UNIVERSITY OF CALIFORNIA,
IRVINE

Design of an End-to-End Consumer Imaging System Simulator With an Emphasis on Efficient Application of Spatially-Varying Optical Kernels

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical Engineering

by

Misagh Khayambashi

Dissertation Committee:
Professor A. Lee. Swindlehurst, Chair
Professor Glenn Healey
Professor Ozdal Boyraz

2018

TABLE OF CONTENTS

ABSTRACT OF THE DISSERTATION

Design of an End-to-End Consumer Imaging System Simulator With an Emphasis on
Efficient Application of Spatially-Varying Optical Kernels

By

Misagh Khayambashi

Master of Science in Electrical Engineering

University of California, Irvine, 2018

Professor A. Lee. Swindlehurst, Chair

In this thesis, we discuss the theoretical background and the practical aspects of the design of a simulation toolchain for prototyping and testing new consumer imaging technologies during the design cycle. To enable a fully computer-aided design and test process, mechanisms are provided to model and investigate all the entities involved in the imaging system. In particular, modeling of the scenes, optics, and sensors are included.

In addition, the computational and storage bottlenecks of the simulator are examined and more efficient methods are provided when appropriate. In particular, a method based on Singular Value Decomposition (SVD) is proposed to lower the storage-footprint of the scene model. Furthermore, an efficient algorithm based on Principal Component Analysis (PCA) and Fast Fourier Transform (FFT) is proposed to implement the convolution of the scene with the spatially-varying optical kernels. The quality and complexity of the proposed method are compared with previous work and it is shown that higher quality is achievable with lower complexity.

Chapter 1

Introduction

The design of modern consumer imaging systems requires a solid understanding of the scene radiometry, sophisticated optical systems, sensor technologies, and the image processing pipeline. Consequently, engineers and scientists with different expertise need to communicate their knowledge during the design phase to implement the ideas. Product development is slowed down when different specialists on the same product team cannot communicate effectively.

New designs need to be tested and verified under real-world conditions before proceeding to the mass-production phase. Traditionally, the testing requires fabricating prototypes, which is both expensive and time-consuming. The availability of an imaging system simulator can significantly reduce the cost of developing new products. To be useful, an ideal simulator needs to have a number of features:

End-to-End Simulation: The simulator should simulate the system all the way from the scene to the quality assessment. Rather than performing field experiments, the designers can use the realistic scene models provided by the simulator to test the design entirely on the computer and save time. The simulator should also be able to provide accurate and realistic models of the optical system and the sensor system to avoid the prototype fabrication cost associated with these components. In addition to modeling individual components, the simulator should be able to accurately model the interaction of the scene, optics, and sensors as well.

Modularity: The simulator should provide clear and well-defined interfaces for the individual components and for connecting the components to form a complete system. With a clean and well-encapsulated implementation, engineers of different expertise can use the interfaces to utilize the components and test their ideas without having to dig deep into other domains of expertise or having to spend time communicating with other members. The encapsulation and provision of clean interfaces calls for an object-oriented software design.

Extensibility: The simulator should be extensible. In other words, it should be possible to add new component models. Importantly, using the new components should not require making changes to the simulator itself. Such a plug-in architecture requires that the simulator defines interface contracts for any component that is to be plugged into the simulator.

Efficiency: A good simulator should be quick while providing accurate results. For modern imaging systems, hours of processing time are considered normal for simulating a single realistic image capture. A good simulator should be able to provide results in much shorter time to encourage the industry and the users to invest their time in the simulator.

Documentation: A good simulator should have a detailed and accessible description of the individual models and the interaction of the components. The lack of technically detailed documentation may discourage the developers from adding new modules to the simulator and extending its capabilities.

This thesis attempts to bring together the simulation elements required to create a simulator with the features mentioned above. After reviewing the theoretical background for each component, we also look into the practical aspects of implementing each component. We will discuss both the time complexity and the space complexity of the models and their interaction. We also identify the computational bottlenecks of the system at each stage and propose ideas to mitigate them.

In doing so, it was observed that the highest computational cost belongs to the image formation by the optical system. The optical system is responsible for converting the scene's radiometric parameters into the sensor's radiometric parameters. For realistic systems, this process requires convolving a large 2D dataset with a shift-variant impulse response. The previous work on the efficient implementation of the space-varying kernels has been studied and a new method based on PCA has been suggested. The computational complexity and quality of the proposed method has been investigated and compared to the previous work. Through simulations over a data-set of 40 scenes, we have shown that the proposed method offers higher quality results compared to the previous work while requiring a time complexity that is only linear in the scene size.

Chapter 2

Elements of Simulation

Photography systems use the interaction of electromagnetic fields and matter to capture information about the environment. The goal of an End-to-End simulator is to model this interaction as accurately as possible and provide mechanisms for using the model to predict interaction under a target scenario. The interaction chain starts by the scene emitting light, either through radiation from the sources or through reflection from reflecting surfaces. Then, the emitted light passes through the camera optical system (the lenses) and is delivered to the sensors to capture the information electrically. Once the information is captured, it will go through an image signal processing pipeline to remove the undesired qualities injected into the capture by the physical system. In the following sections, we model the different components of the system. The implementation of the models as well as the computational aspects are then discussed. When applicable and possible, methods with lower computational complexity are proposed and tested.

2.1 Scene Modelling

The scene-device interaction starts with the scene objects radiating or reflecting electromagnetic waves toward the device. Although very similar in nature, the mathematical formalization of reflection and radiation are different. In the following subsections, we provide formal models of radiation and reflection separately.

2.1.1 Radiation

To formalize the sources of electromagnetic fields in the scene, one could consider small elements of radiating matter as the building blocks of the model and then create the scene by assembling these small elements together.

Infinitesimally Small Harmonic 1D Dipole

Consider an infinitesimally small dipole d of length ℓ oriented in z direction located at the origin, with a unit-amplitude harmonic current oscillation at frequency ω . The far-field electric and magnetic field vectors observed at spherical coordinate $\mathbf{r} = (r, \theta, \phi)$ due to the dipole d are given by [1]:

$$\mathbf{H}_{d,\mathbf{r}}(t) = \frac{k\ell \sin\theta}{4\pi r} \cos(\omega t - kr)\hat{\boldsymbol{\phi}} \tag{2.1}$$

$$\mathbf{E}_{d,\mathbf{r}}(t) = \eta \frac{k\ell \sin\theta}{4\pi r} \cos(\omega t - kr)\hat{\boldsymbol{\theta}} \tag{2.2}$$

where k is the wave number and η is the intrinsic impedance of the medium. The dependence of normal vectors $\hat{\boldsymbol{\phi}}$ and $\hat{\boldsymbol{\theta}}$ on θ and ϕ has been dropped for notational simplicity.

Infinitesimally Small Incoherent 1D dipole

In reality, these small elements exhibit a stochastic temporal behaviour rather than a deterministic and perfect harmonic oscillation once excited by an internal or external source of energy (e.g. passing electric current or incident light). Although the temporal current variation of a small element of matter may look like a random signal, the frequency spectrum typically exhibits well-defined spectral power features. Formally, the temporal behaviour of an incoherent source may be modeled by Fourier techniques as a weighted superposition of harmonic activites, with matter-specific weights for different frequencies, but uncorrelated phases among different harmonics. Mathematically, if the magnitude of the dipole current is denoted by s:

$$s(t) = \int_0^\infty \sqrt{S(f)}\cos(2\pi ft + \nu_f)df \tag{2.3}$$

where $S(f)$ is the positive real-valued spectral power distribution, and ν_f is a random phase independetly selected for different frequencies. For such a source model, the linearity of Maxwell's equaitons can be used to obtain the following electric and magnetic fields for the incoherent oscillator dipole d located at the origin and aligned with z axis:

$$\begin{aligned}\mathbf{H}_{d,\mathbf{r}}(t) =&\hat{\boldsymbol{\phi}}\frac{k\ell\sin\theta}{4\pi r}\int_0^\infty \sqrt{S(f)}\cos(2\pi ft - kr + \nu_f)df\\ \mathbf{E}_{d,\mathbf{r}}(t) =&\hat{\boldsymbol{\theta}}\eta\frac{k\ell\sin\theta}{4\pi r}\int_0^\infty \sqrt{S(f)}\cos(2\pi ft - kr + \nu_f)df\end{aligned}$$

Infinitesimally Small Incoherent 2D Patches

As mentioned before, a realistic scene can be modeled as a collection of small dipoles, scattered in 3D space, radiating electromagnetic waves and interacting with each other. However, from the observer's point of view, if the scene objects are all opaque, the scene can be essentially modeled by a potentially smaller subset of dipoles that are visible to the observer. This is because the light sources behind opaque objects do not contribute significantly to the observation if the effect of multiple scattering from the entire space is negligible.

Formally, for each look angle originating at the observer's location, which is assumed to be the origin, a single visible dipole will lie on the first intersection of the line-of-sight and the scene. Mathematically, this intersection of the look-angle with the scene can be expressed as a function $r(\theta, \phi)$ which returns the distance from the observer to the visible dipole on look angle (θ, ϕ) in the observer's coordinate system. For a continuous range of θ and ϕ, $r(\theta, \phi)$

is basically expressing a 2D continuous plane. Consequently, it is more natural to consider the scene as a 2D surface consisting of smaller 2D patches rather than small one-dimensional dipoles.

With this model, there exists a small 2D patch surrounding the intersection of each look angle and the scene. A complete characterization of the 2D patches requires specifying a scene surface normal $\hat{\mathbf{n}}(\theta, \phi)$ at each look angle.

To formulate the radiation of a 2D patch, consider a small 2D patch located at the origin of a coordinate system $\mathcal{C}_{\text{patch}}$. Assume that the patch has a normal aligned with $\hat{\mathbf{z}}$ and a small area dA. Under incoherent radiation, any such patch can be considered as an ensemble of randomly oriented dipoles (but localized within the patch) with random-phase activity, but all following the same power spectral density. This is because the power spectral density is material-dependant, and since dA is small we can assume that all dipoles within dA belong to the same material. Let $\mathcal{D}_p$ denote the set of dipoles within the patch p. The complete characterization of the patch p requires provision of a set of dipole directions $\{(\theta_d, \phi_d), d \in \mathcal{D}_p\}$ specifying the orientation of individual 1D dipoles. This information may be provided as a discrete set of orientations per dipoles or alternatively as a probability distribution $P_p(\theta_d, \phi_d)$ for the dipoles in $\mathcal{D}_p$.

Particularly, the normal direction of a single dipole can be expressed as:

$$\hat{\mathbf{u}}_d = \sin\theta_d \cos\phi_d \hat{\mathbf{x}} + \sin\theta_d \sin\phi_d \hat{\mathbf{y}} + \cos\theta_d \hat{\mathbf{z}} \tag{2.4}$$

Note that from a far-field perspective, all dipoles can be assumed to be located at the center of the patch. For an individual dipole oriented as (θ_d, ϕ_d) and located at the origin, the fields at the observation point $\mathbf{r}_o = (\theta_o, \phi_o, r_o)$ are expressed by:

$$\mathbf{H}_{d,\mathbf{r}_o}(t) =\hat{\boldsymbol{\phi}}_{d,o}\frac{k\ell\sin\theta_{d,o}}{4\pi r_o}\int_0^\infty \sqrt{S(f)}\cos(2\pi ft-kr+\nu_{d,f})df \tag{2.5}$$

$$\mathbf{E}_{d,\mathbf{r}_o}(t) =\hat{\boldsymbol{\theta}}_{d,o}\eta\frac{k\ell\sin\theta_{d,o}}{4\pi r_o}\int_0^\infty \sqrt{S(f)}\cos(2\pi ft-kr+\nu_{d,f})df \tag{2.6}$$

where $\hat{\boldsymbol{\phi}}_{d,o}$ and $\hat{\boldsymbol{\theta}}_{d,o}$ are normal vectors of spherical coordinates with respect to the dipole coordinate system (with the dipole aligned to the z axis). Similarly, $\theta_{d,o}$ is the elevation of the observation point in the dipole's coordinate system. Using geometrical arguments, the three parameters $\hat{\boldsymbol{\phi}}_{d,o}$, $\hat{\boldsymbol{\theta}}_{d,o}$, and $\theta_{d,o}$ are linked with the parameters of coordinate system $\mathcal{C}_{\text{patch}}$ as:

$$\hat{\boldsymbol{\phi}}_{d,o} =\hat{\mathbf{r}}_o\times\hat{\mathbf{u}}_d \tag{2.7}$$

$$\hat{\boldsymbol{\theta}}_{d,o} =(\hat{\mathbf{r}}_o\times\hat{\mathbf{u}}_d)\times\hat{\mathbf{r}}_o \tag{2.8}$$

$$\cos\theta_{d,o} =\hat{\mathbf{u}}_d.\hat{\mathbf{r}}_o \tag{2.9}$$

Note that $\hat{\boldsymbol{\phi}}_{d,o}$ and $\hat{\boldsymbol{\theta}}_{d,o}$ satisfy $\hat{\boldsymbol{\theta}}_{d,o}\times\hat{\boldsymbol{\phi}}_{d,o}=\hat{\mathbf{r}}_o$.

