified. As such, for image

5.5. DEEP LEARNING METHODS

processing algorithm developments, the community often collects their own datasets with a larger field of view and a wider range of image content.

Testing Datasets: Small academic research labs typically do not have access to high-end equipment. One workaround solution reported in the literature is to build a hot-air chamber [165]. This idea is similar to the heat tunnel reported in [162] but on a much smaller scale. As shown in Figure 5.17, the heat chamber consists of multiple heat lamps located along the optical path. At the end of the optical path, a monitor is used to display the target pattern, and at the other end of the optical path, a camera with a long-range lens is used to capture the scene. By turning on and off the heat chambers, we can collect the turbulence and the ground truths, respectively. Some limitations of the heat chamber should be mentioned. For example, the heat lamps introduce a strong infrared signal that can alter the spectral profile of the scene. The color is thus distorted. Also, unless the total optical path is sufficiently long, there is no guarantee that one can accurately replicate the true long-range effect. The heat is localized and so the turbulence could be isoplanatic.

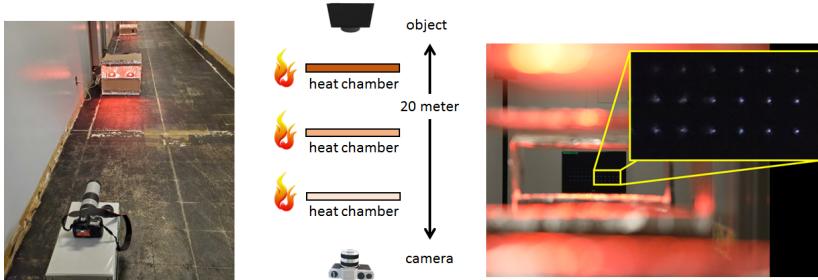


Figure 5.17: A heat chamber is a good optical instrument to record a controllable level of turbulence. Shown here is a setup developed by Mao et al. [165] at Purdue University. It is a small scale, low-cost, and easier version of [162].

For long-range data acquisition, the question one should ask is the purpose of the dataset. If the dataset is used to evaluate restoration algorithms, then the data must have some level of differentiating power to rank the methods. Generic images such as flowers and trees can be difficult because, in the absence of ground truth, typical metrics such as the peak signal-to-noise ratio (PSNR) cannot be used, with

CHAPTER 5. IMAGE RESTORATION

some analysis of metrics offered by Groff et al. [168]. An alternative solution here is to consider a **joint restoration-recognition** task where the restored images will be fed to a downstream object recognition algorithm to report the detected classes. For example, we can ask whether the restoration algorithm has successfully recovered the face of a subject by checking the predicted identity. Figure 5.18 shows an example of **text recognition**. Texts are useful here because it directly tests whether the restoration can recover the desired resolution. We can prepare a list of texts of different fonts, and see to what scale the restoration can perform.



Figure 5.18: Real text data collected by an off-the-shelf camera with an optical zoom. Shown on the left are example distorted images, whereas the subfigures on the right are the actual camera setup.

One ongoing debate in adopting deep learning-based algorithms is how much class specificity we can assume. For example, in CVPR 2022’s UG2+ challenge, winning teams exploited the fact that the evaluation metric is based on optical character recognition. As such, the teams trained the neural networks using text data. While this is a legitimate approach for solving the specific task, it fails to demonstrate the other side of the developed neural networks which is to handle unseen situations. One can argue that class-specific training is valid because the users shall know the type of content and so the best model should be picked. Or, it is possible to pre-train a few models and use one model during inference. The opposite side would then argue that in many military missions, storing a suite of class-specific models can be difficult resource-wise.

Training Dataset: As far as training data is concerned, synthetic data is likely the path to move forward because this is the only way we can generate thousands to millions of examples images to train a deep neural network. As of today, a consistent finding in the literature is that a better turbulence simulator tends to give a better image

5.5. DEEP LEARNING METHODS

restoration result, assuming that the network architecture is fixed. Table 5.2 shows a comparison between two turbulence simulators [41] and [169]. [41] is based on the phase-to-space transform and has a guarantee of the turbulence statistics. [169] assumes a non-rigid deformation with less consideration of turbulence physics. Table 5.2 indicates that when [41] is used to prepare a training dataset which is then used to train a U-Net, the performance is substantially better than that using [169].

D/r_0	[128]	[41]+U-Net	[169]+U-Net
1.5	27.33dB	27.18dB	26.59dB
3.0	27.04dB	26.98dB	26.11dB
4.5	25.85dB	26.01dB	25.40dB

Table 5.2: PSNR values of the reconstruction results, averaged over 30 testing sequences.

At the time of writing this book, the largest training dataset is the TMT dataset generated by Zhang et al. [167], summarized in Table 5.3. The TMT dataset consists of static and dynamic parts. TMT uses the place dataset [170] for synthesizing static scenes. It randomly selects 9,017 images in the original dataset as input for the simulator. TMT generates 50 turbulence images and their associated distortion-free images for every single input, resulting in 9,017 pairs of image sequences of static scenes. TMT splits them into 7,499 pairs and 1,518 pairs for training and testing, respectively.

Table 5.3: Specification of the TMT dataset, where each sequence for the static scene data has 50 frames.

	Static	Dynamic
Source	Place [170]	Sports [171] and TSRWGAN [172]
Amount	9,017 sequences	4,684 videos (1,979,564 frames)
Training	7,499 sequences	3,500 videos
Testing	1,518 sequences	1,184 videos

For dynamic scenes, the TMT dataset contains many videos. The source datasets for our dynamic scene data are the Sports Video in the Wild (SVW) dataset [171] and all ground truth videos used in TSRWGAN [172]. These videos are mixed, generating 4,684 samples

CHAPTER 5. IMAGE RESTORATION

with a total number of frames of 1,979,564. TMT generates 4,684 pairs of full turbulence and distortion-free videos, then randomly split them into 3,500 videos for training and 1,184 for testing, keeping at most 120 frames per testing video.

The significance of the data can be seen from the comparison shown in Figure 5.19 where we compare the TMT dataset and the TSRWGAN dataset. The simulation tool that TSRWGAN [172] used is [173], which generates physically valid tilts and spatially invariant blur kernels, but higher-order aberrations are not simulated. [172] also produces synthetic data by artificial heat sources to create turbulence effects in a short distance. However, this approach tends to generate highly correlated degradation with a weak blur. We observed that their released model does not generalize well on CLEAR’s real-world dataset [137], OTIS [161] and the text dataset [166] where the turbulence effect is stronger than that in the synthetic data used by TSRWGAN. Fine-tuning the TSRWGAN model with the TMT’s data shows a significant improvement of the TSRWGAN model on those out-of-distribution datasets in Fig. 5.19.

5.5.2 Network Designs

The designs of deep neural networks for mitigating atmospheric turbulence can vary significantly from one to the other. In this subsection, we mention a few of the approaches documented in the recent literature.

In the simplest way, we can use an existing network architecture such as a UNet but trained with the synthetic turbulence data, e.g., the BRDNet shown in Figure 5.20. A comprehensive report of the comparisons of several commonly used neural network architectures has been documented by Vint et al [146]. In that report, the authors compared BRDNet [174], RDN [175], CAE-Unet [176], SuperSR [177], DnCNN [154], RCAN [178]. The conclusion of the report was profoundly interesting, that none of these networks are able to produce a substantial improvement in image quality. For several of the testing scenarios, the peak signal-to-noise ratio (PSNR) of the restored image is just marginally better than the raw distorted image. Worse are some cases where the restored image introduces a large number of artifacts that make the image unrecognizable. We should mention that the networks are all trained using synthetic turbulence data containing one million images. However, to fit into the memory constraints, most network sizes are significantly reduced. Whether the shrinkage of the network

5.5. DEEP LEARNING METHODS

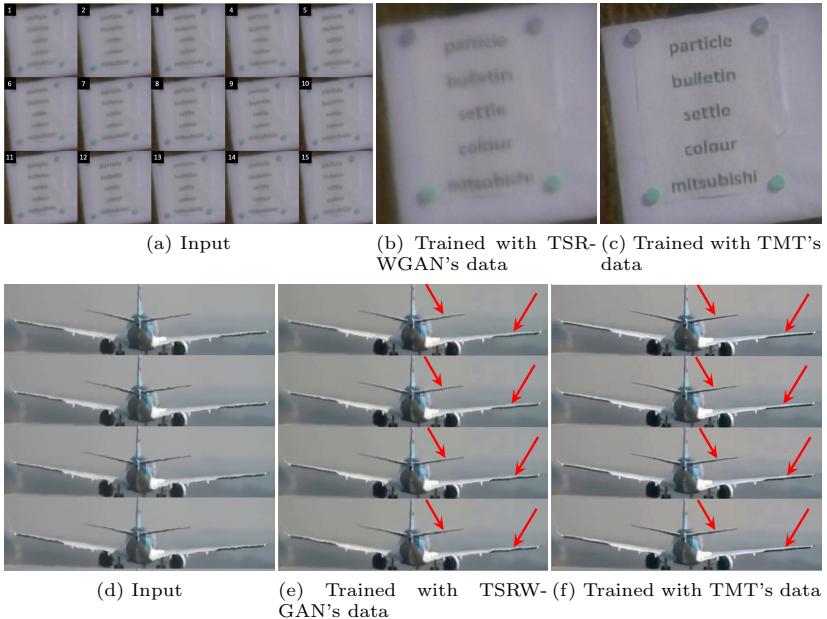


Figure 5.19: Comparing the significance of data.

capacity has an impact to the restoration is unknown.

A natural improvement of the network architecture is to use the network as one module of the overall restoration pipeline, for example, in Figure 5.21. Here, the network is used to replace a traditional image deblurring step which is usually a transform-domain method or an optimization method. Replacing a traditional module with a neural network ensures interpretability because all the other steps are based on classical approaches. It also allows users to debug if anything goes wrong.