Now, in order to combine the constituent dipoles of the patch, we need to investigate the spectral power and phase of the individual dipoles. The Poyting vector $\mathbf{S}$ is used to analyze the power of electromagnetic (EM) fields. The instantaneous Poynting vector at location $\mathbf{r}$ and time t is proportional to $\mathbf{E}_\mathbf{r}(t)\times\mathbf{H}_\mathbf{r}(t)$ and is interpreted as the amount of energy passing through a unit area perpendicular to $\mathbf{E}_\mathbf{r}(t)\times\mathbf{H}_\mathbf{r}(t)$ in a unit time. For the discussed patch model, with subscript p specifying the patch, we can write:

$$\begin{aligned}\mathbf{S}_{p,\mathbf{r}_o}(t) &\propto \mathbf{E}_{p,\mathbf{r}_o}(t)\times\mathbf{H}_{p,\mathbf{r}_o}(t)\\ &=\left(\sum_{d\in\mathcal{D}_p}\mathbf{E}_{d,\mathbf{r}_o}(t)\right)\times\left(\sum_{d\in\mathcal{D}_p}\mathbf{H}_{d,\mathbf{r}_o}(t)\right)=\sum_{d,d'}\mathbf{E}_{d,\mathbf{r}_o}(t)\times\mathbf{H}_{d',\mathbf{r}_o}(t)\end{aligned} \tag{2.10}$$

where $\mathbf{S}_{p,\mathbf{r}_o}(t)$ is the Poynting vector due to the patch p at observation point $\mathbf{r}_o$. The individual terms $\mathbf{E}_{d,\mathbf{r}_o}(t) \times \mathbf{H}_{d',\mathbf{r}_o}(t)$ are:

$$\mathbf{E}_{d,\mathbf{r}_o}(t) \times \mathbf{H}_{d',\mathbf{r}_o}(t) \propto \left(\frac{k\ell}{4\pi}\right)^2 \eta\hat{\mathbf{r}}_o \frac{\sin\theta_{d,o}\sin\theta_{d',o}}{r_o^2} \times$$
$$\left[\int_0^\infty \sqrt{S(f)}\cos(2\pi f t - kr + \nu_{d,f})df\right]\left[\int_0^\infty \sqrt{S(\lambda)}\cos(2\pi \lambda t - kr + \nu_{d',\lambda})d\lambda\right] \quad (2.11)$$

This is a vector pointing in the $\hat{\mathbf{r}}_o$ direction whose magnitude randomly oscillates in time. In practice, we are not interested in the actual temporal variation. Rather, we are interested in the *time-avergaed* flow of power given by:

$$\mathbf{S}_{p,o} = \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathbf{S}(t)_{p,o} \quad (2.12)$$

Thus:

$$\mathbf{S}_{p,o} \propto \left(\frac{k\ell}{4\pi}\right)^2 \eta \sum_{d,d'} \hat{\mathbf{r}}_o \frac{\sin\theta_{i,o}\sin\theta_{j,o}}{r_o^2} \times$$
$$\lim_{T\to\infty}\frac{1}{T}\int_0^T\int_0^\infty\int_0^\infty \sqrt{S(f)S(\lambda)}\cos(2\pi f t - kr + \nu_{d,f})\cos(2\pi\lambda t - kr + \nu_{d',\lambda})df d\lambda dt$$
$$(2.13)$$

Using the identities $2\cos(a)\cos(b) = \cos(a+b)+\cos(a-b)$ and $\lim_{T\to\infty}(\int_0^T \cos(\omega t)dt)/T = 0$ we can write:

$$\mathbf{S}_{p,o} \propto \left(\frac{k\ell}{4\pi}\right)^2 \eta \sum_{d,d'} \hat{\mathbf{r}}_o \frac{\sin\theta_{d,o}\sin\theta_{d',o}}{r_o^2} \times \int_0^\infty S(f)\cos(\nu_{d,f} - \nu_{d',f})df \quad (2.14)$$

which can be further extended as:

$$\mathbf{S}_{p,o} \propto \left(\frac{k\ell}{4\pi}\right)^2 \eta \left[\sum_{d\neq d'} \hat{\mathbf{r}}_o \frac{\sin\theta_{d,o}\sin\theta_{d',o}}{r_o^2} \times \int_0^\infty S(f)\cos(\nu_{d,f} - \nu_{d',f})df \right.$$
$$\left. + \sum_{d} \hat{\mathbf{r}}_o \frac{\sin^2\theta_{d,o}}{r_o^2} \times \int_0^\infty S(f)df\right] \quad (2.15)$$

When the number of dipoles in a patch is very large, the independence of the ν terms causes the first term to vanish. Thus:

$$\mathbf{S}_{p,o} \propto \left(\frac{k\ell}{4\pi}\right)^2 \eta \sum_d \hat{\mathbf{r}}_o \frac{\sin^2\theta_{d,o}}{r_o^2} \times \int_0^\infty S(f)df = \left(\frac{k\ell}{4\pi}\right)^2 \eta \frac{\hat{\mathbf{r}}_o}{r_o^2} \left(\int_0^\infty S(f)df\right) \sum_d \sin^2\theta_{d,o} \tag{2.16}$$

Given the probability distribution of the dipole orientations in a patch p with D dipoles, $P_p(\theta_d, \phi_d)$, the summation can be replaced by the following averaging:

$$\begin{aligned}\sum_d \sin^2\theta_{d,o} &= D \int P_p(\theta_d, \phi_d)(1 - \cos^2(\theta_{d,o}))d\theta_d d\phi_d \\ &= D \int P_p(\theta_d, \phi_d)(1 - (\hat{\mathbf{u}}_d.\hat{\mathbf{r}}_o)^2)d\theta_d d\phi_d \\ &= D \int P_p(\theta_d, \phi_d)(1 - (\sin\theta_d \cos\phi_d \sin\theta_o \cos\phi_o \\ &\qquad + \sin\theta_d \sin\phi_d \sin\theta_o \sin\phi_o \\ &\qquad + \cos\theta_d \cos\theta_o)^2)d\theta_d d\phi_d \\ &= DL(\theta_o, \phi_o)\cos\theta_o \end{aligned} \tag{2.17}$$

To be consistent with the classic radiometric definitions, the integral result $L(\theta_o, \phi_o)\cos\theta_o$ explicitly factors a cosine term. The integral result depends on the material-specific distribution $P_p(\theta_d, \phi_d)$. As an example, the distribution $P_p(\theta_d, \phi_d)$ for a Lambertian surface is such that $L(\theta_o, \phi_o)$ is equal to 1.

Plugging the last result into the Poynting vector equation results in:

$$\mathbf{S}_{p,o} \propto \left(\frac{k\ell}{4\pi}\right)^2 D\eta \frac{\hat{\mathbf{r}}_o}{r_o^2} \left(\int_0^\infty S(f)df\right) L(\theta_o, \phi_o)\cos\theta_o \tag{2.18}$$

The frequency integration is simply adding power contributions from different frequencies. Also, the radiated energy is proportional to D, which is directly related to patch area dA. Furthermore, the ℓ^2 term may be absorbed into $S(f)$ as it only affects the radiated power as a scaling. After factoring all the constants into $S(f)$, we may write:

$$\mathbf{S}_{p,o} = dA \cos\theta_o \frac{\hat{\mathbf{r}}_o}{r_o^2} \left(\int_0^\infty S(f) df \right) L(\theta_o, \phi_o) \tag{2.19}$$

If interpreted in a geometric optics context, the term $S(f)L(\theta_o, \phi_o)$ is equivalent to the spectral radiance at $\mathbf{r}_p$, with the direction $\hat{\mathbf{n}}_{p,o}$, per unit frequency around f, namely $P(\mathbf{r}_p, \hat{\mathbf{n}}_{p,o}, f)$. The spectral radiance has the units of $W/stradian.Hz.m^2$. Alternatively, if S was defined with respect to the wavelength λ, the spectral intensity would have a unit of $W/str.m^3$. Other radiometric quantities are defined based on spectral radiance.

Collection of 2D patches

The formulations so far have been derived in the local coordinate system of a patch. In a real scene, we deal with a large number of patches, each with its own local coordinate system, but located in a global coordinate system $\mathcal{C}_{\text{global}}$. Suppose that the scene is tesselated (discretized) by a large number of small patches, with each patch p specified by:

- $\hat{\mathbf{r}}_p$ as the location of the center of the patch in the global coordinate system
- a local coordinate system whose $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ are given by $\hat{\mathbf{t}}_{1,p}$, $\hat{\mathbf{t}}_{2,p}$, and $\hat{\mathbf{n}}_p$ in the global coordinate system
- area dA_p
- Spectral radiance $S_p(f)L_p(\theta_{p,o}, \phi_{p,o})$, with respect to the local coordinate system

Using the single patch results, the power received per unit frequency from all patches by an observation patch at location $\hat{\mathbf{r}}_o$ in the global coordinate system, with area dA_o, and patch normal $\hat{\mathbf{n}}_o$ is given by:

$$p_o = \sum_p \left(dA_o (\hat{\mathbf{n}}_{p,o} . \hat{\mathbf{n}}_o)^- \right) \frac{dA_p \cos\theta_{p,o}}{r_{p,o}^2} S_p(f) L_p(\theta_{p,o}, \phi_{p,o}) \tag{2.20}$$

The subscript p has been added to S and L to emphasize that different patches can have different spectral characterizations and radiation patterns. Also, $(\theta_{p,o}, \phi_{p,o}, r_{p,o})$ is the spherical coordinate of obsevation point o when observed from the local coordinate system of patch p. Finally, $\hat{\mathbf{n}}_{p,o}$ is the unit normal pointing from patch p toward observation point o in global coordinates. The term $(\hat{\mathbf{n}}_{p,o}.\hat{\mathbf{n}}_o)^-$ is 0 when the patch is not facing the observation point (when $\hat{\mathbf{n}}_{p,o}.\hat{\mathbf{n}}_o$ is positive). The $(.)^-$ operator returns the absolute value of the input if the input is negative, and returns 0 otherwise.

Furthermore, we can write the intermediate variables $\theta_{p,o}$ and $\phi_{p,o}$ explicitly as:

$$\begin{aligned}\theta_{p,o} =& \text{acos}(\hat{\mathbf{n}}_{p,o}.\hat{\mathbf{n}}_p)\\ \phi_{p,o} =& \text{atan}(y_{p,o}, x_{p,o})\end{aligned} \tag{2.21}$$

where $y_{p,o}$ and $x_{p,o}$ are the x and y component of $\hat{\mathbf{n}}_{p,o}$ observed in the local coordinates of p, and atan is a full quadrant inverse tangant with first argument y and second argument x. From geometry:

$$\begin{aligned}x_{p,o} =& \hat{\mathbf{t}}_{1,P}.\text{orth}_{\hat{\mathbf{n}}_p}(\hat{\mathbf{n}}_{p,o})\\ y_{p,o} =& \hat{\mathbf{t}}_{2,P}.\text{orth}_{\hat{\mathbf{n}}_p}(\hat{\mathbf{n}}_{p,o})\end{aligned} \tag{2.22}$$

with $\text{orth}_{\mathbf{b}}(\mathbf{a})$ returning the component of vector $\mathbf{a}$ orthogonal to $\mathbf{b}$:

$$\text{orth}_{\mathbf{b}}(\mathbf{a}) = \mathbf{a} - \hat{\mathbf{b}}(\mathbf{a}.\hat{\mathbf{b}}) \tag{2.23}$$

2.1.2 Reflection

A reflector patch receives energy from the environment and radiates some of the energy in response. Note that a reflector is essentially a radiator if the radiation from the reflector is given. However, we can go one step further and, rather than assuming that the radiation pattern of the reflector is given, calculate the pattern in terms of the radiation received from the environment and the 'reflectivity properties' of the patch.

Consider a reflector patch q of area dA_q at location $\hat{\mathbf{r}}_q$ with a local cartesian coordinate system whose unit vectors are $\hat{\mathbf{t}}_{1,q}$, $\hat{\mathbf{t}}_{2,q}$, and $\hat{\mathbf{n}}_q$. Suppose that q is exposed to the radiation of a radiator patch p of area dA_p at location $\hat{\mathbf{r}}_p$. The patch p has a local cartesian coordinate system whose unit vectors are $\hat{\mathbf{t}}_{1,p}$, $\hat{\mathbf{t}}_{2,p}$, $\hat{\mathbf{n}}_p$. Furthermore, p radiates according to the spectral radiance $S_p(f)L_p(\theta_{p,o}, \phi_{p,o})$.

The directional power received at q per unit frequency due to p is given by:

$$\hat{\mathbf{n}}_{p,q} df \frac{dA_p \cos\theta_{p,q}}{r_{p,q}^2}(dA_q \cos\theta_{q,p}) S_p(f) L_p(\theta_{p,q}, \phi_{p,q}) \tag{2.24}$$

A generic reflector will respond to this received power with a pattern that potentially depends on both the direction of incident energy $\hat{\mathbf{n}}_{p,q}$ and the frequency f.

Formally, the reflection should relate the emitted spectral radiance from q toward some $\hat{\mathbf{n}}_{out}$, $P(\mathbf{r}_q, \hat{\mathbf{n}}_{out}, f)$, to the received spectral irradiance $I(\mathbf{r}_q, \hat{\mathbf{n}}_{in}, f)$, where spectral irradiance $I(\mathbf{r}_q, \hat{\mathbf{n}}_{in}, f)$ is defined as the power received by a unit area of the receiver (not the unit projected area) per unit frequency from a given direction from a cone of unit stradian. Note that this definition assumes that the power received in a narrow frequency band will not be shifted to a different band upon reflectance. The reflectance is then defined by:

$$R(\mathbf{r}_q, \hat{\mathbf{n}}_{in}, \hat{\mathbf{n}}_{out}, f) = \frac{P(\mathbf{r}_q, \hat{\mathbf{n}}_{out}, f)}{P(\mathbf{r}_q, \hat{\mathbf{n}}_{in}, f) \cos\theta_{in} d\Omega_{in}} = \frac{P(\mathbf{r}_q, \hat{\mathbf{n}}_{out}, f)}{I(\mathbf{r}_q, \hat{\mathbf{n}}_{in}, f)} \tag{2.25}$$

The simplest model of reflection belongs to a Lambertian surface where the reflectance is a constant and consequently independent of the incident and exit direction. Some of the other common models are Phong, Specular, Diffuse, Generalized Lambertian, and Pitted Surfaces [2].

2.2 Practical Considerations for the Scene

In order to make the proposed software platform extensible to more complex scene models, we require that a scene provides an interface to calculate the radiance at the entrance pupil of the camera optics. This is because the response of the optical system is uniquely determined by the radiance received at the optics entrance pupil. The optics component of the simulator does not care about the internal implementation of the scene and its complexity. The advantage of this structure is that more complex scene models that comply with this requirement may be developed in the future and used in the simulator.

In the most general scenario, a scene may be constructed from a set of 3D voxels, where each voxel could radiate light as a source, reflect the light as a reflector, or pass the light through. A voxel may generally play all these roles at the same time. Although such fully-3D models are superior for rendering quality, they do suffer from much higher space and time complexity compared to surface scene models where the scene consists of 2D patches located throughout the space. Furthermore, if a simulator emphasizes testing of consumer camera systems, a surface scene models will provide sufficient flexibility for testing scenarios. Therefore, this thesis will focus on opaque surface scene models, where the scene consists of 2D patches throughout the space. The 2D patches can serve as radiators or opaque reflectors, but not both at the same time. This is because transparent objects typically have no significance in a camera testing scenario; A 2D scene with opaque surfaces will be flexible enough to test image quality metrics such as color accuracy, sharpness, dynamic range compression, and optical fidelity among many other metrics of interest.

The most general surface scene model may be obtained by tessellating the continuous surfaces into 2D polygon patches and then characterizing each constituent 2D patch both spatially and radiometrically. The spatial characterization of each patch consists of a) the location of the center of the patch with respect to some coordinate system, b) the surface normal in that coordinate system, and c) either the area of the patch or the vertices of the patch, the latter of which imposes higher space complexity. In this thesis, we describe the spatial span of each patch with its scalar area rather than by its vertices. However, vertex-specified scenes can also be used in the simulator as long as they comply with the interface.

Finally, two more assumptions are added to reduce the computational complexity of the scene. First, it is assumed that, for point of views (PoVs) of interest, there is a direct line of sight between each patch and the PoV if the patch is not pointing away from the PoV; this will avoid the computational complexity of solving the visibility problem [3]. Furthermore, we ignore the effect of multiple reflections and scattering [4]. In other words, the only interactions in the scene are source-to-reflector and reflector-to-PoV, and there won't exist any reflector-to-reflector or reflector-to-source interaction. Nevertheless, if a scene model is internally capable of handling the visibility problem and multiple scattering, it may be used in our platform as long as it complies with the interface.

With these assumptions, a scene may be specified by a set of reflector objects and a set of illuminator objects located within a coordinate system. Each illuminator or reflector object is then characterized by a set of 2D patches, and each 2D patch is in return specified spatially and spectrally. The spatial properties include the location of the center of the patch, its area, and its normal vector. Note that the vector properties are all given in the coordinate system that is included in the object description. The spectral characterization of each patch depends on whether the patch is a reflector or an illuminator. For an illuminator, the radiance is specified as a mapping from the (frequency,direction) space to $\mathbb{R}$. For a reflector, the reflectance is a mapping from (frequency,direction,direction) space to $\mathbb{R}$.

It is noteworthy that different objects may be specified in different coordinate systems. In order to combine the elements of different objects together, it is necessary that the local coordinate system of each object is specified within a global coordinate system as a reference. Without a global coordinate system, it is impossible to formulate the interactions between objects and calculate the received radiance at a viewpoint.

In the following subsections, we discuss some of the design aspects and challenges of the proposed framework. The reader should keep in mind that there is no *best* scene implementation; while the implementation footprint may be reduced under certain assumptions, complicated models may be necessary under more realistic scenarios. Regardless of the selected scene implementation method, the scene may be used in the simulation if it provides a method to calculate the radiance resulting from the scene at a PoV.

2.2.1 Spatial Information Storage

Consider a scene with a total of N_p patches. Without further information about the scene, each patch should include the location of its center, its normal vector, and its area. This will require storing 7 numbers per patch, leading to a total spatial complexity of $7N_p$. A typical scene can have polygon counts anywhere from 10k to 500k depending on the level of detail. With 500k polygons, and assuming that each number is stored as a double, the total required storage will be about 27 megabytes. Although this is typically not a problem for a modern computer systems, the storage footprint may be reduced for some special structures. The following is an incomprehensive list of possibilities:

Planar Scene: If the object of interest is a rectangular plane, and if the patches form a uniform square tessellation of the plane, then all the patches have the same normal vector and area. Furthermore, their location is easily encoded as a pixel index (x, y) on the plane. Consequently, we only need to store the plane dimensions, the spacing between patch centers, the location of its center, and the normal vector of the plane. As a result, the memory footprint is very small and independent of the number of patches.

General Parameterized Surface: If the location and normal of each point on the surface is available analytically as a function of some parameter vector $\mathbf{p}$, then there is no need to store the location and normal per patch on the surface. Note that this is a generalization of the planar scene. In this case, we only need to store the information about the discretization of the parameter space $\mathbf{p}$. A possible disadvantage is that the function should be called every time we need the spatial characteristics of a patch; if the function is complex, the time complexity increases and we may end up shifting the spatial complexity to time complexity.

An example of such a surface is the corrugated surface $z - \sin(2\pi x) = 0$ for $\mathbf{p} = (x, y) \in [-3, 3] \times [-3, 3]$, with normal vector $\hat{\mathbf{n}}(x, y) = (-2\pi \cos(2\pi x), 0, 1)/||(-2\pi \cos(2\pi x), 0, 1)||$

2.2.2 Spectral Information Storage

Illuminant Spectral Information

For the 2D patches that serve as light sources, the radiance should be available to calculate the interaction with the reflectors. Theoretically, for each 2D patch, we should store a mapping (direction,frequency)$\rightarrow \mathbb{R}$, where the direction is specified with respect to the patch's local coordinate. This will be a data cube per 2D patch. The spatial footprint depends on how we represent this mapping. As an example, consider discretizing the data cube along frequency, elevation, and azimuth axes as 400:10:700 nm, $-\pi/2 : \pi/60 : \pi/2$ radians, and $0 : \pi/60 : 2\pi$. This discretization requires $30 \times 60 \times 120 \approx 200k$ numbers per patch, or equivalently 1.6 megabytes per patch. Even with a less-than-typical 10k patche scene, the memory footprint will be about 16 gigabytes. Clearly, a raw representation of the data cube is highly inefficient. The spatial complexity may be reduced under certain conditions:

Radiance Available As a Function: If the radiance per patch is available as a function, the spatial complexity per patch reduces to storing the parameters of the corresponding function.

Patches share the Same Radiance: If multiple patches share the same radiance model, the memory footprint may be reduced significantly. For instance, if a scene consists of 100 different surface types with 100 distinct radiance models, each patch only needs to hold the index of the corresponding radiance model. Consequently, the spatial footprint is equal to the number of patches plus the space required for storing the information about the 100 distinct radiance models. For a 10k patch scene with a radiance data cube size of 1.6 megabytes per patch, the memory footprint is equal to 10 kilobytes plus 100×1.6 megabytes ≈ 160 megabytes total. If the storage per radiance model is reduced, the memory footprint may be further lowered.

Factorable Radiance: If the radiance function can be factored into a spectral component and a direction component as $P(\theta, \phi, f) = D(\theta, \phi)F(f)$, the spatial complexity reduces from $O(D)O(F)$ to $O(D) + O(F)$, where $O(D)$ and $O(F)$ are the space required for storing $D(\theta, \phi)$ and $F(f)$ respectively. If the function $D(\theta, \phi)$ can be further factored as $\Theta(\theta)\Phi(\phi)$, the memory footprint becomes even smaller. For a radiance data cube discretization along frequency, elevation, and azimuth axes as 400:10:700 nm, $-\pi/2 : \pi/60 : \pi/2$ radians, and $0 : \pi/60 : 2\pi$, a fully factored representation requires only $30 + 60 + 120 \approx 200$ numbers per patch, or 1.6 kilobytes per patch. In a 10k patch system, this only requires a total storage of 16 megabytes.

Correlation Between Radiance Functions: Often, the set of radiance functions of all the patches is highly redundant. In other words, strong correlations or similarities exist between the radiance functions of certain patches. Consequently, we may think of the set of radiance functions as points lying on a low dimensional manifold / surface in the space of radiance functions. In other words, dimensionality reduction methods such as the Singular Value Decomposition (SVD) may be applied to decompose each radiance function as a weighted sum of some basis functions. The same concept may also be applied to reduce the dimension of reflection representation as discussed later.

Radiance is independent of direction: The minimal spatial footprint may be obtained when the radiance function is independent of the direction and only depends on the frequency. Furthermore, it may be possible to represent the frequency component efficiently using any combination of previous scenarios.

Reflector Spectral Information

For the 2D patches that serve as reflectors, the reflection should be available to calculate the interaction with the illuminants and to find the radiance received at some point of interest. In general, each 2D patch requires a mapping (incident direction, outgoing direction, frequency)$\rightarrow \mathbb{R}$, where the incident and outgoing direction are specified with respect to the patch's local coordinate. This will be a 5D data hyper cube per 2D patch. Compared to the illuminant spectral representation, the general reflector spectral representation imposes a much higher footprint. Thus, efficient representations of the mapping is critical for reflectors. As an example, consider discretizing the 5D data hyper-cube along frequency, elevation, and azimuth axes as 400:10:700 nm, $-\pi/2 : \pi/60 : \pi/2$ radians, and $0 : \pi/60 : 2\pi$. This translates to $30 \times 60^2 \times 120^2 \approx 1.5$ billion numbers per patch, or equivalently 10 gigabytes per patch. Clearly, encoding a many-patch scene is impossible using this raw representation. The spatial complexity may be reduced under various assumptions:

Reflectance Available as a Function: If the reflectance per patch is available as a function, the spatial complexity per patch reduces to storing the corresponding function parameters.

Patches Share the Same Reflectance: If multiple patches share the same reflectance model, the memory footprint may be reduced significantly by requiring each patch to only hold the index of the corresponding reflectance model.

Factorable Reflectance: Similar to the factorable radiance functions, if the radiance function $R(f, \theta_{in}, \phi_{in}, \theta_{out}, \phi_{out})$ can be factored into components, the spatial complexity reduces from the product of individual complexities to the sum of individual complexities.

Although any factorization can help, a full factorization of the form:

$$R = F(f)\Theta_{in}(\theta_{in})\Phi_{in}(\phi_{in})\Theta_{out}(\theta_{out})\Phi_{out}(\phi_{out}) \tag{2.26}$$

is the most efficient. For a radiance data cube discretization along frequency, elevation, and azimuth axes as 400:10:700 nm, $-\pi/2 : \pi/60 : \pi/2$ radians, and $0 : \pi/60 : 2\pi$, a fully factored representation requires only $30 + 2 \times 60 + 2 \times 120 \approx 400$ numbers per patch, or 3.2 kilobytes per patch. In a 10k patch system, this only requires a total storage of 32 megabytes.

Correlation Between Reflectance Functions: As mentioned before , the set of reflectance functions of a scene is highly redundant, and the strong correlations may be exploited to project the radiance functions into a low dimensional manifold / surface. As an example, consider the reflectance of color patches of a Munsell chart as shown in Fig. 2.1. Each reflectance curve is embedded in a matrix, and then the SVD is applied to the matrix to obtain the singular values and vectors shown in Fig. 2.2. Given the fast decay of the singular values, it is possible to represent the reflectances by storing a) only the first ten singular vectors as the basis vectors, and b) the 10×1 vector of projections of each reflectance curve onto the basis vectors. For this example, if the original discretization of the reflectance information per patch was over the grid 400:10:700 nm, the spatial footprint reduces from 30 numbers per patch to 10 numbers per patch. Note that a spatial overhead is imposed by the requirement to store the 10 basis vectors as well.

Reflectance Only Depends on Frequency: For many diffused reflectors, the reflectance is only a function of the frequency and independent of the incident and outgoing angles. For such reflectors, each patch may only be required to store a purely spectral information, which in turn may be represented efficiently using any combination of previous scenarios.