The design shown in Figure 5.21 has several limitations. If we only use the deep neural network in the last stage of the pipeline, its functionality is only deblurring. The bigger challenge of aligning frames, picking the lucky patches, and handling the moving objects are all done by traditional methods. The rigidity of the framework does not fully leverage the power of deep learning that can be trained end-to-end. As we see in many of our experiments, end-to-end training is particularly important because it ensures that the intermediate features

CHAPTER 5. IMAGE RESTORATION

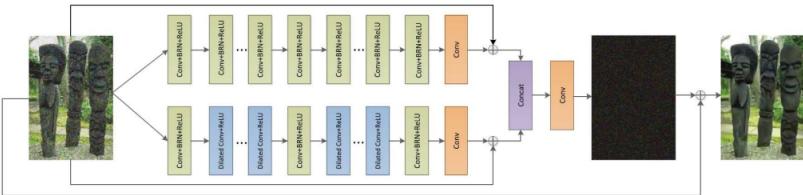


Figure 5.20: Network architecture of BRDNet. This network is one of the older networks that use dilated convolutions and batch normalization. The performance of the network is not great for turbulence mitigation tasks as reported in [146]. Source: [174].

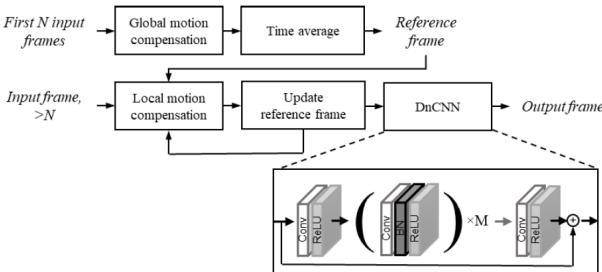


Figure 5.21: Neural networks can be used as a module of the overall turbulence mitigation pipeline. In this work, the idea is to use DnCNN as a deblurring module after performing standard operations such as lucky imaging, registration, and fusion. The performance of such a design is limited by the first half of the pipeline which is a collection of classical techniques. Source: [179].

are relevant to the final output.

If a simple re-use of an existing architecture and a simple plug-in to a classical framework both don't work, then a natural next step is to consider a fully end-to-end network that takes some turbulence physics into consideration. Figure 5.22 shows a complex UNet proposed by Anantrasirichai et al. [180]. The idea is to use an encoder to extract the features of the distorted images, and to use a decoder to reconstruct the clean image from the codewords. In between the encoder-decoder, there is a pyramid residual module to pull and aggregate features at different scales (and in the residue domain.) A complex-valued network is used to handle the random jittering of turbulence. This makes the network

5.5. DEEP LEARNING METHODS

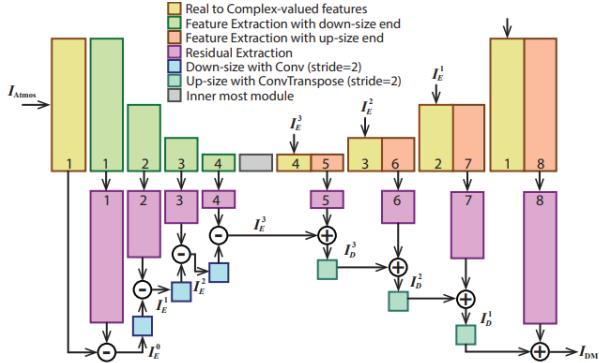


Figure 5.22: Complex valued convolutional neural network proposed by Anantrasirichai [180]. In this network, the idea is to use an encoder-decoder to extract the features and reconstruct the image. Residual connections and pyramid structures are introduced to improve the restoration. Source: [180].



Figure 5.23: Image reconstructions produced by [180]. Source: [180].

reasonably similar to the classical methods such as CLEAR [137] which also performs complex-value computations across different scales. In terms of performance, results shown Figure 5.23 appear to have fewer artifacts than some other networks such as UNet and FFDNet.

5.5.3 Learning Uncertainty

Moving away from a purely black box or a simple plug-in framework, there is an increasing effort to explicitly model the turbulence forward model during the restoration task. As we mentioned before, a turbulence inverse problem can generally be framed as (after flipping the order of tilt and blur for computational efficiency although we know that the correct order is tilt-then-blur):

$$\hat{\mathbf{J}} = \underset{\mathbf{J}}{\operatorname{argmin}} \|\mathbf{I} - \mathcal{T}(\mathcal{B}(\mathbf{J}))\|^2 + \lambda g(\mathbf{J}), \quad (5.53)$$

where \mathcal{B} denotes the blur operator, and \mathcal{T} denotes the tilt operator. The observed noisy image is denoted as \mathbf{I} , and the optimization variable \mathbf{J} is the latent recovered image. The regularization function $g(\mathbf{J})$ is more of a placeholder to emphasize that we put regularizations over the latent image \mathbf{J} .

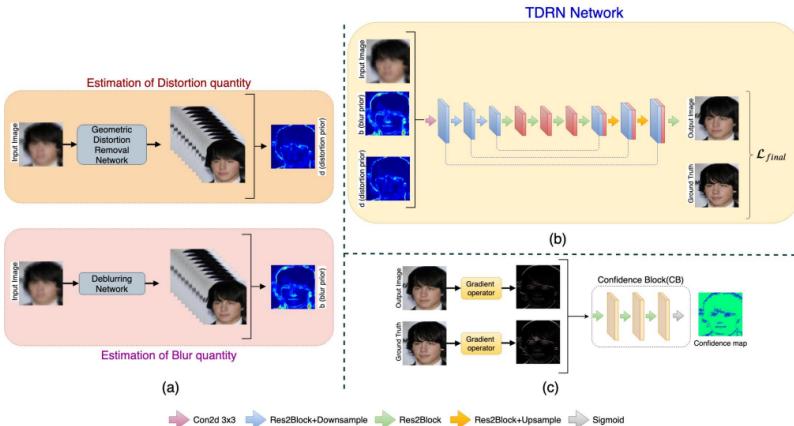


Figure 5.24: Learning the uncertainty map alleviates the difficulty of learning the precise turbulence forward model. In this design, the network separately learns the uncertainty maps of the tilt and the blur. Then, the uncertainty maps are concatenated to the noisy input and processed by the restoration network. Source: [164].

Recovering \mathbf{J} would be a lot easier if we know \mathcal{B} and \mathcal{T} . However, since atmospheric turbulence is random, knowing \mathcal{B} and \mathcal{T} requires us to know the instantaneous realizations of \mathcal{B} and \mathcal{T} , which is not

feasible in practice. The workaround solution here is to estimate the **uncertainty map** instead of the actual turbulence distortion. Proposed by Yasarla et al. [164], the idea is to first generate an indicator map of the tilt and an indicator map of the blur, as shown in Figure 5.24. These indicators are not the true \mathcal{T} and \mathcal{B} , nor even an approximation of them. Instead, we should think of them as *confidence maps* that are concatenated with the input (distorted) image to serve as an additional feature. Here, by confidence, we mean that if a pixel has a high value, then it is likely distorted by tilt (or blur). So the confidence maps tell us *where* and *how strong* the respective distortions are. After the estimation of the confidence maps, another network is used to perform the actual restoration.

The advantage of estimating the uncertainty map is that it bypasses the difficulty of recovering the actual turbulence distortion. However, the physical relationship between the uncertainty map and the turbulence is weak. There is no way we can take the uncertainty map, apply it to the clean image, and obtain the distorted image.

5.5.4 Single-image Restoration by Re-degradation

An alternative approach to the uncertainty map is the concept of **re-degradation** proposed by Mao et al [165]. Instead of constructing the uncertainty map and using it as an additional channel to the restoration neural network, a turbulence degradation block is introduced as a forward process to re-degrade the estimated image. The overall design is shown in Figure 5.25. Starting with the turbulence-distorted image, the method first extracts the features using any backbone feature extractor. In this case, the method uses the vision transformer. The extracted features are then sent to a pair of modules known as the turbulence degradation module and a reconstruction module. In conventional deep neural networks, the turbulence degradation module is absent and so the extracted features are directly sent to a decoder to reconstruct the image. Here, the extracted features are sent to the degradation model to re-generate the distorted image.

The degradation module, in principle, should be a numerical turbulence simulator such as phase-to-space introduced in Chapter 4. However, to use the full phase-to-space transform, it is necessary to make the simulator **differentiable** so that we can back propagate the gradient to update the network parameters. Building a differentiable forward model is an ongoing effort, although some have reported initial

CHAPTER 5. IMAGE RESTORATION

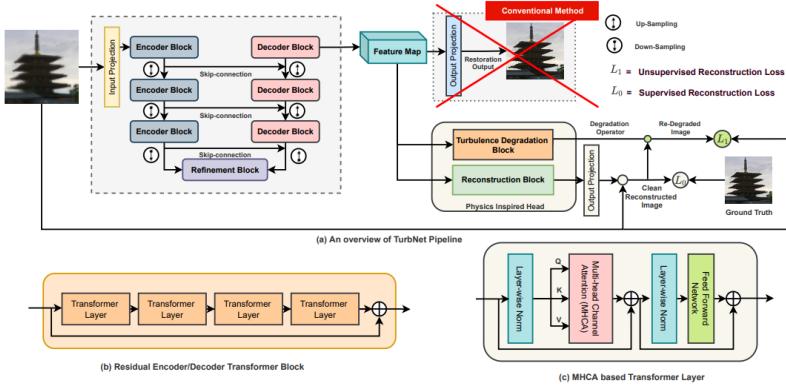


Figure 5.25: Re-degradation and linear approximation model by Mao et al [165]. The core idea here is that after the features are extracted, we re-degrade the features so that the decoded image is the distorted frame. A simulated distorted frame is then compared with the re-degraded image to compute the loss. This allows us to perform self-supervised learning. Source: [165].