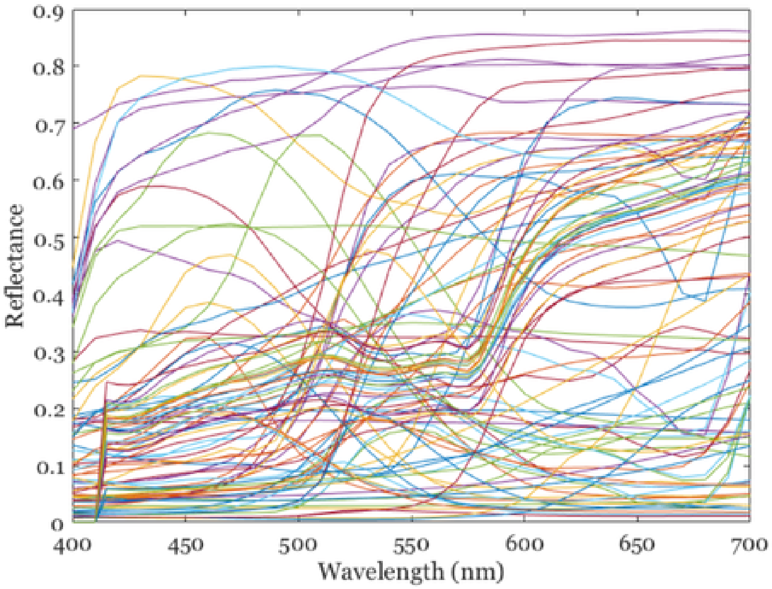


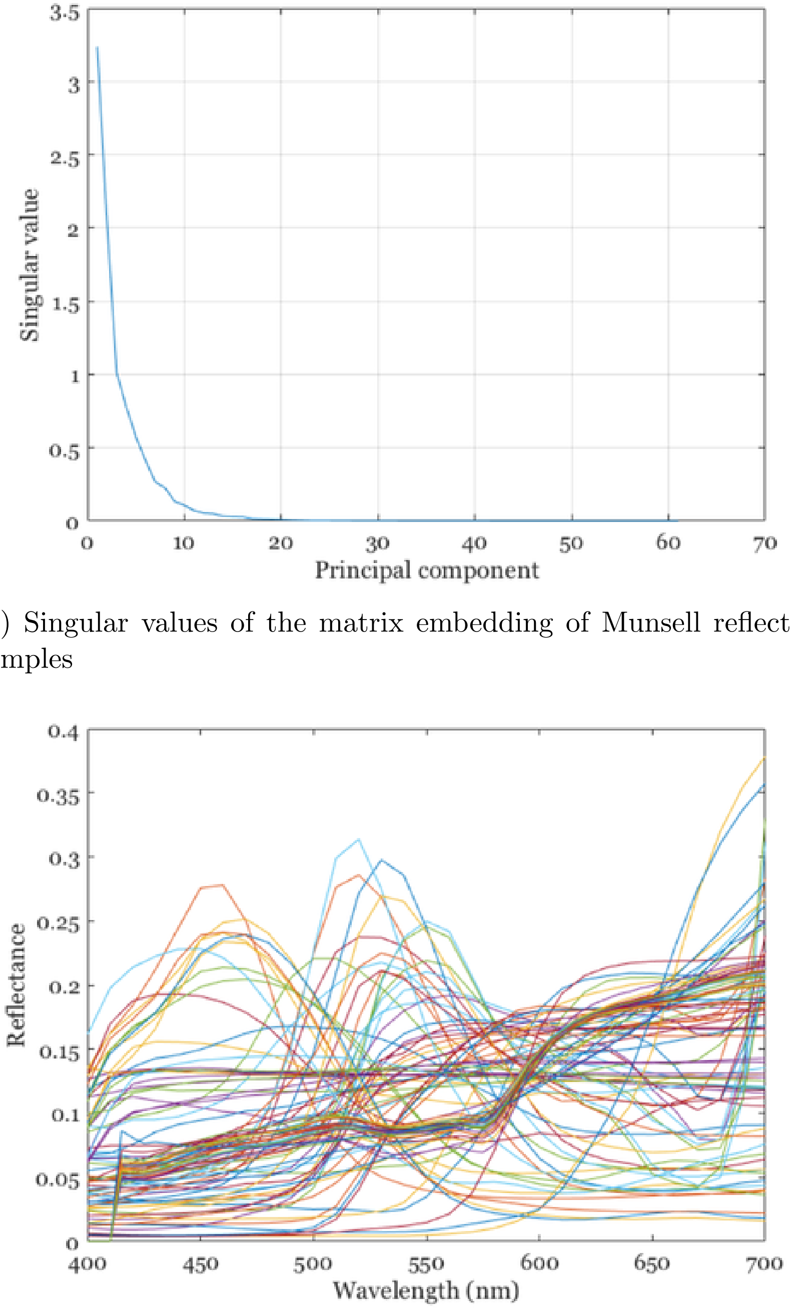
Figure 2.1: Reflectance of Munsell color chart patches

2.3 Optics Modelling

Ideally, the optics object used in the simulation toolchain should convert the radiance of the scene patches to the radiance received at points on a measurement plane hidden behind the optics. In a camera system, the sensors coincide with the measurement plane and convert the received radiance to electrical signals that may be recorded and processed later through the image processing pipeline.

Formally, consider a scene patch p located at $\mathbf{r}_p$ with normal vector $\hat{\mathbf{n}}_p$. The radiance of this patch is given by some function $P_p(f, \theta_{p*}, \phi_{p*})$, where θ_{p*} and ϕ_{p*} are the polar angles of a direction emanating from p and measured in p's local coordinate system. For a given measurement point m located at $\mathbf{r}_m$, the optical system uses the patch radiance function $P_p(f, \theta_{p*}, \phi_{p*})$ as well as the spatial information about the patch, optics, and measurement point, and calculates the radiance $P_m(f, \theta, \phi)$ received at m for all incident angles (θ, ϕ) at m.

Accurate sensor models require knowledge of the received radiance to calculate the measured electrical signal. However, calculating the radiance per reception point is relatively



(a) Singular values of the matrix embedding of Munsell reflectance samples

(b) Singular vectors of the matrix embedding of Munsell reflectance samples

Figure 2.2: Efficient reflectance representation by dimensionality reduction

time consuming since it has to consider all reception angles. To simplify, typical sensor models use an approximation scheme where the recorded electrical signal is related to a) the received irradiance, which is simply the radiance integrated over all incident angles and thus independent of the angle, and b) a scaling of the irradiance governed by the relative position / orientation of the sensing cell with respect to the optics. With this irradiance-based model, the optical system may be designed to convert the scene radiance directly to the irradiance received at the measurement plane. Since irradiance does not depend on a specific look angle, an irradiance-based optical model has lower time and space complexity.

The conversion of scene radiance to measurement plane (ir)radiance may be implemented with different levels of accuracy and complexity depending on the scene and optics model. Alternatively, it is possible to have a generic implementation with a fixed complexity which accommodates the most complicated scene and optics model without exploiting the potential efficient implementations for the simpler models. The latter is not desirable from a computational complexity point of view. Therefore, in order to enable a scene- and optic-dependent conversion, we propose the addition of a measurement plane object to the simulator. The measurement plane object takes as input both the scene object and the optics object and calculates the received (ir)radiance based on the scene and optics models. It is important to keep in mind that not all optical models provide mechanisms to calculate the irradiance at arbitrary measurement planes / surfaces. As an example, a diffraction limited optical model can only calculate the irradiance received at its focal plane.

In the following subsections, we discuss different optics models and how they can be used together with different scene models to generate the irradiance at the valid measurement planes/surfaces. Different optics models may be used depending on the level of accuracy required and the tolerable computational complexity. Examples are shift-invariant optics, ray-trace methods, and diffraction-limited optics. In shift-invariant optics, there is a different shift-invariant pointspread function (PSF) for each wavelength. In the ray-trace method, optical analysis software such as Zemax may be used to create a shift-variant set of PSFs along with a geometric distortion function. The simplest optical model is the diffraction limited optics which uses a combination of geometrical optics and diffraction theory to calculate the shift-invariant PSFs.

2.3.1 Diffraction-Limited Optics

The diffraction-limited model of the optics starts with a perfect geometrical model of a thin lens system and then augments the results of gemoetric optics with physical optics and diffraction theory at various levels of accuracy. Although this 2-stage scheme is only an approximation to an accurate EM-based method, it is typically more than sufficient for typical imaging applications.

A perfect thin lens system may be characterized by a set of interchangable parameters, the most common of which is the f-number and the field-of-view. Consider a thin lens lying on the $x-y$ plane and centered at the origin. Denote the diameter of the lens by D and the focal length of the lens by L. The thin lens is characterized by two properties: 1) a ray of light incident on the center of the lens will continue undisturbed, and 2) a ray of light hitting the edge of the lens will be directed toward the focal point ($z=-D$) of the lens if the incoming ray is within the field of view of the lens. The field of view is defined as $FoV = 2\text{atan}((D/2)/L)$ in radians. Furthermore, f-number is defined as the ratio of the focal length to the diameter as $f\#=L/D$.

Now, consider a general light emitting/reflecting patch p located at $\mathbf{r}_p$ with normal $\hat{\mathbf{n}}_p$ and radiance $P(f,\theta_{p*},\phi_{p*})$. Assuming that the light source and the optics are far enough away, the irradiance received at the optics entrance pupil due to patch p is given by:

$$I_p = \left[\frac{dA_p(-\hat{\mathbf{n}}_p.\hat{\mathbf{r}}_p)}{r_p^2}\right]\left[\pi(\frac{D}{2})^2\cos\theta_p\right]P(f,\theta_{p,o},\phi_{p,o}) \tag{2.27}$$

Here, $\hat{\mathbf{r}}_p$ is the unit vector pointing from the origin to the patch, and θ_p is the elevation of

p seen from the origin. Assuming that the patch p is within the field of view of the lens and $z_p > L$, the perfect lens will concentrate the received irradiance maximally at a point (x'_p, y'_p, z'_p) behind the optics as follows:

$$\frac{1}{-z'_p} = \frac{1}{L} - \frac{1}{z_p} \tag{2.28}$$

$$x'_p = m_p x_p \tag{2.29}$$

$$y'_p = m_p y_p \tag{2.30}$$

where m_p is the magnification given by $m_p = L/(L - z_p)$. To formulate the response of the optical system to patch p, we need to consider the planes or surfaces that cut through the geometric cone of light that converges on $\mathbf{r}'$. The response of the system is then formulated with respect to the coordinates describing the cross section of the plane and the cone of light.

In the simplest scenario, if a measurement plane touches the point $\mathbf{r}' = (x'_p, y'_p, z'_p)$, it will receive the entire irradiance at that point. Mathematically, we say that the irradiance response of the optical system to the patch p on a plane $\mathcal{P}_{\mathbf{r}'}$ passing through location $\mathbf{r}'$ is (up to a scaling) equal to $\delta(\mathbf{r}_{\mathcal{P}_{\mathbf{r}'}} - \mathbf{r}')$, where δ is 3 dimensional Dirac delta function, and $\mathbf{r}_{\mathcal{P}_{\mathbf{r}'}}$ is some point on $\mathcal{P}_{\mathbf{r}'}$. It is noteworthy that the energy of light arrives at $\mathbf{r}'$ from different directions, and this angular distribution of the incident energy can affect the energy received by a sensor located at $\mathbf{r}'$. However, the irradiance response disregards this angular information. To bridge this gap, the irradiance information may be augmented with an assumed incident direction of $\hat{\mathbf{n}}_{o,\mathbf{r}'}$. This approximation is accurate for most systems since the angular distribution of the incident energy is highly concentrated around $\hat{\mathbf{n}}_{o,\mathbf{r}'}$.

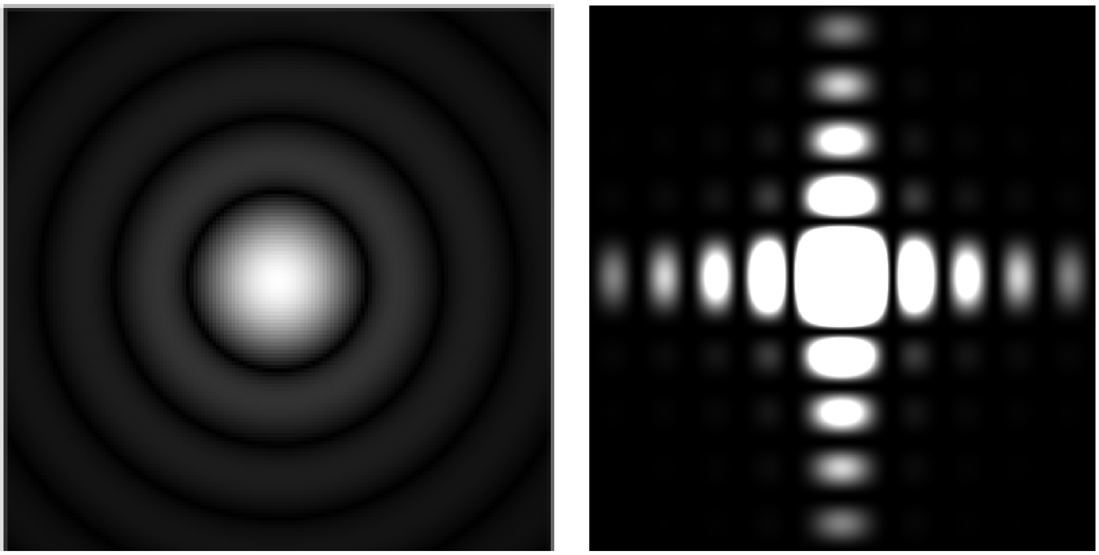
For other planes that cut through the cone of light but do not pass through $\mathbf{r}'$, the response is spread over the surface, thus leading to the idea of point spread function (PSF). For any such surface, one could consider the infinitesimally small patches comprising the surface, and calculate the (ir)radiance received on each patch. There is usually no closed form solution for such problems, and numerical methods such as ray tracing are required to calculate the response at such non-focal measurement planes. Common software solutions are also capable of taking optical aberrations such as defocus, tilt, spherical aberration, astigmatism, coma, distortion, petzval field curvature, and chromatic aberrations into account.