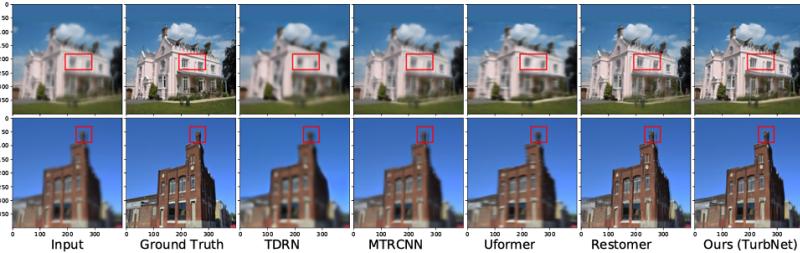


Figure 5.26: Excerpt of the image reconstruction results reported by Mao et al. Source: [165].

success in doing so. In the absence of a full simulator, the degradation module in Figure 5.25 can still be made by considering a *linear* degradation model. As the name suggests, the linear degradation is an approximation to the full simulator. It is a trainable neural network that takes a restored image (or its features) and re-degrades it to produce the distorted image (or its features). This network can be fully convolutional or with some variations. The key is that it applies a set of linear filters to the input image (features) to produce the re-degraded

5.5. DEEP LEARNING METHODS

image. It is a rough (but trained) approximation to the full simulator. On the positive side, the method bypasses the difficulty of recovering the exact turbulence distortion by implicitly using a network to learn how the distorted images are generated. Figure 5.26 shows some image reconstruction results.

The re-degradation concept opens the door to **self-supervised learning**. By re-degrading the restored images, we avoid the difficulty of collecting a large amount of ground truth images which is often not feasible. When we start to train the network, we can use synthetic images to build a pre-trained model. Since synthetic images can be generated in whatever way the simulator allows, we can easily cover a range of turbulence conditions with ground truths. However, synthetic images always have a domain gap with real data. Being able to perform self-supervision makes the model fine-tuning without ground truth achievable. As such, we can use self-supervision to bridge the domain gap, and hence improve generalization.

One important result reported by Mao et al. [165] is an evaluation scheme. In turbulence mitigation, the ground truth is often very difficult to obtain because the appearance of the target pattern changes due to the weather condition. Even if there is no turbulence, the ambient light, the color, etc. will alter the ground truth. Therefore, in some applications, it makes sense to use an alternative downstream recognition task as an evaluation metric. Figure 5.27 reports an attempt to use optical character recognition as a way to assess how well a mitigation algorithm performs. If a mitigation algorithm performs well, we expect the optical character recognition to improve.

5.5.5 Multi-frame End-to-End Transformer

A very recent approach, when this book is written, is a direct end-to-end transformer by Zhang et al. [167] as shown in Figure 5.28 known as the turbulence mitigation transformer (TMT). Building upon the highly successful video restoration transform [181], the TMT presented in [167] brings together two interesting ideas:

Decoupling into tilt and blur. Unlike the previous deep-learning turbulence mitigation methods which are mostly black boxes, TMT explicitly decouples the turbulence into tilt and blur. The method first mitigates the tilts, because tilt correction is a slightly easier task as it is evident from the success of deformable neural networks [182]. The tilt removal sub-network shown in Figure 5.28 uses a **multi-scale**

CHAPTER 5. IMAGE RESTORATION

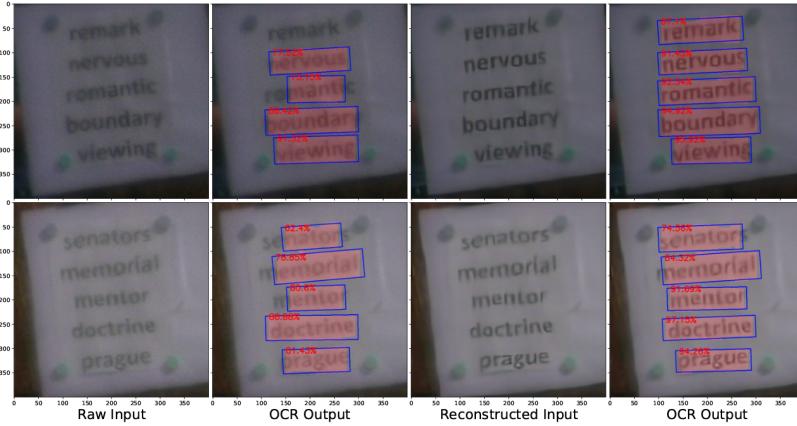


Figure 5.27: Excerpt of the optical character recognition results reported by Mao et al. Source: [165].

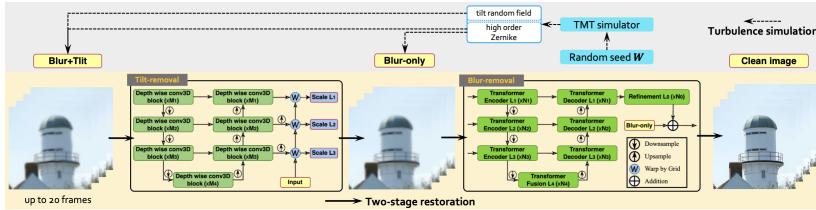


Figure 5.28: Turbulence mitigation transformer by Zhang et al [167].

depthwise convolution to pull the features and align them. The multi-scale structure is critical here because at a lower resolution the tilt is significantly easier to be compensated than at a high resolution. The output of the sub-network is a set of three tilt-corrected images at different resolutions. The ground truth tilt-corrected images are generated by the phase-to-space simulator. In this way, the nature of turbulence is implicitly maintained.

After tilts are removed, the residual blur is still spatially varying. Convolutional neural networks are known to be a poor candidate for spatially varying degradations because the convolution kernels are global operators. Vision transformers use self-attention to extract local attention. Instead of using a global convolutional kernel, the self-attention gives weights to individual kernels so that each pixel would

5.5. DEEP LEARNING METHODS

experience a different combination of convolutions. In other words, transformers allow us to restore the image according to the local distortion strength. This added degree of freedom plays a significant role in the restoration task.

Temporal channel joint attention. The challenge of designing a turbulence mitigation algorithm is how to generate global temporal attention of all the frames without suffering from a high complexity and memory requirement. In conventional transformers, generating attention has a complexity that grows quadratically with the number of pixels in the video. Thus, conventional transformers adopt a window-based strategy to compute the attention locally. In TMT’s transformer, the spatial coordinates are connected purely via convolution layers. This makes the interaction among neighboring pixels smooth, which avoids the inconsistent performance around the margin of local windows. The core module of TMT is temporal-channel self-attention (TCJA). For each pixel, TMT computes the self-attention matrix on the temporal and channel axes. Although the convolution operation is spatially invariant, the combination of the features of each pixel is spatially independent, this enables spatially varying restoration. The dynamics of the blur and residue jitter largely follow a zero-mean random process. Therefore, by enabling full temporal connection over more frames, TMT can be more efficient than conventional transformers in capturing temporal dynamics.



Figure 5.29: Image reconstruction results of a *real* video sequence using the turbulence mitigation transformer by Zhang et al. [167].

The end-to-end transformer developed by Zhang et al. [167] is

CHAPTER 5. IMAGE RESTORATION

known as the turbulence mitigation transformer (TMT). A few snapshots of the reconstruction results of TMT, and their comparisons with other state-of-the-art restoration methods, are shown in Figure 5.29. One important remark here is that the competing methods are strong baselines re-trained with the turbulence data. These baselines are significantly more powerful than the old networks (such as UNets or DnCNN) shown in the previous subsection. That TMT is able to outperform the baselines is an important message that the physics-inspired transformer networks are promising.

5.5.6 Generative Methods

Generative models have taken a significant footprint in the field of image restoration due to their realistic pixel rendering capability. The biggest strength of generative models (in particular the **generative adversarial networks**, GANs) is the way it measures the similarity of two distributions instead of using the conventional mean squared error. GANs typically consist of a generator and a discriminator. The generator is a deep neural network that takes a random code (typically a white Gaussian vector) and turns it into an image. The image will then be compared with a ground truth sample from the training dataset by a discriminator. The discriminator is another deep neural network that gives a binary decision of whether the generated image is close enough to the ground truth. If not, the generator has to be updated until the discriminator can no longer differentiate the generated image and the true image.

The advantage of a GAN is that it is not limited to the particular type of degradation caused by the physics. More interestingly, the resolution of the recovered image is not limited by the actual optical resolution limit but by the prior distribution of the training set. For example, even if one part of the image is extremely blurry, GAN can generate sharp pixels by sampling the closest samples according to the prior distribution it has learned. In some sense, GANs are a powerful tool to hallucinate image content.

In the context of atmospheric turbulence, a recently proposed method TurbuGAN [99] shows how GANs can be implemented. A schematic diagram of TurbuGAN is shown in Figure 5.30. In TurbuGAN, the role of the discriminator is to differentiate between a real turbulence-distorted image and a GAN-generated image. The GAN generator takes a random code, renders a clean image, and sends it

5.5. DEEP LEARNING METHODS

through the turbulence simulator (e.g., phase-to-space transform). The synthesized image is then compared with the real turbulence image by the discriminator.

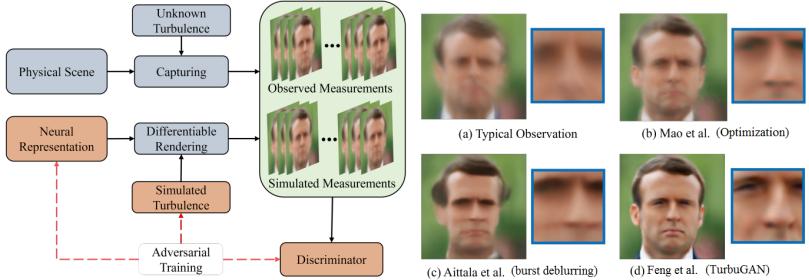


Figure 5.30: Generative adversarial network (GAN) based algorithm by Feng et al. [99].