The second mechanism that leads to a spatially-spread response is due to the wave nature of light. A point response at the focal plane, as discussed in the previous paragraphs, is not attainable due to the wave nature of the light and how light gets diffracted from the edges of the optics entrance pupil. Although less accurate than a full-blown EM analysis, the Huygens–Fresnel principle proposes that every point to which a luminous disturbance reaches becomes a source of a spherical wave; the sum of these secondary waves determines the form of the wave at any subsequent time. The concept of wave front and phase plane introduced by the Huygens–Frensel principle can be utilized to calculate the interference patterns and point-spread functions caused by the wave nature of light in optical systems. For the optical systems used in consumer imaging solutions, the Franhaufer approximation of diffraction is highly accurate and may facilitate calculating the wave interference patterns on the measurement plane. Unlike geometrical optics where modeling the dependence on light frequency is optional, the results of physical optics inherently depend on the frequency of light as it directly affects the wavefront interference. Fig. 2.3 illustrates the point spread function resulting from a circular and a rectangular aperture and formed on the focal plane of a perfect lens.

In summary, the two-stage irradiance-based method combines the geometric optics results with the physical optics using the following procedure:

1. The description of the scene and the measurement surface are available as inputs. The measurement surface $\mathcal{M}$ is generally parameterized by two continuous coordinates (r, s) in a given domain $\mathcal{R} \times \mathcal{S}$ and the surface is described by $(x(r, s), y(r, s), z(r, s))$.

2. For each patch p on the scene, geometrical optics and ray tracing is used to find a center-of-mass $\mathbf{r}(r_p, s_p)$ for the intersection of rays with the measurement surface



(a) The point spread function (PSF) of a circular aperture at the focal plane of a perfect lens

(b) The point spread function (PSF) of a rectangular aperture at the focal plane of a perfect lens

Figure 2.3: Point spread functions for circular and rectangular aperture

$\mathcal{M}$. Geometrical distortions may be taken into account at this step by modeling the deviation of the actual intersection point from the expected ideal intersection point. The irradiance received at the optics entrance pupil is assumed to be concentrated at this center of mass. Using geometric optics or ray tracing techniques, it is also possible to augment the received irradiance with a single direction of arrival to be used later with sensor calculations.

3. Depending on the material used to fabricate a lens, the lens might let different light frequencies through preferentially. If the transmittance of the lens is available as a spectral curve, it may also be applied to the irradiance information at this point. Note that this information may also be available through the commercial software solutions that will be discussed shortly. In this case, the transmittance information is factored into the point-spread-functions returned by the software.

4. The ray-tracing and geometrical optics may also provide a smearing kernel or point-spread function $K_{\text{geo}}(\mathbf{r} - \mathbf{r}(r_p, s_p))$, for any location $\mathbf{r}$ on $\mathcal{M}$ and w.r.t to the center-of-

mass, caused by optical aberrations present in the optical system.

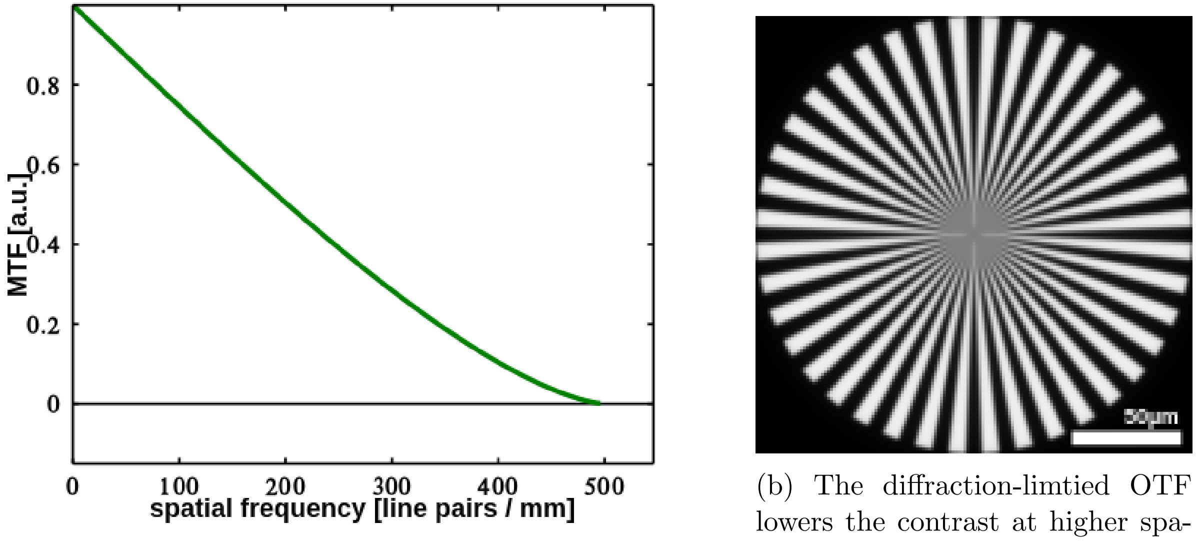
5. For each patch on the scene, the smearing kernel $K_{\text{phys}}(\mathbf{r}-\mathbf{r}(r_p, s_p))$, around the center-of-mass and over the measurement surface, caused by physical optics is calculated using analytical methods (such as Franhaufer or Frensel approximations, or Fourier optics), wavefront analysis (using Zernike polynomials for example), or software packages such as Zemax.

6. Given the smearing kernels at each point, the concentrated point-like irradiances received at the measurement plane are convolved with the kernels to apply the smearing effect. Upon completion, the irradiance will be available for each point on the measurement plane.

In the last step, if the point-spread functions (PSFs) are shift-invariant, or in other words if the shape of the PSFs do not depend on the scene patch location, the convolution of the received point-like irradiance received at the measuement plane with the PSFs may be efficiently implemented with the Fast Fourier Transform. Particularly, the 2-dimensional FFT of the received irradiance is multipled by the Fourier-Transform of the PSF, namely the Optical Transfer Function (OTF), and then the result is conevrted back to the original domain by the IFFT.

As an alternative to PSF specification, the optical transfer function (OTF) may be used to describe the optical system. The OTF describes how different spatial frequencies over the measurement plane will be affected by the smearing of the kernels. In general, the OTF is 2-dimensional since it is the FFT of the 2-dimensional PSF. However, for an ideal diffraction-limited lens, the OTF only depends on the magnitude of the spatial frequency and may be expressed as a 1D curve. Over a plane parallel to the lens, the OTF of a diffraction-limited optical system is given by:

$$OTF(\rho, \lambda) = \frac{2}{pi}\left[\text{acos}\frac{\rho}{\rho_c} - \frac{\rho}{\rho_c}\sqrt{1-\left(\frac{\rho}{\rho_c}\right)^2}\right] \tag{2.31}$$



(a) The diffraction-limited OTF with $\rho_c = 500$ line pairs/mm

(b) The diffraction-limtied OTF lowers the contrast at higher spatial frequencies by acting as a low-pass spatial filter.

Figure 2.4: The diffraction-limtied OTF and its effect on a Spokes diagram

where ρ is the spatial frequency, and ρ_c is called the cut-off spatial frequency and defined as $\rho_c = D/(\lambda d)$, where D is the aperture diameter, λ is the wavelength of light, and d is the distance of the imaging plane from the lens. Fig. 2.4 shows the diffraction limited OTF and how it affects a Spokes diagram by reducing the contrast at higher spatial frequencies.

2.3.2 Shift-Invariant Optics

Shift-invariant (SI) optics are a generalization of the diffraction-limited optics where the lens imperfections are factored into the PSFs. But importantly, it is still assumed that the shape of the PSFs does not depend on the location of the scene patch.

From a geometrical optics perspective, the lens system imperfections may be studied with ray tracing and spot diagrams. As an example, consider the lens system in Fig 2.5. that suffers from spherical aberration. Tracing the intersection of rays (at a given wavelength) with any imaging plane results in the spot diagrams. For most imaging systems, the breadth of the smearing caused by this geometric aberrations is larger than the diffraction-limited PSF breadth. For such systems, it suffices to ignore the physical optics effects and construct

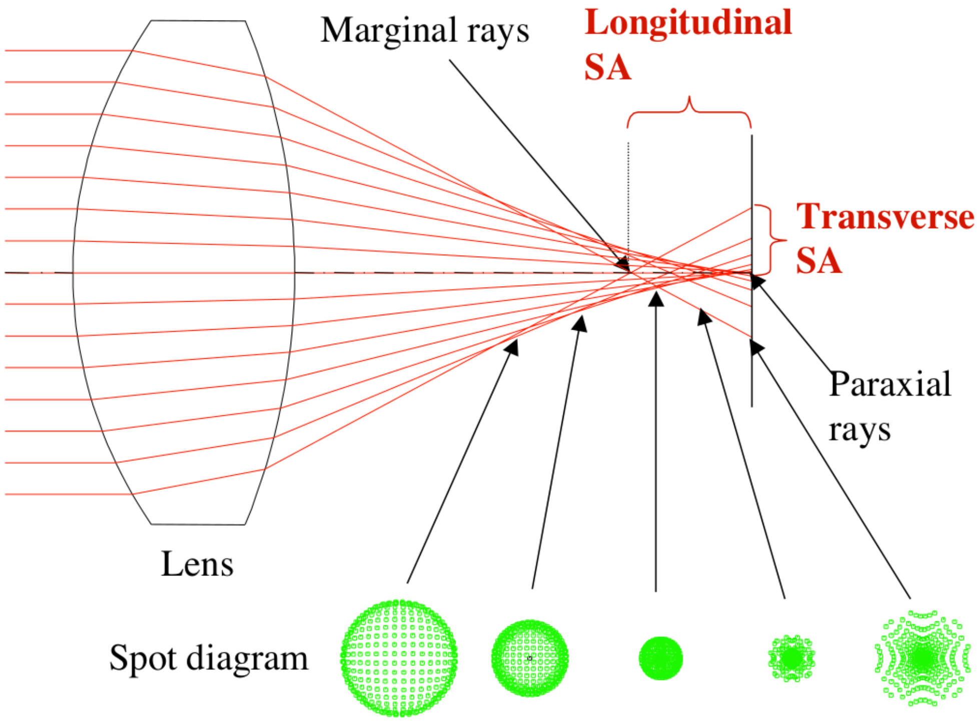


Figure 2.5: Spot diagrams across different measurement planes for a lens with spherical aberration (SA).

the PSF by sampling the spatial density of the spot diagrams.

In reality, however, the spot diagrams are only a first order approximation since they ignore the wave nature of light. In practice, performing a full-blown EM analysis or accurate physical optics analysis of the problem is either computationally expensive or analytically intractable. In the latter case, numerical methods and software solutions are used to find the PSFs over a given measurement plane. In the former case, certain approximate techniques in physical optics may be used to replace or augment the results of geometric optics. Regardless of the method used, the final result is a set of shift-invariant PSFs per wavelength of interest.

One of the most powerful analytical approximations to physical optics is based on the Fourier optics or scalar diffraction theory. Fourier optics is built upon the idea that the wave nature of light is modeled by the Huygens-Fresnel principle, that every point on a wavefront of light is the source of new spherical waves. The summed interaction of these spherical waves at some observation plane determines what is observed at that plane. The polarization of the electromagnetic field is neglected, and consequently light can simply be modeled by a wave function that obeys the standard wave equation and represents an electric or magnetic field. Hence, Fourier optics is also called scalar diffraction theory.

The mathematical foundation of Fourier optics is the concept of pupil function. At a given wavelength, the wavefront at the entrance pupil of the optics is given by a complex function:

$$P(r,s) = A(r,s)e^{jW(r,s)} \tag{2.32}$$

where A is a real-valued amplitude, W is a real-valued phase, and the coordinates (r,s) parameterize the entrance surface. In Fourier optics, an optical system is characterized by a transmittance function which maps the optic's entrance pupil function to the optic's exit pupil function:

$$T : P(r,s) \rightarrow P(u,v) \tag{2.33}$$

where the coordinates (u,v) parameterize the exit pupil surface. The transmittance function may be found through analytical approximations, ray tracing techniques, or numerical wavefront simulations. In general, the irradiance received at coordinate (z,w) of the measurement plane is found, up to a scale, by the integral:

$$I(z,w) \propto \int P(u,v)\frac{e^{-jk|\mathbf{r}(z,w)-\mathbf{r}(u,v)|}}{|\mathbf{r}(z,w)-\mathbf{r}(u,v)|^2}dA(u,v) \tag{2.34}$$

where k is the wave number at the given wavelength. Without further assumptions, this integral should be calculated numerically. In practice, however, certain assumptions may be added to implement the integral efficiently. As an example, if the exit pupil plane is parallel to the measurement plane, and if the problem dimensions satisfy the Franhaufer conditions, the integral may be written as:

$$I(z,w) \propto \int\int dudvP(u,v)e^{-jk(zu+vw)} \tag{2.35}$$

which is the Fourier Transform of the pupil function $P(u,v)$ and may be implemented efficiently using the FFT. Care must be taken to sample $P(u,v)$ finely enough to avoid aliasing; the phase of this function must be slowly varying at the chosen sampling rate. In addition, care must be taken to pad $P(u,v)$ with enough zero values to avoid the ringing effects associated with the periodic boundary conditions of the discrete Fourier transform.

The utility of pupil functions results from the fact that many wavefront aberrations are modeled by phase variations in the pupil function. In order to conveniently describe the different types of aberrations that occur, the Optical Society [5] has developed a standard convention [6] for circular pupils based upon the complete basis set of Zernike polynomials. These polynomials are orthogonal over the unit circle and are indexed either by a pair of numbers (n,m) or by their overall mode number j. The radial order n specifies the power of the radial portion of the polynomial, while the angular frequency m specifies the number of angular modes. For each radial order $n \geq 0$, there are $n+1$ polynomials forming a pyramid structure as shown in Fig. 2.6. To make indexing more convenient, a single index of the form $j = [n(n+2)+m]/2$ may be used. Table 2.1 lists the relationship between the most common aberrations and the corresponding Zernike polynomials. The PSFs resulting from some of the aberrations are illustrated in Fig. 2.7.