TurbuGAN is a good demonstration that GAN is capable of handling complex distortions such as atmospheric turbulence. It also demonstrates the importance of a physics-based simulator, for without the simulator it would be impossible to render a realistic turbulence effect. In terms of image recovery, GAN generates sharper images than many traditional methods. This observation is consistent with ATFaceGAN [183], another recently proposed GAN-based image restoration method. Though the nature of training a GAN is based on synthetic data, there are works such as [92] which seek to use a GAN for learning statistics of a scene.

Methods based on the GAN concepts generally suffer from several bottleneck challenges. For example, there is always a controversy about whether the hallucinated pixels are real or fake. For human consumption, the borderline between the real and the fake is less of an issue. However, for machine vision where we need to make a decision, the hallucinated pixels may not contain the true information. Thus, in many cases, there is no guarantee that a GAN-generated image can lead to, for example, a higher face detection performance.

From an implementation perspective, the core part of a GAN-based image restoration method is the generator. The generator can use any standard network architecture, for example, a convolutional network. If the network architecture is fixed, and if the training set is also fixed, then the only difference between a GAN-based network and a non-GAN-based network is the loss function. With complex

degradations such as turbulence, the loss function alone does not make the difference. In fact, the TSRWGAN results shown in Figure 5.29 are based on the training of the generator of TSRWGAN [163]. The image quality is just on par with other networks. Therefore, GAN methods will likely have a long journey of obstacles to go through in the near future.

The **denoising diffusion probabilistic model** is gaining strong momentum in the image restoration community. At a high level, the idea is to train a *diffusion model* that allows us to draw samples from the posterior distribution. The sampling process can be thought of as a sequence of denoising steps where we gradually remove the noise by estimating the key parameters of the signal. Without going into the theoretical details of the method, we highlight a recent result published by Nair et al. [184], as shown in Figure 5.31. As we can see the restored image shows superior object details although some are noticeable artifacts. The promise offered by the model is worth noting.

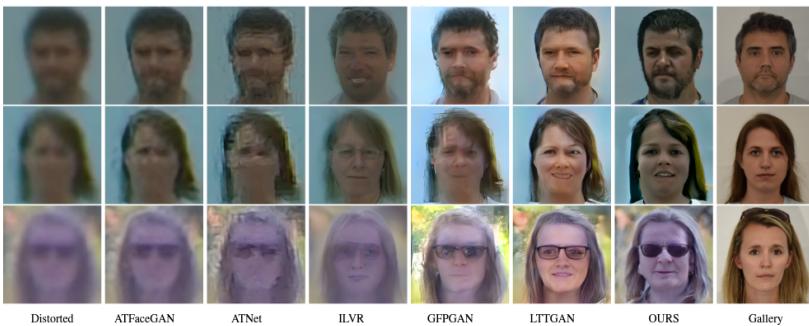


Figure 5.31: Image restoration by denoising diffusion probabilistic model by Nair et al. [184]. Image source: [184].

5.6 Summary

The topic of imaging through turbulence from a reconstruction perspective is presented. The effective application of deep learning to this problem is perhaps the biggest open problem today. While we have witnessed many great innovations over the past few years, we anticipate more results to come. This Chapter may be summarized as follows.

Part 1 Atmospheric Forward Model: We stress the importance of having an accurate and fast-forward model. This is not only for the purpose of simulating realistic training and testing data, but it also gives us the chance to integrate the forward model into the inverse problem. We have seen in this Chapter that data simulated by a more accurate forward model improves significantly the image restoration results, even if we use the exact same neural network architecture.

The problem of tilt-then-blur or blur-then-tilt has been considered. We stress that the correct order is tilt-then-blur, but we also argue that for efficient image restoration, the order can be flipped as long as the benefits to the restoration algorithm are more than the harms.

Part 2 Lucky Imaging and Classical Modeling: Classical methods such as lucky imaging and image registration are still highly insightful. When viewing an image through atmospheric turbulence, one may notice the image going in and out of focus, with the high-quality observations being known as lucky observations. Lucky patches, in the case of atmospheric imaging, imply that a digital lucky fusion must be performed in order to piece together a high-quality image. A good image restoration method should do its best to recover the lucky information. To this end, we argue that multi-frame restorations are significantly more beneficial than single-frame restorations.

Part 3 Deep Learning Approaches: Application of deep learning towards reconstruction and modeling has shown great success in many fields, though for turbulence, there still remains some gap. There are some methods that have produced convincing and promising results, and with the continued development of simulation capabilities, it is reasonable this upward trend will continue. From the perspective of the authors, integration of the true forward model is critical.

Bibliography

- [1] K. Akiyama, A. Alberdi, W. Alef, K. Asada, R. Azulay, A.-K. Baczko, D. Ball, M. Baloković, J. Barrett, D. Bintley, *et al.*, “First m87 event horizon telescope results. iv. imaging the central supermassive black hole,” *The Astrophysical Journal Letters*, vol. 875, no. 1, p. L4, 2019.
- [2] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, “Image and depth from a conventional camera with a coded aperture,” *ACM Transactions on Graphics*, vol. 26, p. 70–es, jul 2007.
- [3] N. Antipa, G. Kuo, R. Heckel, B. Mildenhall, E. Bostan, R. Ng, and L. Waller, “Diffusercam: lensless single-exposure 3d imaging,” *Optica*, vol. 5, pp. 1–9, Jan 2018.
- [4] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, “Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing,” *ACM Transactions on Graphics*, vol. 26, no. 3, p. 69–es, 2007.
- [5] R. Raskar, A. Agrawal, and J. Tumblin, “Coded exposure photography: Motion deblurring using fluttered shutter,” *ACM Transactions on Graphics*, vol. 25, p. 795–804, jul 2006.
- [6] M. Khorasaninejad, W. T. Chen, R. C. Devlin, J. Oh, A. Y. Zhu, and F. Capasso, “Metalenses at visible wavelengths: Diffraction-limited focusing and subwavelength resolution imaging,” *Science*, vol. 352, no. 6290, pp. 1190–1194, 2016.
- [7] M. Born and E. Wolf, *Principles of Optics; Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Cambridge University Press, seventh ed., 1999.

BIBLIOGRAPHY

- [8] J. D. Jackson, *Classical electrodynamics*. New York, NY: Wiley, third ed., 1999.
- [9] R. Ramamoorthi, “Precomputation-based rendering,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 4, pp. 281–369, 2009.
- [10] P. H. Christensen and W. Jarosz, “The path to path-traced movies,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 10, no. 2, pp. 103–175, 2016.
- [11] J. W. Goodman, *Introduction to Fourier Optics*. Roberts and Company, third ed., 2005.
- [12] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*. USA: Prentice-Hall, Inc., 1996.
- [13] E. Hecht, *Optics*. Pearson Education, Inc., fifth ed., 2015.
- [14] R. P. Feynman, *The Feynman lectures on physics*. Reading, Mass. : Addison-Wesley Pub. Co., 1963-1965., 1965 1963.
- [15] N. Chimitt, X. Zhang, Y. Chi, and S. H. Chan, “Scattering and gathering for spatially varying blurs,” 2023. Available online: <https://arxiv.org/abs/2303.05687>.
- [16] A. Kolmogorov, *Foundations of the Theory of Probability*. AMS Chelsea Publishing Series, Chelsea Publishing Company, 1956.
- [17] A. N. Kolmogorov, “The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers,” *Doklady Akademii Nauk SSSR*, vol. 30, pp. 301 – 305, 1941.
- [18] U. Frisch, *Turbulence: The Legacy of A. N. Kolmogorov*. Cambridge University Press, 1995.
- [19] A. M. Obukhov, “On the distribution of energy in the spectrum of turbulent flow,” *Doklady Akademii Nauk SSSR*, vol. 32, pp. 22 – 24, 1941.
- [20] A. M. Obukhov, “Spectral energy distribution in a turbulent flow,” *Izv. Akad. Nauk S.S.S.R, Ser. Geogr. Geofiz.*, vol. 5, pp. 453 – 466, 1941.

BIBLIOGRAPHY

- [21] A. N. Kolmogorov, “Dissipation of energy in the locally isotropic turbulence,” *Doklady Akademii Nauk SSSR*, vol. 32, pp. 16 – 18, 1941.
- [22] V. I. Tatarskii, *Wave Propagation in a Turbulent Medium*. New York: Dover Publications, 1961.
- [23] D. L. Fried, “Statistics of a geometric representation of wavefront distortion,” *Journal of the Optical Society of America*, vol. 55, pp. 1427 – 1435, Nov. 1965.
- [24] D. L. Fried, “Optical resolution through a randomly inhomogeneous medium for very long and very short exposures,” *Journal of the Optical Society of America*, vol. 56, no. 10, pp. 1372 – 1379, 1966.
- [25] D. L. Fried, “Differential angle of arrival: Theory, evaluation, and measurement feasibility,” *Radio Science*, vol. 10, no. 1, pp. 71 – 76, 1975.
- [26] D. L. Fried, “Varieties Of Isoplanatism,” in *Imaging Through the Atmosphere*, Proc. SPIE 0075, 1976.
- [27] D. L. Fried, “Probability of getting a lucky short-exposure image through turbulence,” *Journal of the Optical Society of America*, vol. 68, pp. 1651 – 1658, Dec. 1978.
- [28] A. Ishimaru, *Wave propagation and scattering in random media*, vol. 2. Academic press New York, 1978.
- [29] A. Ishimaru, “Theory and application of wave propagation and scattering in random media,” *Proceedings of the IEEE*, vol. 65, no. 7, pp. 1030–1061, 1977.
- [30] V. Tatarskii, A. Ishimaru, and V. Zavorotny, *Wave Propagation in Random Media (Scintillation): Proceedings of the Conference Held 3-7 August, 1992 at the University of Washington, USA*. SPIE press monographs, Taylor & Francis, 1993.
- [31] D. P. Greenwood, “Bandwidth specification for adaptive optics systems,” *Journal of the Optical Society of America*, vol. 67, pp. 390–393, Mar 1977.

BIBLIOGRAPHY

- [32] R. E. Hufnagel and N. R. Stanley, “Modulation transfer function associated with image transmission through turbulent media,” *Journal of the Optical Society of America*, vol. 54, pp. 52–61, Jan 1964.
- [33] R. E. Hufnagel, “Propagation through atmospheric turbulence,” *The Infrared Handbook*, Chapter 6, 1978.
- [34] M. C. Roggemann and B. M. Welsh, *Imaging through Atmospheric Turbulence*. Taylor & Francis, 1996.
- [35] J. P. Bos, M. C. Roggemann, and V. S. R. Gudimetla, “Anisotropic non-kolmogorov turbulence phase screens with variable orientation,” *Applied Optics*, vol. 54, pp. 2039–2045, Mar. 2015.
- [36] B. M. Welsh, “Fourier-series-based atmospheric phase screen generator for simulating anisoplanatic geometries and temporal evolution,” in *Propagation and Imaging through the Atmosphere*, Proc. SPIE 3125, 1997.
- [37] R. K. Tyson, “Adaptive optics and ground-to-space laser communications,” *Applied Optics*, vol. 35, pp. 3640–3646, Jul 1996.
- [38] R. Tyson, *Principles of Adaptive Optics*. CRC Press, 2010.
- [39] R. J. Noll, “Zernike polynomials and atmospheric turbulence,” *Journal of the Optical Society of America*, vol. 66, pp. 207 – 211, Mar. 1976.
- [40] N. A. Roddier, “Atmospheric wavefront simulation using Zernike polynomials,” *Optical Engineering*, vol. 29, no. 10, pp. 1174 – 1180, 1990.
- [41] N. Chmitt and S. H. Chan, “Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated Zernike coefficients,” *Optical Engineering*, vol. 59, no. 8, p. 083101, 2020.
- [42] Z. Mao, N. Chmitt, and S. H. Chan, “Accelerating atmospheric turbulence simulation via learned phase-to-space transform,” in *IEEE/CVF International Conference on Computer Vision*, pp. 14759 – 14768, Oct. 2021.

BIBLIOGRAPHY

- [43] N. Chmitt, X. Zhang, Z. Mao, and S. H. Chan, “Real-time dense field phase-to-space simulation of imaging through atmospheric turbulence,” *IEEE Transactions on Computational Imaging*, 2022.
- [44] J. W. Goodman, *Statistical Optics*. John Wiley and Sons Inc., second ed., 2015.
- [45] J. D. Schmidt, *Numerical simulation of optical wave propagation: With examples in MATLAB*. SPIE Press, 2010.
- [46] O. Korotkova, *Random Light Beams*. CRC Press, 2017.
- [47] G. C. Valley, “Isoplanatic degradation of tilt correction and short-term imaging systems,” *Applied Optics*, vol. 19, pp. 574–577, Feb 1980.
- [48] R. R. Parenti and R. J. Sasiela, “Laser-guide-star systems for astronomical applications,” *Journal of the Optical Society of America A*, vol. 11, pp. 288 – 309, Jan. 1994.
- [49] J. W. Goodman, *Speckle Phenomena in Optics: Theory and Applications*. Roberts & Company, 2007.
- [50] A. M. Obukhov, “Effect of weak inhomogeneities in the atmosphere on sound and light propagation,” *Izv. Akad. Nauk S.S.R. Ser. Geofiz.*, vol. 2, pp. 155 – 165, 1953.
- [51] D. Fried, “Optical heterodyne detection of an atmospherically distorted signal wave front,” *Proceedings of the IEEE*, vol. 55, no. 1, pp. 57–77, 1967.
- [52] R. P. Feynman and A. R. Hibbs, *Quantum mechanics and path integrals*. International series in pure and applied physics, McGraw-Hill, 1965.
- [53] V. U. Zavorotnyi, V. I. Kliatskin, and V. I. Tatarskii, “Strong fluctuations of electromagnetic-wave intensity in randomly inhomogeneous media,” *Zhurnal Eksperimentalnoi i Teoreticheskoi Fiziki*, vol. 73, pp. 481–497, Aug. 1977.
- [54] V. Tatarskii and V. Zavorotnyi, “Strong fluctuations in light propagation in a randomly inhomogeneous medium,” *Progress in Optics*, vol. 18, pp. 204–256, 1980.

BIBLIOGRAPHY

- [55] M. Charnotskii, J. Gozani, V. Tatarskii, and V. Zavorotny, “Wave propagation theories in random media based on the path-integral approach,” *Progress in Optics*, vol. 32, pp. 203–266, 1993.
- [56] R. Dashen, “Path integrals for waves in random media,” *Journal of Mathematical Physics*, vol. 20, pp. 894–920, May 1979.
- [57] G. Heidbreder, “Image degradation with random wavefront tilt compensation,” *IEEE Transactions on Antennas and Propagation*, vol. 15, no. 1, pp. 90 – 98, 1967.
- [58] D. Voelz, *Computational Fourier Optics: A MATLAB Tutorial*. SPIE Press, 2011.
- [59] R. C. Hardie, J. D. Power, D. A. LeMaster, D. R. Droege, S. Gladysz, and S. Bose-Pillai, “Simulation of anisoplanatic imaging through optical turbulence using numerical wave propagation with new validation analysis,” *Optical Engineering*, vol. 56, no. 7, p. 071502, 2017.
- [60] J. P. Bos and M. C. Roggemann, “Technique for simulating anisoplanatic image formation over long horizontal paths,” *Optical Engineering*, vol. 51, no. 10, p. 101704, 2012.
- [61] M. C. Roggemann, B. M. Welsh, D. Montera, and T. A. Rhoadarmer, “Method for simulating atmospheric turbulence phase effects for multiple time slices and anisoplanatic conditions,” *Applied Optics*, vol. 34, pp. 4037 – 4051, Jul. 1995.
- [62] F. Wang, I. Toselli, and O. Korotkova, “Two spatial light modulator system for laboratory simulation of random beam propagation in random media,” *Applied Optics*, vol. 55, pp. 1112 – 1117, Feb. 2016.
- [63] B. Rodenburg, M. Mirhosseini, M. Malik, O. S. Magaña-Loaiza, M. Yanakas, L. Maher, N. K. Steinhoff, G. A. Tyler, and R. W. Boyd, “Simulating thick atmospheric turbulence in the lab with application to orbital angular momentum communication,” *New Journal of Physics*, vol. 16, p. 033020, Mar. 2014.
- [64] J. D. Phillips, M. E. Goda, and J. Schmidt, “Atmospheric turbulence simulation using liquid crystal spatial light modulators,”

BIBLIOGRAPHY

- in *Advanced Wavefront Control: Methods, Devices, and Applications III*, vol. 5894, p. 589406, International Society for Optics and Photonics, SPIE, 2005.
- [65] M. A. Vorontsov and G. W. Carhart, “Anisoplanatic imaging through turbulent media: image recovery by local information fusion from a set of short-exposure images,” *Journal of the Optical Society of America A*, vol. 18, pp. 1312 – 1324, Jun. 2001.
 - [66] Y. Tian, S. G. Narasimhan, and A. J. Vannevel, “Depth from optical turbulence,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246–253, 2012.
 - [67] C. J. Pellizzari, M. F. Spencer, and C. A. Bouman, “Imaging through distributed-volume aberrations using single-shot digital holography,” *Journal of the Optical Society of America A*, vol. 36, pp. A20–A33, Feb. 2019.
 - [68] C. J. Pellizzari, M. T. Banet, M. F. Spencer, and C. A. Bouman, “Demonstration of single-shot digital holography using a bayesian framework,” *Journal of the Optical Society of America A*, vol. 35, pp. 103–107, Jan. 2018.
 - [69] B. L. McGlamery, “Restoration of turbulence-degraded images,” *Journal of the Optical Society of America*, vol. 57, pp. 293 – 297, Mar. 1967.
 - [70] C. Macaskill and T. E. Ewart, “Computer Simulation of Two-dimensional Random Wave Propagation,” *IMA Journal of Applied Mathematics*, vol. 33, pp. 1 – 15, Jul. 1984.
 - [71] M. Spivack and B. Uscinski, “The split-step solution in random wave propagation,” *Journal of Computational and Applied Mathematics*, vol. 27, no. 3, pp. 349 – 361, 1989.
 - [72] J. M. Martin and S. M. Flatté, “Intensity images and statistics from numerical simulation of wave propagation in 3-d random media,” *Applied Optics*, vol. 27, pp. 2111 – 2126, Jun. 1988.
 - [73] W. A. Coles, J. P. Filice, R. G. Frehlich, and M. Yadlowsky, “Simulation of wave propagation in three-dimensional random media,” *Applied Optics*, vol. 34, pp. 2089 – 2101, Apr. 1995.