2.3.3 Shit-Variant Optics

In reality, no optical system is shift-invariant. Technically speaking, for any measurement plane $\mathcal{M}$ parameterized by coordinates (r,s), the shape of the PSF due to a scene patch p located at $\mathbf{r}_p$ depends on $\mathbf{r}_p$ (unlike the shift-invariant optics). For shift-invariant optics, the PSF due to the patch p could be written as $K(\mathbf{r}-\mathbf{r}'_p)$ with $\mathbf{r} \in \mathcal{M}$ and $\mathbf{r}'_p$ is the center-of-mass

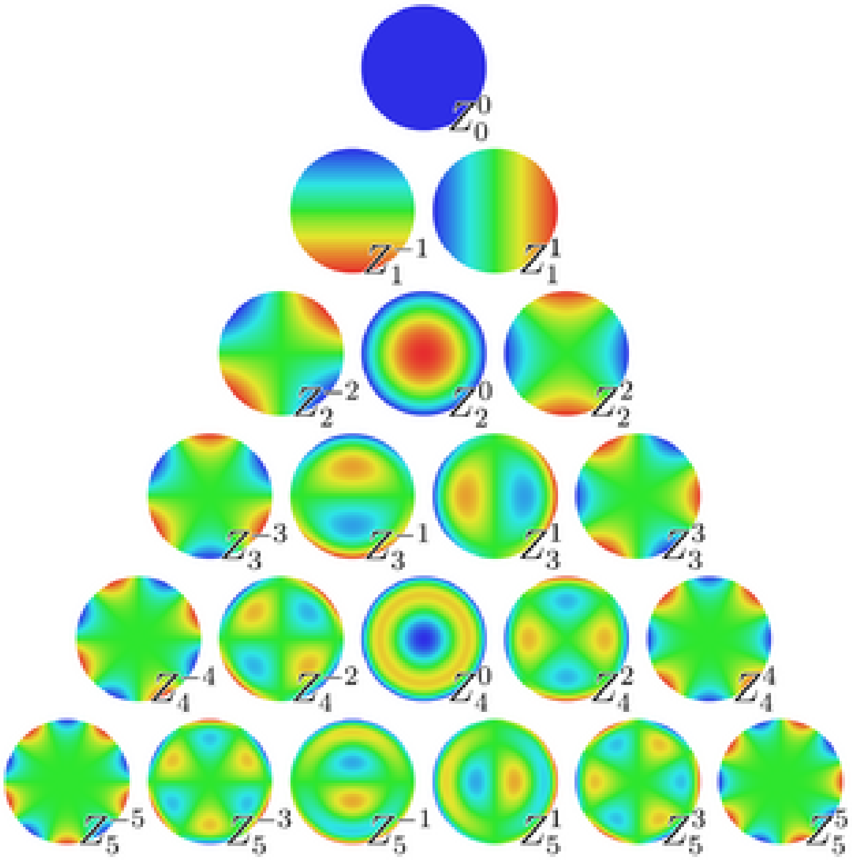


Figure 2.6: The pyramidal structure of Zernike polynomials

Zernike Polynomial	Aberration Name
$Z_0(\rho,\theta) = 1$	Piston
$Z_1(\rho,\theta) = 2\rho\sin\theta$	y tilt
$Z_2(\rho,\theta) = 2\rho\cos\theta$	x tilt
$Z_3(\rho,\theta) = \sqrt{6}\rho^2\sin 2\theta$	Primary astigmatism at 45°
$Z_4(\rho,\theta) = \sqrt{3}\,(2\rho^2 - 1)$	Defocus
$Z_5(\rho,\theta) = \sqrt{6}\rho^2\cos 2\theta$	Primary astigmatism at 0°
$Z_7(\rho,\theta) = \sqrt{8}\,(3\rho^3 - 2\rho)\sin\theta$	Primary y coma
$Z_8(\rho,\theta) = \sqrt{8}\,(3\rho^3 - 2\rho)\cos\theta$	Primary x coma
$Z_{12}(\rho,\theta) = \sqrt{5}\,(6\rho^4 - 6\rho^2 + 1)$	Primary spherical

Table 2.1: A subset of Zernike polynomials and the associated aberrations. ρ and θ are the polar coordinates over the optics circular exit pupil

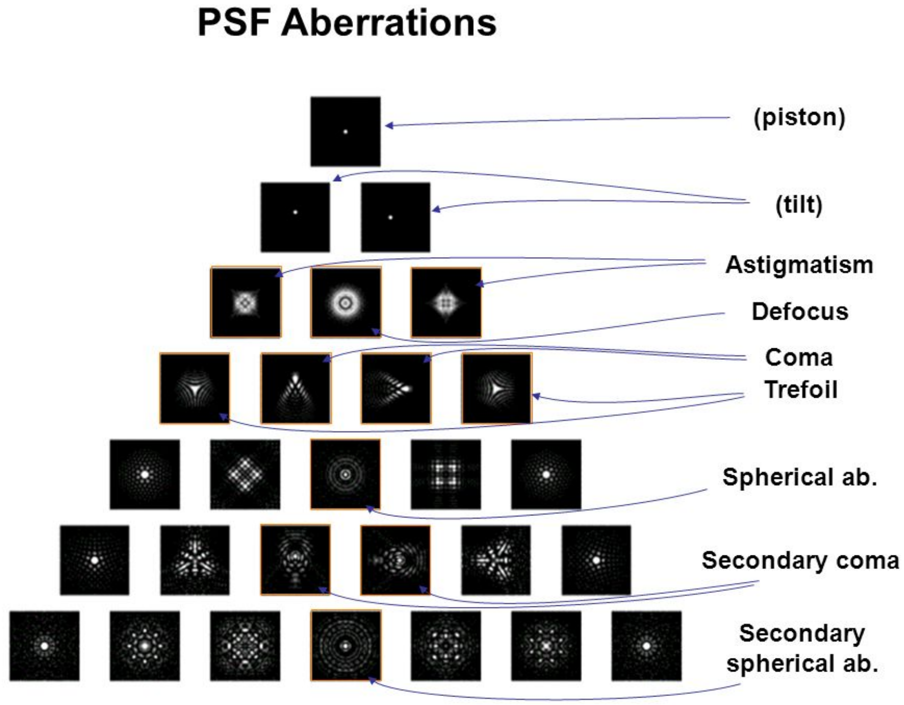


Figure 2.7: The PSFs resulting from some of the aberrations. The PSF of piston aberration is the airy disk pattern, not a point.

of patch p projected onto $\mathcal{M}$. Importantly, the position of the patch only affects the center of the PSF but not its shape. On the other hand, the shape of the PSF of a shift variant optical system depends on the source location as well and is in general written as $K(\mathbf{r}, \mathbf{r}_p)$. Fig. 2.8 illustrates an example of the spatially-varying PSFs of a lens suffering from exaggerated coma aberration obtained by ray tracing and spot diagrams. More realistic results may be obtained using physical optics at the cost of higher computational complexity.

For shift-variant optics, the steps required to calculate the irradiance received at the measurement plane are very similar to the steps used for shift-invariant optics. Fig 2.9 shows an example of convolving the image of a 2D array of light sources with a hypothetical set of space-varying PSFs. Note that since the PSF varies with space, the convolution of the unblurred irradiance with the PSFs cannot be efficiently implemented using FFT. In later chapters, we discuss several methods used in the literature to mitigate the computational complexity of the basic implementation of the convolution in the original domain. We will

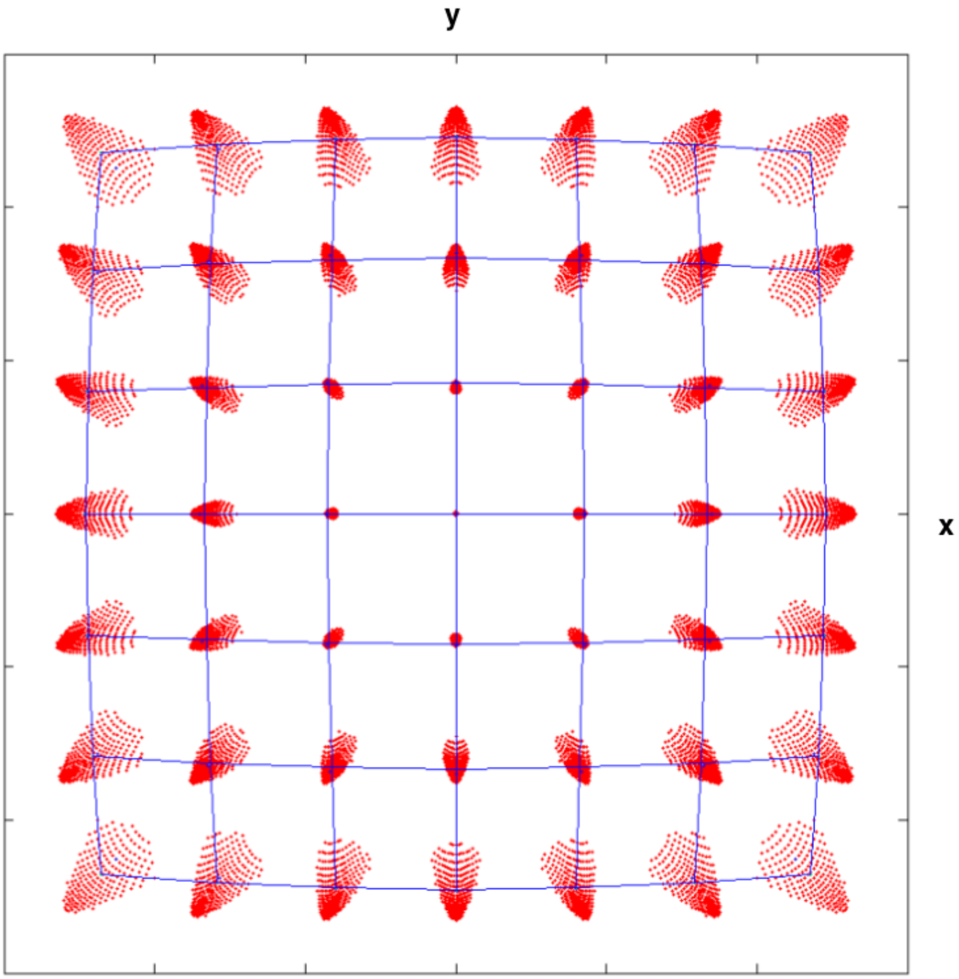


Figure 2.8: The space-varying PSF of an imperfect lens with exaggerated coma aberration. The spot diagrams are found by ray tracing and ignoring physical optics. More accurate and realistic results may be obtained by physical optics techniques such as wave front analysis.

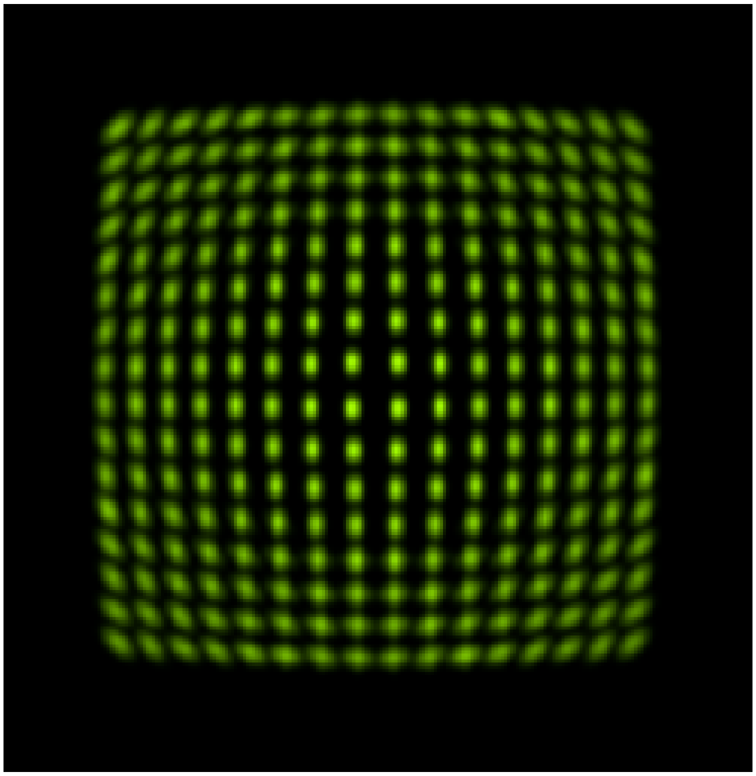


Figure 2.9: Irradiance of a 2D array of light sources on the focal plane of an imperfect hypothetical lens. The effect of geometrical distortion, here a barrel distortion, has also been included in projecting the point source onto its center of mass on the focal plane.

also propose a new method for implementing the convolution with much lower computational complexity and will compare the results of our method with previous work.

2.4 Practical Considerations for the Optics

The first practical aspect in modeling the optics is how the radiance due to a given scene patch at the optics entrance pupil is transformed to the radiance at the exit pupil.

A common and computationally inexpensive implementation includes propagating geometric rays through the optical system and keeping track of the optical distance travelled by each ray. The optical distance may then be converted into a phase shift of the ray at its intersection with the exit plane. The density and phase distribution of the received rays may be interpolated over the exit pupil to calculate the radiance at the exit pupil. Although this method is computationally inexpensive, it is not capable of fully accounting for the wave nature of the light as it ignores the Huygen's principle.