BIBLIOGRAPHY

- [74] R. Frehlich, “Simulation of laser propagation in a turbulent atmosphere,” *Applied Optics*, vol. 39, pp. 393 – 397, Jan. 2000.
- [75] W. P. Burckel and R. N. Gray, “Turbulence phase screens based on polar-logarithmic spectral sampling,” *Applied Optics*, vol. 52, pp. 4672 – 4680, Jul. 2013.
- [76] B. J. Herman and L. A. Strugala, “Method for inclusion of low-frequency contributions in numerical representation of atmospheric turbulence,” in *Propagation of High-Energy Laser Beams Through the Earth’s Atmosphere*, Proc. SPIE 1221, 1990.
- [77] D. L. Fried and T. Clark, “Extruding kolmogorov-type phase screen ribbons,” *Journal of the Optical Society of America A*, vol. 25, pp. 463 – 468, Feb. 2008.
- [78] R. G. Lane, A. Glindemann, and J. C. Dainty, “Simulation of a Kolmogorov phase screen,” *Waves in Random Media*, vol. 2, no. 3, pp. 209 – 224, 1992.
- [79] E. Anzuola and S. Gladysz, “Modeling dynamic atmospheric turbulence using temporal spectra and Karhunen–Loève decomposition,” *Optical Engineering*, vol. 56, no. 7, p. 071508, 2017.
- [80] R. C. Hardie, M. A. Rucci, S. R. Bose-Pillai, R. V. Hook, and B. K. Karch, “Modeling and simulation of multispectral imaging through anisoplanatic atmospheric optical turbulence,” *Optical Engineering*, vol. 61, no. 9, p. 093102, 2022.
- [81] W. H. Chak, C. P. Lau, and L. M. Lui, “Subsampled turbulence removal network,” *Mathematics, Computation and Geometry of Data*, vol. 1, pp. 1 – 33, 2021.
- [82] C. P. Lau, Y. H. Lai, and L. M. Lui, “Restoration of atmospheric turbulence-distorted images via RPCA and quasiconformal maps,” *Inverse Problems*, Mar. 2019.
- [83] X. Zhu and P. Milanfar, “Removing atmospheric turbulence via space-invariant deconvolution,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 157 – 170, Jan. 2013.
- [84] S. H. Chan, “Tilt-then-blur or blur-then-tilt? clarifying the atmospheric turbulence model,” *IEEE Signal Processing Letters*, vol. 29, pp. 1833–1837, 2022.

BIBLIOGRAPHY

- [85] E. Repasi and R. Weiss, “Computer simulation of image degradations by atmospheric turbulence for horizontal views,” in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXII*, Proc. SPIE 8014, 2011.
- [86] E. Repasi and R. Weiss, “Analysis of image distortions by atmospheric turbulence and computer simulation of turbulence effects,” in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XIX*, Proc. SPIE 6941, 2008.
- [87] K. R. Leonard, J. Howe, and D. E. Oxford, “Simulation of atmospheric turbulence effects and mitigation algorithms on stand-off automatic facial recognition,” in *Optics and Photonics for Counterterrorism, Crime Fighting, and Defence VIII*, Proc. SPIE 8546, Oct. 2012.
- [88] G. Potvin, J. L. Forand, and D. Dion, “A simple physical model for simulating turbulent imaging,” in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXII* (G. C. Holst and K. A. Krapels, eds.), vol. 8014, p. 80140Y, International Society for Optics and Photonics, SPIE, 2011.
- [89] D. H. Tofsted, D. Quintis, S. G. O’Brien, J. Yarbrough, M. D. Bustillos, and G. M. T. Vaucher, “Test report on the november 2005 nato rtg-40 active imager land field trials,” tech. rep., Army Research Lab, 2006.
- [90] D. Tofsted, S. O’Brien, J. Yarbrough, D. Quintis, and M. Bustillos, “Characterization of atmospheric turbulence during the NATO RTG-40 land field trials,” in *Atmospheric Propagation IV*, Proc. SPIE 6551, 2007.
- [91] K. J. Miller, B. Preece, T. W. D. Bosq, and K. R. Leonard, “A data-constrained algorithm for the emulation of long-range turbulence-degraded video,” in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXX* (G. C. Holst and K. A. Krapels, eds.), vol. 11001, p. 110010J, International Society for Optics and Photonics, SPIE, 2019.
- [92] K. J. Miller and T. D. Bosq, “A machine learning approach to improving quality of atmospheric turbulence simulation,” in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXXII*, p. 117400N, Proc. SPIE 11740, 2021.

BIBLIOGRAPHY

- [93] M. A. Vorontsov and V. Kolosov, “Target-in-the-loop beam control: basic considerations for analysis and wave-front sensing,” *Journal of the Optical Society of America A*, vol. 22, pp. 126 – 141, Jan. 2005.
- [94] S. L. Lachinova, M. A. Vorontsov, V. V. Dudorov, V. V. Kolosov, and M. T. Valley, “Anisoplanatic imaging through atmospheric turbulence: brightness function approach,” in *Atmospheric Optics: Models, Measurements, and Target-in-the-Loop Propagation*, Proc. SPIE 6708, 2007.
- [95] S. L. Lachinova, M. A. Vorontsov, G. A. Filimonov, D. A. LeMaster, and M. E. Trippel, “Comparative analysis of numerical simulation techniques for incoherent imaging of extended objects through atmospheric turbulence,” *Optical Engineering*, vol. 56, no. 7, 2017.
- [96] D. G. Voelz, E. Wijerathna, A. Muschinski, and X. Xiao, “Computer simulations of optical turbulence in the weak- and strong-scattering regime: angle-of-arrival fluctuations obtained from ray optics and wave optics,” *Optical Engineering*, vol. 57, no. 10, p. 104102, 2018.
- [97] J. R. Peterson, J. G. Jernigan, S. M. Kahn, A. P. Rasmussen, E. Peng, Z. Ahmad, J. Bankert, C. Chang, C. Claver, D. K. Gilmore, E. Grace, M. Hannel, M. Hodge, S. Lorenz, A. Lupu, A. Meert, S. Nagarajan, N. Todd, A. Winans, and M. Young, “Simulation of astronomical images from optical survey telescopes using a comprehensive photon monte carlo approach,” *The Astrophysical Journal Supplement Series*, vol. 218, p. 14, May 2015.
- [98] B. R. Hunt, A. L. Iler, C. A. Bailey, and M. A. Rucci, “Synthesis of atmospheric turbulence point spread functions by sparse and redundant representations,” *Optical Engineering*, vol. 57, no. 2, p. 024101, 2018.
- [99] B. Y. Feng, M. Xie, and C. A. Metzler, “TurbuGAN: An adversarial learning approach to spatially-varying multiframe blind deconvolution with applications to imaging through turbulence,” 2022. Available online: <https://arxiv.org/pdf/2203.06764.pdf>. Accessed Jun. 30, 2022.

BIBLIOGRAPHY

- [100] M. A. Rucci, R. C. Hardie, and R. K. Martin, “Simulation of anisoplanatic lucky look imaging and statistics through optical turbulence using numerical wave propagation,” *Applied Optics*, vol. 60, pp. G19–G29, Sep 2021.
- [101] N. Chmitt and S. H. Chan, “Anisoplanatic optical turbulence simulation for near-continuous c_n^2 profiles without wave propagation,” 2023. Available online: <https://arxiv.org/abs/2305.09036>.
- [102] J. Y. Wang and J. K. Markey, “Modal compensation of atmospheric turbulence phase distortion,” *Journal of the Optical Society of America*, vol. 68, pp. 78–87, Jan 1978.
- [103] R. Gray, “Zernikecalc: A MATLAB function to work with zernike polynomials over circular and non-circular pupils.” Available online: <https://www.mathworks.com/matlabcentral/fileexchange/33330-zernikecalc>.
- [104] S. Basu, J. E. McCrae, and S. T. Fiorino, “Estimation of the path averaged atmospheric refractive index structure constant from time lapse imagery,” in *Laser Radar Technology and Applications XX; and Atmospheric Propagation XII*, p. 94650T, Proc. SPIE 9465, May 2015.
- [105] G. A. Chanan, “Calculation of wave-front tilt correlations associated with atmospheric turbulence,” *Journal of the Optical Society of America A*, vol. 9, pp. 298 – 301, Feb. 1992.
- [106] N. Takato and I. Yamaguchi, “Spatial correlation of Zernike phase-expansion coefficients for atmospheric turbulence with finite outer scale,” *Journal of the Optical Society of America A*, vol. 12, pp. 958 – 963, May 1995.
- [107] M. R. Whiteley, M. C. Roggemann, and B. M. Welsh, “Temporal properties of the zernike expansion coefficients of turbulence-induced phase aberrations for aperture and source motion,” *Journal of the Optical Society of America A*, vol. 15, pp. 993–1005, Apr 1998.
- [108] A. J. E. M. Janssen, “Extended nijboer–zernike approach for the computation of optical point-spread functions,” *Journal of the Optical Society of America A*, vol. 19, pp. 849–857, May 2002.

BIBLIOGRAPHY

- [109] S. van Haver, *The Extended Nijboer-Zernike Diffraction Theory and its applications*. PhD thesis, Delft University of Technology, 2010.
- [110] Y. Lou, S. Ha Kang, S. Soatto, and A. Bertozzi, “Video stabilization of atmospheric turbulence distortion,” *Inverse Problems and Imaging*, vol. 7, pp. 839 – 861, Aug. 2013.
- [111] C. P. Lau, Y. H. Lai, and L. M. Lui, “Restoration of atmospheric turbulence-distorted images via RPCA and quasiconformal maps,” *Inverse Problems*, Mar. 2019.
- [112] Y. Mao and J. Gilles, “Non rigid geometric distortions correction - application to atmospheric turbulence stabilization,” *Inverse Problems and Imaging*, vol. 3, pp. 531 – 546, 2012.
- [113] M. Shimizu, S. Yoshimura, M. Tanaka, and M. Okutomi, “Super-resolution from image sequence under influence of hot-air optical turbulence,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1 – 8, 2008.
- [114] A. S. S. Gepshtain and B. Fishbain, “Restoration of atmospheric turbulent video containing real motion using rank filtering and elastic image registration,” in *European Signal Processing Conference*, pp. 477 – 480, Sep. 2004.
- [115] D. H. Frakes, J. W. Monaco, and M. J. T. Smith, “Suppression of atmospheric turbulence in video using an adaptive control grid interpolation approach,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1881 – 1884, 2001.
- [116] Y. Xie, W. Zhang, D. Tao, W. Hu, Y. Qu, and H. Wang, “Removing turbulence effect via hybrid total variation and deformation-guided kernel regression,” *IEEE Transactions on Image Processing*, vol. 25, pp. 4943 – 4958, Oct. 2016.
- [117] R. He, Z. Wang, Y. Fan, and D. Feng, “Atmospheric turbulence mitigation based on turbulence extraction,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1442 – 1446, Mar. 2016.
- [118] T. Çaliskan and N. Arica, “Atmospheric turbulence mitigation using optical flow,” in *International Conference on Pattern Recognition*, pp. 883 – 888, 2014.

BIBLIOGRAPHY

- [119] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling, “Efficient filter flow for space-variant multiframe blind deconvolution,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 607 – 614, 2010.
- [120] C. J. Pellizzari, R. Trahan, H. Zhou, S. Williams, S. E. Williams, B. Nemati, M. Shao, and C. A. Bouman, “Optically coherent image formation and denoising using a plug and play inversion framework,” *Applied Optics*, vol. 56, pp. 4735–4744, Jun. 2017.
- [121] D. Fried, “Anisoplanatism in adaptive optics,” *Journal of the Optical Society of America*, vol. 72, no. 1, pp. 52 – 61, 1982.
- [122] F. Roddier, *Adaptive Optics in Astronomy*. Cambridge University Press, 1999.
- [123] M. A. Ealey and J. A. Wellman, “Deformable mirrors: design fundamentals, key performance specifications, and parametric trades,” in *Active and Adaptive Optical Components*, Proc. SPIE 1543, 1992.
- [124] A. T. Watnik and D. F. Gardner, “Wavefront sensing in deep turbulence,” *Optics and Photonics News*, vol. 29, pp. 38–45, Oct. 2018.
- [125] M. C. Roggemann, C. A. Stoudt, and B. M. Welsh, “Image-spectrum signal-to-noise-ratio improvements by statistical frame selection for adaptive-optics imaging through atmospheric turbulence,” *Optical Engineering*, vol. 33, no. 10, pp. 3254 – 3264, 1994.
- [126] M. Aubailly, M. A. Vorontsov, G. W. Carhart, and M. T. Valley, “Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach,” in *Atmospheric Optics: Models, Measurements, and Target-in-the-Loop Propagation III*, Proc. SPIE 7463, Aug. 2009.
- [127] N. Joshi and M. F. Cohen, “Seeing Mt. Rainier: Lucky imaging for multi-image denoising, sharpening, and haze removal,” in *IEEE International Conference on Computational Photography*, pp. 1–8, 2010.
- [128] Z. Mao, N. Chimitt, and S. H. Chan, “Image reconstruction of static and dynamic scenes through anisoplanatic turbulence,”

BIBLIOGRAPHY

- IEEE Transactions on Computational Imaging*, vol. 6, pp. 1415 – 1428, Oct. 2020.
- [129] A. Buades, B. Coll, and J. . Morel, “A non-local algorithm for image denoising,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 60 – 65, 2005.
 - [130] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3D transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, pp. 2080 – 2095, Aug. 2007.
 - [131] M. Delbracio and G. Sapiro, “Removing camera shake via weighted fourier burst accumulation,” *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3293 – 3307, 2015.
 - [132] G. Gilles and S. Osher, “Wavelet burst accumulation for turbulence mitigation,” *Journal of Electronic Imaging*, vol. 25, no. 3, p. 033003, 2016.
 - [133] A. Ranjan and M. J. Black, “Optical flow estimation using a spatial pyramid network,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4161 – 4170, 2017.
 - [134] C. Liu, *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Massachusetts Institute of Technology, Jan. 2009.
 - [135] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2462 – 2470, 2017.
 - [136] T.-W. Hui, X. Tang, and C. C. Loy, “LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8981 – 8989, 2018.
 - [137] N. Anantrasirichai, A. Achim, N. G. Kingsbury, and D. R. Bull, “Atmospheric turbulence mitigation using complex wavelet-based fusion,” *IEEE Transactions on Image Processing*, vol. 22, pp. 2398 – 2408, Jun. 2013.

BIBLIOGRAPHY

- [138] N. Anantrasirichai, A. Achim, and D. Bull, “Atmospheric turbulence mitigation for sequences with moving objects using recursive image fusion,” in *IEEE International Conference on Image Processing*, pp. 2895 – 2899, Oct. 2018.
- [139] R. Nieuwenhuizen, J. Dijk, and K. Schutte, “Dynamic turbulence mitigation for long-range imaging in the presence of large moving objects,” *EURASIP Journal on Image and Video Processing*, vol. 2, no. 2, 2019.
- [140] C. S. Huebner, “Turbulence mitigation of short exposure image data using motion detection and background segmentation,” in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXIII*, Proc. SPIE 8355, May 2012.
- [141] O. Oreifej, X. Li, and M. Shah, “Simultaneous video stabilization and moving object detection in turbulence,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 450 – 462, Feb. 2013.
- [142] K. K. Halder, M. Tahtali, and S. G. Anavatti, “Geometric correction of atmospheric turbulence-degraded video containing moving objects,” *Optics Express*, vol. 23, pp. 5091 – 5101, Feb. 2015.
- [143] B. Xue, Y. Liu, L. Cui, X. Bai, X. Cao, and F. Zhou, “Video stabilization in atmosphere turbulent conditions based on the Laplacian-Riesz pyramid,” *Optics Express*, vol. 24, pp. 28092 – 28103, Nov. 2016.
- [144] D. R. Droege, R. C. Hardie, B. S. Allen, A. J. Dapone, and J. C. Blevins, “A real-time atmospheric turbulence mitigation and super-resolution solution for infrared imaging systems,” in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXIII*, Proc. SPIE 8355, 2012.
- [145] R. C. Hardie, M. A. Rucci, A. J. Dapone, and B. K. Karch, “Block matching and wiener filtering approach to optical turbulence mitigation and its application to simulated and real imagery with quantitative error analysis,” *Optical Engineering*, vol. 56, no. 7, p. 071503, 2017.
- [146] D. Vint, G. D. Caterina, J. Soraghan, R. Lamb, and D. Humphreys, “Analysis of deep learning architectures for tur-

BIBLIOGRAPHY

- bulence mitigation in long-range imagery,” in *Artificial Intelligence and Machine Learning in Defense Applications II*, Proc. SPIE 11543, 2020.
- [147] A. Levin, “Blind motion deblurring using image statistics,” *Advances in Neural Information Processing Systems*, vol. 19, 2006.
 - [148] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman, “Removing camera shake from a single photograph,” *ACM Transactions on Graphics*, vol. 25, pp. 787 – 794, Jul. 2006.
 - [149] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, “Understanding blind deconvolution algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2354 – 2367, 2011.
 - [150] T. Michaeli and M. Irani, “Blind deblurring using internal patch recurrence,” in *European Conference on Computer Vision*, pp. 783 – 798, Springer, 2014.
 - [151] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, “Blind motion deblurring from a single image using sparse approximation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 104 – 111, 2009.
 - [152] D. Krishnan, T. Tay, and R. Fergus, “Blind deconvolution using a normalized sparsity measure,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 233 – 240, IEEE, 2011.
 - [153] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, “Plug-and-play priors for model based reconstruction,” in *IEEE Global Conference on Signal and Information Processing*, pp. 945 – 948, IEEE, 2013.
 - [154] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142 – 3155, 2017.
 - [155] S. H. Chan, X. Wang, and O. A. Elgendy, “Plug-and-play ADMM for image restoration: Fixed-point convergence and applications,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84 – 98, 2017.

BIBLIOGRAPHY

- [156] Y. Romano, M. Elad, and P. Milanfar, “The little engine that could: Regularization by denoising (RED),” *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804 – 1844, 2017.
- [157] R. Ahmad, C. A. Bouman, G. T. Buzzard, S. Chan, S. Liu, E. T. Reehorst, and P. Schniter, “Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery,” *IEEE Signal Processing Magazine*, vol. 37, no. 1, pp. 105 – 116, 2020.
- [158] S. H. Chan, “Performance analysis of plug-and-play admm: A graph signal processing perspective,” *IEEE Transactions on Computational Imaging*, vol. 5, no. 2, pp. 274 – 286, 2019.
- [159] K. Novak and A. T. Watnik, “Imaging through deconvolution with a spatially variant point spread function,” in *Computational Imaging VI*, p. 1173105, Proc. SPIE 11731, 2021.
- [160] Y. Tian and S. G. Narasimhan, “Seeing through water: Image restoration using model-based tracking,” in *IEEE International Conference on Computer Vision*, pp. 2303–2310, 2009.
- [161] J. Gilles and N. B. Ferrante, “Open turbulent image set (OTIS),” *Pattern Recognition Letters*, vol. 86, pp. 38 – 41, 2017.
- [162] M. T. Velluet, C. Bell, J. F. Daigle, J. Dijk, S. Gladysz, A. Kanaev, A. Lambert, D. Lemaster, G. Potvin, and M. Vorontsov, “Data collection and preliminary results on turbulence characterisation and mitigation techniques,” in *Electro-Optical and Infrared Systems: Technology and Applications XVI*, Proc. SPIE 11159, 2019.
- [163] D. Jin, Y. Chen, Y. Lu, J. Chen, P. Wang, Z. Liu, S. Guo, and X. Bai, “Neutralizing the impact of atmospheric turbulence on complex scene imaging via deep learning,” *Nature Machine Intelligence*, vol. 3, pp. 876 – 884, 2021.
- [164] R. Yasarla and V. M. Patel, “CNN-Based restoration of a single face image degraded by atmospheric turbulence,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 2, pp. 222 – 233, 2022.
- [165] Z. Mao, A. Jaiswal, Z. Wang, and S. H. Chan, “Single frame atmospheric turbulence mitigation: A benchmark study and a new