More complicated but accurate results may be obtained by propagating the pupil function along the optical system through wavefront analysis. Although closed form results exits for thin lenses, numerical methods based on the Fresnel diffraction integral should be applied for general non-thin compound lens systems [7, 8]. Similar numerical approaches may then be used to calculate the irradiance at an arbitrary measurement plane from the exit pupil's pupil function.

Storing the space-varying PSFs may also impose a high spatial footprint. For typical optical systems where the subject is far from the optics, the PSF may be obtained per elevation and azimuth rather than per point in the space, thus reducing the number of dimensions of the required hypercube from 5 to 4. Furthermore, most optical system have a rotational symmetry. For such systems, the variation of the PSF with azimuth may be removed as well, leading to a 3 dimensional data cube required for storing PSF data. However, the PSF should be rotated appropriately at the convolution step unless the measurement surface is specified in some sort of polar coordinate system. Finally, most PSFs vary smoothly with space. The smooth variation of the PSFs may be exploited to project the set of PSFs into a lower dimensional space (using PCA for example), leading to a significantly lower storage footprint for the PSFs.

Another practical aspect is the discretization of the measurement plane. First, the patches of the measurement plane should be small enough to resolve the fine features of the PSFs and avoid aliasing. Second, the sample points should ideally match the sensor locations in order to avoid interpolating the PSF values at sensor locations from the PSFs over the misaligned points on the measurement surface grid.

The most critical computational bottleneck is convolving the scene radiance with the space-variant PSFs. In the next chapter, we discuss the computational complexity of such a convolution and review some of the previous work on the efficient calculation of such

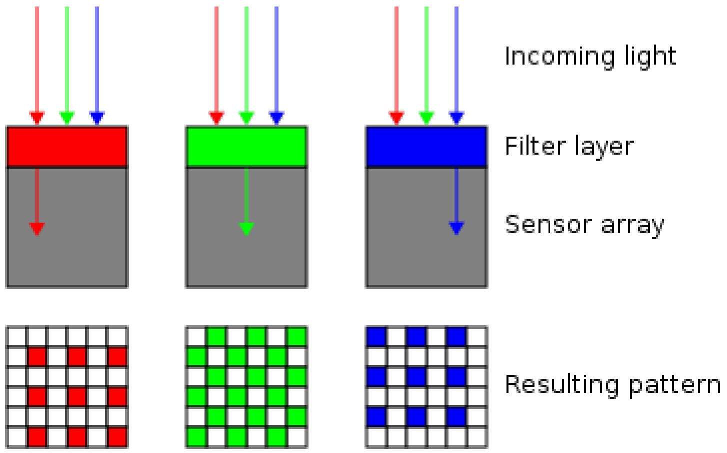
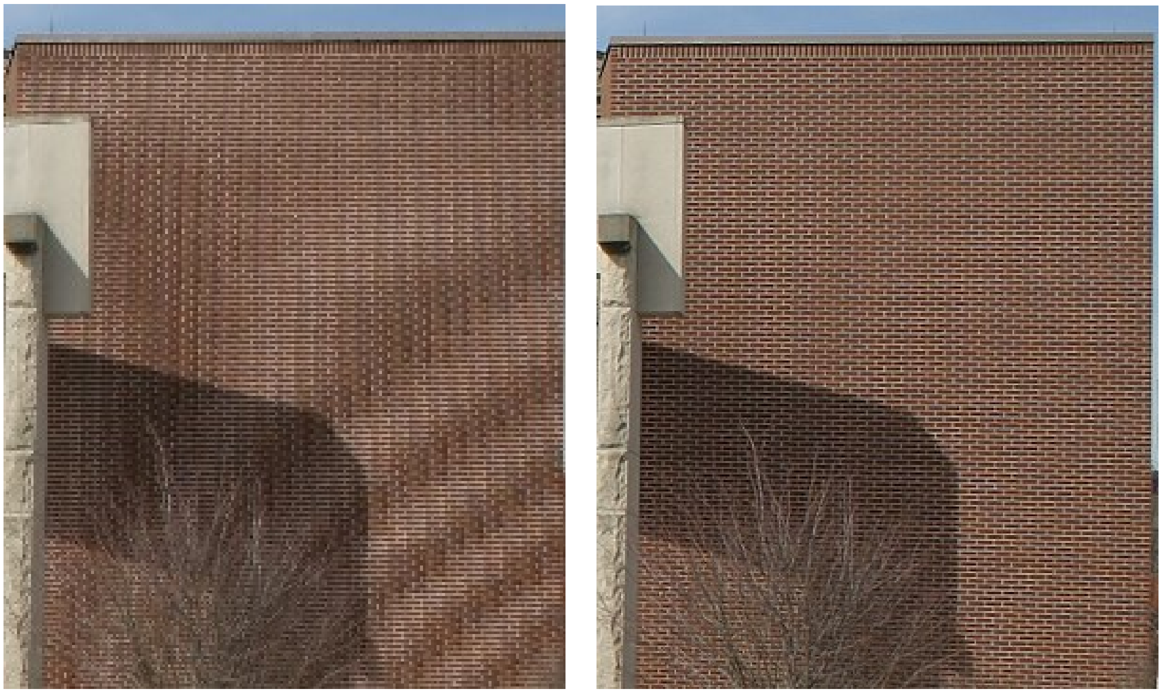


Figure 2.10: The Bayer pattern of RGB sensors. Each block consists of two green sensors, one blue sensor, and one red sensor.

a convolution. We also propose a new approach based on Principal Component Analysis (PCA) to efficiently calculate the convolution.

2.5 Sensor Modelling

The sensors in an imaging system are responsible for electrically storing information about the incoming light. The sensors are typically arranged on a 2D surface, which naturally coincides with the measurement surface discussed in 2.3. Due to fundamental limits imposed by the physics of the materials comprising the sensors, the sensors are often inhomogeneous and each store a different piece of information about the received light. A common example is the RGB sensor system which consists of three different types of sensor where each sensor is most sensitive to the wavelengths around red, green, and blue light respectively. The information from different sensors may then be combined to make inference about the structure of the received light. Fig. 2.10 shows a common pattern of RGB sensors known as a Bayer pattern. The wavelength sensitivity of the sensors is implemented by adding optical filters at the entrance of the sensor; the light entering a sensor is filtered with a color filter array to reduce the noise inflicted by the irrelevant wavelengths on the sensor.



(a) Spatial aliasing in the form of a moir pattern. Spatial frequencies close to the feature size of sensor blocks are aliased as spurious low-frequency components.

(b) The anti-aliasing filter mitigates the aliasing by removing high spatial frequencies before being fed into the inhomogeneous sensor array.

Figure 2.11: Aliasing caused by the spatial arrangement of inhomogeneous sensors in the sensor array.

Due to the geometrical arrangement of the inhomogeneous sensors on the measurement plane, certain high spatial-frequency components of the image may be aliased, leading to artifacts such as the Moire pattern shown in Fig. 2.11. To mitigate the aliasing, most sensor arrays are overlaid with an optical diffuser layer which serves as an anti-aliasing filter by smoothing the received irradiance in order to feed the same optical information to the adjacent sensors. The effect of a diffuser might be modelled together with the optics by adding an extra layer of convolution or by combining the diffuser response with the optics response.

The most common technologies used for fabricating sensors in consumer imaging products are semiconductor charge-coupled devices (CCD), active pixel sensors in complementary metal–oxide–semiconductor (CMOS), and N-type metal–oxide–semiconductor (NMOS).

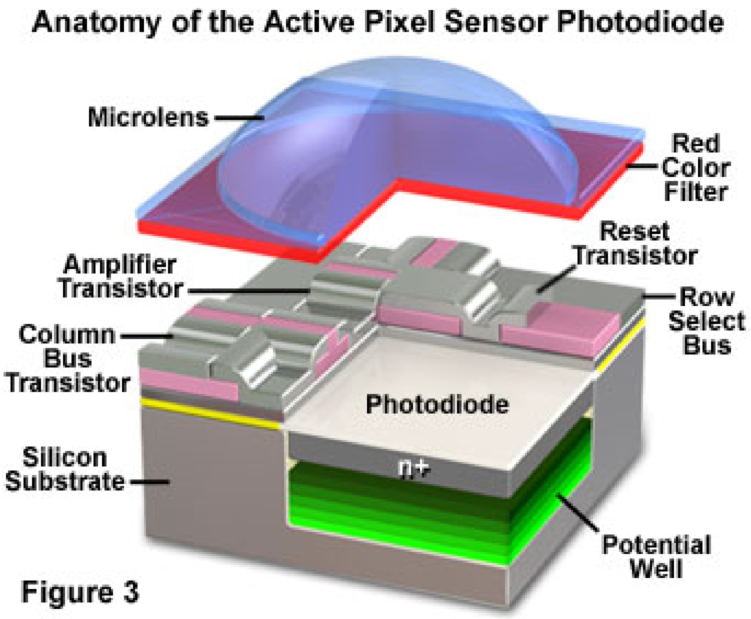


Figure 2.12: Physical structure of a CMOS sensor cell

Most small consumer cameras use a CMOS sensor, because most CMOS sensors are cheaper and smaller than CCDs and have a lower power consumption which is critical for handheld and battery-powered devices.

Different sensor technologies use the same fundamental mechanism to sense the light. As an example, consider a typical structure of a CMOS sensor cell in Fig. 2.12. Each cell mainly consists of a photodiode and a supporting structure. The photodiode is the heart of the sensor where the energy of photons of light are gathered and used to generate electron-hole pairs in the p-n junction. The supporting circuitry is responsible for reading and resetting the recorded signal. Due to the presence of the supporting structures, only a fraction of the cell area may be covered by the photodiode. To mitigate this issue, a microlens is necessary to gather the light over the entire entrance window of the cell and focus the energy on the photodiode to maximially exploit the received light. Without a microlens, one would have to either shrink the area of the entrance window, resulting in lower received energy, or to deal with the light diffractions caused by the light impinging on the supporting structures.

The light sensing mechanism of a p-n junction is illustrated in Fig. 2.13. Each photon of

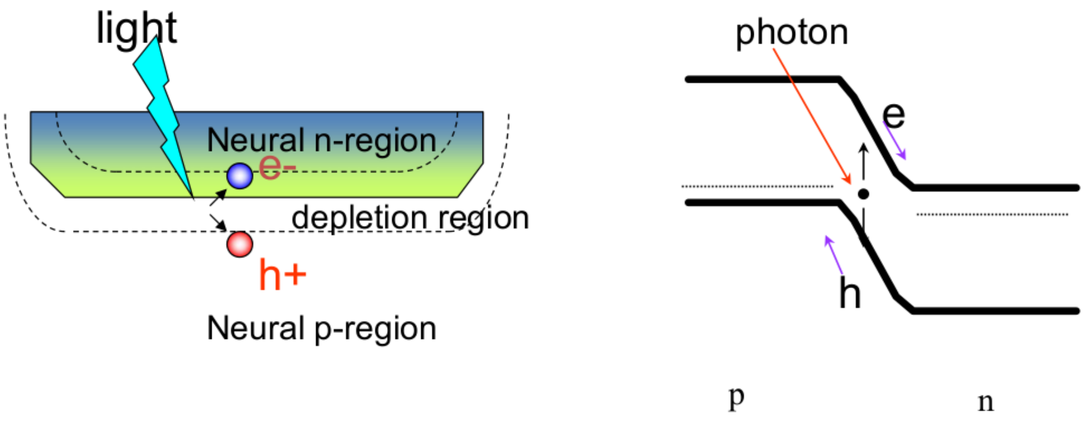


Figure 2.13: The photo sensing mechanism in a p-n junction

frequency ν carries an energy equal to $h\nu$ Jules, where h is the universal Planck's constant and is equal to $6.626 * 10^{-34}$ J.sec. When a photon of sufficient energy strikes the diode, it creates an electron-hole pair. If the absorption of the photon occurs in the p-n junction's depletion region, or one diffusion length away from it, these carriers are swept from the junction by the built-in electric field of the depletion region. Thus holes move toward the anode, and electrons toward the cathode, and a photocurrent is produced. The total current through the photodiode is the sum of the dark current (current that is generated in the absence of light) and the photocurrent.

The electron-generation capability of a sensor is most commonly specified quantitatively with quantum efficiency η, defined as the ratio of the number of carriers collected by the sensor to the number of photons of a given energy incident on the sensor. The main three factors that affect the quantum efficiency are:

1. Reflection at the surface of the photo-diode, denoted by R. This may be minimized by using appropriate coating.

2. The fraction of generated electron or holes that do not get recombined before being collected, denoted by ξ.

3. The fraction of photons that are actually absorbed by the silicon bulk and do not pass

through. The thicker the silicon bulk, the higher will be this fraction.

Therefore, quantum efficiency may be written as:

$$\eta = (1 - R)\xi(1 - e^{-\alpha d}) \tag{2.36}$$

where d is the thickness of the silicon bulk, and α is the absorption factor with a unit of m^{-1}.

In the simulator, the following steps are taken to convert the received (ir)radiance to the sensor recordings:

1. The received radiance, or the direction-augmented irradiance, should pass through the microlens if one is available. The effect of the microlens on the received (ir)radiance may be quantified using the Wigner phase space as explained later. Furthermore, if the microlens material preferentially passes certain wavelengths, the lens spectral transmittance should also be applied.

2. For many sensor technologies, the light gathered by the microlens should pass through multiple stacks of different materials, with different thicknesses and different refractive indices, before reaching the photo-sensitive diode. The transmission of the photons through the stacks may also be modelled with the Wigner phase space. Furthermore, both sensor vignetting and etendue may be modelled naturally by the Wigner representation.