BIBLIOGRAPHY

- physics-inspired transformer model,” in *European Conference on Computer Vision*, 2022.
- [166] “Bridging the gap between computational photography and visual recognition: 5th UG2+ prize challenge.” http://cvpr2022.ug2challenge.org/dataset22_t3.html. Track 3.
 - [167] X. Zhang, Z. Mao, N. Chimitt, and S. H. Chan, “Imaging through the atmosphere using turbulence mitigation transformer.” Available online: <https://arxiv.org/abs/2207.06465>.
 - [168] D. N. Groff, K. J. Miller, and T. W. D. Bosq, “Towards development of improved metrics for quantifying turbulence imposed degradation of long-range video,” in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXXII*, p. 117400P, Proc. SPIE 11740, 2021.
 - [169] C. P. Lau and L. M. Lui, “Subsampled turbulence removal network,” *Mathematics, Computation and Geometry of Data*, vol. 1, no. 1, pp. 1 – 33, 2021.
 - [170] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
 - [171] S. M. Safdarnejad, X. Liu, L. Udpa, B. Andrus, J. Wood, and D. Craven, “Sports videos in the wild (svw): A video dataset for sports analysis,” in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, vol. 1, pp. 1–7, IEEE, 2015.
 - [172] D. Jin, Y. Chen, Y. Lu, J. Chen, P. Wang, Z. Liu, S. Guo, and X. Bai, “Neutralizing the impact of atmospheric turbulence on complex scene imaging via deep learning,” *Nature Machine Intelligence*, vol. 3, pp. 876–884, 2021.
 - [173] E. Repasi and R. Weiss, “Computer simulation of image degradations by atmospheric turbulence for horizontal views,” in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXII*, vol. 8014, p. 80140U, International Society for Optics and Photonics, 2011.

BIBLIOGRAPHY

- [174] C. Tian, Y. Xu, and W. Zuo, “Image denoising using deep cnn with batch renormalization,” *Neural Networks*, vol. 121, pp. 461 – 473, 2020.
- [175] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image restoration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2480 – 2495, 2021.
- [176] G. Chen, Z. Gao, Q. Wang, and Q. Luo, “U-net like deep autoencoders for deblurring atmospheric turbulence,” *Journal of Electronic Imaging*, vol. 28, no. 5, p. 053024, 2019.
- [177] R. Feng, J. Gu, Y. Qiao, and C. Dong, “Suppressing model overfitting for image super-resolution networks,” 2019. Available online: <https://arxiv.org/abs/1906.04809>.
- [178] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image super-resolution using very deep residual channel attention networks,” in *European Conference on Computer Vision*, pp. 294 – 310, 2018.
- [179] R. Nieuwenhuizen and K. Schutte, “Deep learning for software-based turbulence mitigation in long-range imaging,” in *Artificial Intelligence and Machine Learning in Defense Applications*, Proc. SPIE 11169, 2019.
- [180] N. Anantrasirichai, “Atmospheric turbulence removal with complex-valued convolutional neural network,” 2022. Available online: <https://arxiv.org/abs/2204.06989>.
- [181] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, “Restormer: Efficient transformer for high-resolution image restoration,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5728 – 5739, 2022.
- [182] N. Li, S. Thapa, C. Whyte, A. W. Reed, S. Jayasuriya, and J. Ye, “Unsupervised non-rigid image distortion removal via grid deformation,” in *IEEE/CVF International Conference on Computer Vision*, pp. 2522 – 2532, Oct. 2021.
- [183] C. P. Lau, H. Souri, and R. Chellappa, “Atfacegan: Single face semantic aware image restoration and recognition from atmospheric

BIBLIOGRAPHY

- turbulence,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, pp. 240 – 251, Feb. 2021.
- [184] N. G. Nair, K. Mei, and V. M. Patel, “AT-DDPM: Restoring faces degraded by atmospheric turbulence using denoising diffusion probabilistic models,” in *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3434–3443, 2023.

Index

- adaptive optics, 174
 - reference star, 174
 - target star, 174
- Airy disc, 46
- alternating minimization, 190
- amplitude spread function, 52
- amplitude transfer function, 59
 - relation to OTF, 62
- angle-of-arrival, 130
- angular spectrum propagation, 106
- anisoplanaticism, 171
- anisoplanatism, 117
- atmospheric turbulence, 69, 75
 - history, 73
- back propagation, 16
- basis representation, 110
 - definition, 121
 - of phase, 121
 - of PSF, 143
- brightness function, 116
- Cholesky decomposition, 127
- coherence, 54
 - spatial, 54
- complex wave function, 31
- computational image formation, 18
- convolution, 27, 64
 - gathering, 64
 - scattering, 64
- correlation, 54
- deconvolution, 187
- deep learning, 195
- defocus, 58
- denoising diffusion probabilistic model, 212
- differentiable simulator, 205
- differential tilt, 136
- diffraction, 26, 36
 - limit, 45
 - patterns, 45
- focal length, 49
- focal plane, 50
- Fourier burst accumulation, 181
- Fourier optics, 25
- Fraunhofer approximation, 42
- Fraunhofer diffraction, 42
- Fresnel approximation, 40
- Fresnel diffraction, 39
 - definition, 41
- Fried, 73
- Fried parameter, 88
- generative adversarial network, 210
- geometric consistency, 179
- geometric optics, 14
- geometrical optics, 52
- Helmholtz equation, 33
- homogeneity, 75
- Hufnagel-Valley profile, 77
- Huygens-Fresnel principle, 36

- hypothesis function, 19
 image formation
 coherent, 56
 incoherent, 57
 with phase errors, 59
 imaging system
 coherent, 59
 incoherent, 61
 incoherent, 54
 index of refraction
 atmospheric model, 76
 intensity, 45
 inverse problems, 156
 irradiance, 45
 isoplanatic angle, 89
 isoplanatism, 171
 isotropy, 79
 Karhunen–Loëve decomposition, 127
 Kolmogorov, 73
 lens law, 51
 linear space-invariant systems, 27
 log-amplitude, 86
 long exposure OTF, 93
 long-exposure image, 184
 Lucas-Kanade algorithm, 182
 lucky imaging, 170
 digital, 176
 fusion, 178
 lucky probability, 99
 magnification, 51
 Maxwell's equations, 34
 moment generating function, 82
 multi-aperture model, 120
 multiple apertures, 130
 mutual intensity, 55
 NATO RTG-40, 114
 Nijboer-Zernike diffraction theory, 145
 Noll matrix, 126
 non-local averaging, 185
 non-rigid deformation, 182
 numerical aperture, 47
 numerical wave propagation, 102, 105
 optical flow, 182
 optical frequency, 31
 optical transfer function, 61
 instantaneous, 91
 of circular aperture, 63
 relation to ATF, 62
 paraboloidal wave, 33
 paraxial approximation, 49
 path integral, 87
 phase aberration, 92
 phase domain, 117
 phase realization, 106
 phase screen, 102
 generation of, 102
 subharmonic generation, 105
 phase structure function, 89
 multiple layers, 85
 planar wave, 85
 single layer, 84
 spherical wave, 85
 phase-to-space transform, 145
 training of, 146
 phasor, 31
 Pickering's scale, 88
 planar wave, 33
 plug-and-play ADMM, 191
 point source, 32, 50
 point spread function, 57
 power spectral density, 76
 propagation-free methods, 112
 brightness function, 116

INDEX

- direct-to-image, 114
- generative adversarial network, 117
- phase-based simulation, 117
- Zernike-based, 118, 139
- zero-order, 112
- random process, 75
- random variable, 55
- random vector, 110
 - field, 128
- ray tracing, 24
 - rendering, 25
- Rayleigh criteria, 46
- Rayleigh-Sommerfeld diffraction, 38
 - definition, 39
- redegradation, 205
- reference frame, 184
- reference patch, 179
- refraction, 25
 - index of, 3, 26, 30
 - Snell's law, 25
- regularization function, 11, 156
- Robust principal component analysis, 184
- Rytov approximation, 87
- scintillation, 87
- scintillometer, 114
- self-supervised learning, 207
- short exposure, 96
- simulator, 19
- spatially varying blur
 - approximation, 142
- speed of light, 29
- spherical wave, 33
- split-step, 100
 - computational limitations, 110
 - history, 101
 - pseudo-code, 108
- structure constant, 76, 77
- structure function, 79
 - Kolmogorov, 79
 - turbulent phase, 83
- superposition, 64
- Tatarskii, 73
- thin lens, 48
 - model, 25, 48
 - transformation, 48
- tilt, 92, 122
- tilt-then-blur model, 160
 - blur operator, 162
 - tilt operator, 161
- Toeplitz matrix, 134
- total variation
 - anisotropic, 178
 - isotropic, 177
- transformer, 207
- turbulence, 75
 - Kolmogorov PSD, 76
 - layered model, 84
 - scale size, 76
 - structure function, 79
 - von Kármán PSD, 76
- turbulence datasets
 - testing, 197
 - training, 199
- turbulence operator, 157
- turbulence profile, 77
- uncertainty map, 205
- Voelz sampling criterion, 104
- wave equation
 - electromagnetic, 34
 - high dimensional (scalar), 30
 - one dimensional, 30
- Wave Structure Function, 86
- wave vector, 33
- wavefront, 31

INDEX

wavelength, 31
waves, 29
 common types of, 33
wide sense stationarity, 75
Wiener-Khinchin theorem, 76

z-tilt, 136
Zernike, 120
 intermodal correlation, 125, 126
 Noll's indexing, 123
 polynomial definition, 123
 polynomials, 120, 121
 space, 119
Zernike space, 128
 image formation, 138
 spatial correlation, 129
 spatial correlation (approximation), 132, 137
 spatial correlation (exact), 138
 tilt correlation (approximation), 133
Zernike-based simulation, 139
 pseudo-code, 150
 short summary, 147
WSS approximation, 141