3. Once the irradiance at the photo-sensitive diode is calculated, the irradiance should be first interpreted as the spectral influx of photons (number of photons hitting the diode per unit time per unit wavelength) by dividing the irradiance by the photon energy $h\nu$. The photon spectral influx is then multipled by the quantum efficiency curve of the photo-diode to obtain the number of electrons collected by the cell per unit time per unit wavelength. Finally, the electron count should be summed over all wavelengths to

obtain the number of electrons generated per unit time. Using the elementary charge $1.602176634 \times 10^{19}$ Coulombs, the electron generation rate is converted to a baseline current.

4. The current generated by the electrons is then integrated over time to calculate the voltage in the same way that the current into a capacitor is converted to voltage. However, the well-capacity of a sensor, defined as the total number of electrons that the sensor can accumulate, clips the integration at some point. At the end of this stage, a nominal noise-free voltage is obtained.

5. Finally, other electrical and noise effects may be added to the noise-free voltage. The most common noises are shot noise, dark noise, read noise, fixed pattern noise (FPN) including dark signal non-uniformity (DSNU) and photo response non-uniformity (PRNU), and column fixed pattern noise (CFPN). The electrical effects include the addition of analog voltage offset and voltage gain, clipping to the swing voltage, and ADC quantization.

The details of each individual stage is described in the following subsections.

2.5.1 Wigner Phase Space Representation

Wigner phase space is a powerful representation for describing the transmission of geometrical rays through optical systems. To illustrate the concept, consider a truncated piece of line as shown in Fig. 2.14 and a ray of light impinging with an angle of θ. The Wigner phase space representation, $W(x, \theta)$, represents the density of the rays of lights that fall on a point with coordinate x from the angle θ. The Wigner phase space is simply a representation of the concept of radiance. Due to the conservation of the numerical aperture as light propagates through optical systems, it is more common to formulate the Wigner phase space representation in terms of $p = n \sin\theta$ rather than θ. The important advantage of $W(x, p)$ compared to

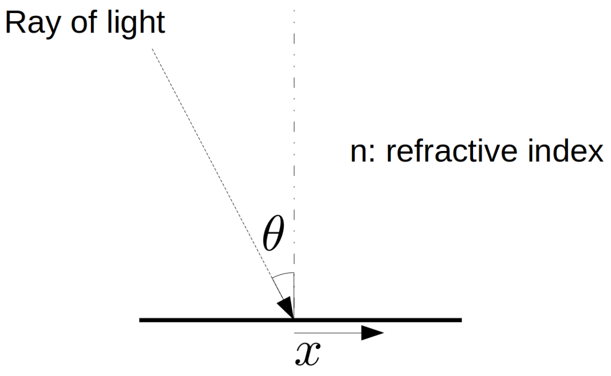


Figure 2.14: Illustration of Wigner Phase Space with a 2D example.

$W(x, \theta)$ is that the representation remains unchanged as light propagates through multiple media with different refractive indices.

For a thin lens, it can be shown that the $W_{out}(x, p)$ at the output of the lens is related to the input $W_{in}(x, p)$ by the following transformation:

$$W_{in}(x, p) = W_{out}(x, p + \sin(\text{atan}\frac{x}{f})) \tag{2.37}$$

given that the incident ray of light is within the field of view of the lens (otherwise, the ray will not be transmitted at all). If the rays of light are close to the optical axis (paraxial regime), the term $\sin(\text{atan}\frac{x}{f})$ may be approximated by $\frac{x}{f}$).

The free unobstructed propagation of the Wigner phase space in any medium up to a depth of d may be written as:

$$W_{in}(x, p) = W_{out}(x - d\tan(\arcsin\frac{p}{n2\pi/\lambda}), p) \tag{2.38}$$

where λ is the wavelength of the light. In the paraxial regime, $d\tan(\arcsin\frac{p}{n2\pi/\lambda}) \approx dp/(n2\pi/\lambda)$.

After performing all necessary transformations, the Wigner phase space at the surface of the photo-diode is integrated over both x and p to obtain the irradiance. Although the provided formulas are only applicable to a 1D problem, the concept may easily be extended to 2D problems as well.

2.5.2 The Photo-Sensitive Cell

Given the irradiance at the photo-diode surface, $I(\lambda)$, the corresponding photon influx $\phi_{ph}(\lambda)$ is given by $\phi_{ph} = I/(h\nu)$ where $\nu = c/\lambda$. Multiplying the photon influx by the quantum efficiency results in the electron-collection rate for each light wavelength. However, the recorded voltage will depend on the aggregate effect of all wavelengths. Thus:

$$\phi_{el} = \int d\lambda \eta(\lambda)\phi_{ph}(\lambda) \quad \text{[electron/second]} \tag{2.39}$$

The effect of a color filter at the entrance of the sensor may be either implemented separately before the Wigner space calculations or could be absorbed into the quantum efficiency term.

The electron influx is then converted to an electrical current i using $i = \phi_{el}q$, where q is the elementary charge in Coulombs. For a linear sensor, the current is then integrated over the exposure time T to find the noise-free voltage:

$$v = \int_0^T dtCi = CiT \tag{2.40}$$

For non-linear sensors, the instantaneous capacity C will depend on the instantaneous voltage v as well. As mentioned before, the result of the integral is limited from above by the sensor's well capacity.

2.5.3 Sensor Noise

The most common sensor noises are shot noise, dark noise, read noise, fixed pattern noise (FPN) including dark signal non-uniformity (DSNU) and photo response non-uniformity (PRNU), and column fixed pattern noise (CFPN).

Shot noise exist because phenomena such as light and electric current consist of the movement of quantized packets. The fundamental physical processes that govern light emission or electron-hole pair generation are inherently random. The shot noise is modelled as a Poisson random variable. The standard deviation of the Poisson random variable is the same as its mean, and the mean may be estimated as the number of electrons comprising the perfect voltage v. For a large mean number of electrons, the Poisson random variable may be be approximated as the mean x plus a normally distributed varation as $\mathcal{N}(0, x^2)$. The dependence of the shot noise on the nominal ideal voltage v is expressed by $n_{\text{shot}}(v)$.

The dark noise results from the random generation and collection of electron-hole pairs in the silicon bulk even in the absence of light. The dark current is mainly due to the random thermal oscillation of the particles. Typical sensor specifications provide a dark current density in Ampere/m^2, which may be converted to a dark current through a multiplication by the area of the photodiode and then integrated in time to find the dark noise voltage. The dependence of the dark noise on the exposure time is denoted by $n_{\text{dark}}(T)$.

The read noise happens at the time of measuring the accumulated voltage and is a consequence of the imperfect operation of physical electronic devices. It is typically modelled as a normal random variable with zero mean and a circuitry-dependent standard deviation.

Fixed pattern noise, or FPN, is a general term that identifies a temporally constant sensor-to-sensor non-uniformity. This problem arises from small differences in the individual responsitivity of the sensor array that might be caused by variations in the pixel size, material or interference with the local circuitry. FPN usually refers to two parameters: One is the DSNU (dark signal non-uniformity), which is the offset from the average across the imaging array at a particular setting (temperature, integration time) but no external illumination, and the PRNU (photo response non-uniformity), which describes the gain or ratio between optical power on a sensor versus the electrical signal output. FPN may be modelled by the following equation:

$$(1 + n_{\text{PRNU}})v + n_{\text{DSNU}} \tag{2.41}$$

where n_{PRNU} and n_{DSNU} are both zero mean normal random variables with variances σ^2_{PRNU} and σ^2_{DSNU} respectively. PRNU is typically specified as a percentage, and DSNU is specified in Volts.

Column FPN is due to the variation in the column amplifier parameters, e.g. offset voltage, feedback capacitor value, reset transistor threshold voltage , and overlap capacitance value. CFPN is formulated similar to FPN, but the sensors in the same column will share the CFPN together.

In summary, the overall voltage may be expressed as:

$$v + n_{\text{dark}}(T) + n_{\text{shot}}(v) + n_{\text{read}} + (1 + n_{\text{PRNU}})v + n_{\text{DSNU}} + (1 + n_{\text{CPRNU}})v + n_{\text{CDSNU}} \tag{2.42}$$

2.5.4 Electrical Circuitry

In modern cameras, the noisy voltage calculated in the previous section is offset and amplified before being fed into the ADC. The relevant parameters are analog offset, V_0, and analog gain, G. The resulting voltage is given by:

$$(v + V_0)/G \tag{2.43}$$

After the offset and amplification, the voltage is clipped to the swing voltage of the circuitry, and then quantized by the ADC.

2.6 Practical Considerations for the Sensors

Ideally, the calculation of the photon influx at the photo-diode surface requires a 2 dimensional Wigner space propagation from the sensor's entrance aperture to the photo-diode surface. Given the large number of sensors in a typical system, the Wigner space computations may become prohibitive.

For most optical systems where the chief ray angle from the center of the optics to the sensor is close to the paraxial regime, it is possible to perform a 1 dimensional Wigner space to calculate the fraction of power that reaches the photo-diode, and then square the fraction to approximate the fraction found by a 2 dimensional Wigner space propagation.

Chapter 3

Efficient Application of Spatially-Varying Point-Spread Functions

The highest computational cost in the processing chain belongs to the image formation by the optical system if a shift-variant optical system is used. For such systems, the image formation requires convolving the received radiance at the sensor surface with a shift-variant impulse response that depends on the location of the sensor.

In the following sections, the previous work on the efficient implementation of the space-varying kernels is presented and a new method based on PCA is suggested. The computational complexity and quality of the proposed method is compared with previous work. Through simulations over a data-set of 40 scenes, we have shown that the proposed method offers higher quality results compared to the previous work while requiring a time complexity that is only linear in the scene size.

3.1 Previous Work

Space Partition Methods: The simplest approximate solution to mitigate the computational complexity of space-varying (SV) PSFs is to partition the input space into isoplanatic patches and assumes that the PSF within each patch is spatially invariant. Then the FFT can be used to efficiently compute the result for each patch; Finally, the results are added together. The authors of [9], [10], and [11] use the overlap-add method which is essentially a partitioning method. The efficient filter flow approach [12] and its extension in [13] are also based on this method. Leveraging the rotational symmetry of the space-varying PSFs, the authors of [14] describe a low-cost algorithmic framework and the associated hardware enabling the space-varying FIR sharpening required to restore largely aberrated optical images. The rotational symmetry of the point spread function (PSF) about the optical axis allows computational savings.

Although very simple, the partitioning approach typically results in observable discontinuities over the partition edges. Although increasing the number of the partitions can mitigate this issue, it comes at a cost of higher computational complexity. Furthermore, the choice of partition count and shape is ad-hoc in the literature and does not follow any methodical principles. Finally, these strategies do not have a property that is highly desirable in numerical analysis: the ability to approximate the original operator with an arbitrary precision.

Subspace Methods: Another approach is to approximate the space-varying convolution as the weighted linear combination of a small number of convolutions with fixed PSF [15, 16]. In [16], smooth PSF variations are exploited to obtain the space-varying PSFs as an interpolation of multiple PSF samples. Other authors have also noticed that PSF variations could be modeled by interpolation and yet lead to a fast blurring model [17, 15, 12]. Similar approaches are followed in [18] and [19] by decomposing the space-varying PSF onto a lower-dimensional subspace of PSFs.

Matrix Source Coding: With a focus on the relatively large PSF of stray lights in optical systems, Wei et al. [20] view the convolution problem as a matrix multiplication problem. Denoting the vector of the input by x, the output of convolution, y is given by:

$$y = Ax + w \tag{3.1}$$

where w is additive noise, and $A = (1-\beta)I + \beta S$ with β being a small positive number ($\ll 1$) and S being a dense matrix representing the convolution with the spatially-varying stray light PSF. The authors focus on the computational complexity of calculating the product Sx. To speed up the calculation, a lossy transform coding of the rows and columns of S is proposed. The resulting coded matrix then becomes sparse after quantization, and this sparsity reduces both storage and computation. By looking into the MSE error of a generic lossy transform of S, the authors suggest that the input x, which is assumed to be a zero mean random vector, should be pre-whitened so that minimizing the MSE of the error in encoding S minimizes the error in the output vector as well, leading to the following transformation:

$$y = Sx = W_1^{-1}\hat{S}Tx \tag{3.2}$$

where W_1 is an orthonomal transform, and T is an invertible transform. If $\hat{S}$ is sparse, the computational complexity may be reduced significantly. The optimal situation is when $\hat{S}$, or a quantized version of it, is diagonal. The authors show that an optimal choice for W_1 and T exist and may be found through an eigenvalue decomposition of the covariance matrix of x. Since such an optimal choice typically leads to dense matrices for W_1 and T, the authors suggest a sub-optimal approach using the wavelet transform to approximate the row and column decorrelation performed by the optimal choice.

Sparse Approximation of Integral Operators: The authors of [21] apply the widely studied methods of sparsely approximating integral operators by viewing the convolution as an integral operator. The main contribution of this paper consists of showing that the integral operator can be replaced by a sparse approximation, and the replacement can lead to arbitrarily close approximations. More precisely, the authors show that $||H - \Phi S \Phi^*||$ can be made arbitrarily small, where H is the convolution matrix, Φ denotes a forward wavelet transform, and S is a sparse matrix. Similar approaches have been suggested in [22, 23, 24, 25].

Row-Column Decomposition of the spatially Varying PSF: Similar to [20], the authors of [26] view the convolution problem as a matrix multiplication problem. Under the assumption that the spatially varying PSFs may be decomposed into the product of a row-dependent PSF and a column-dependent PSF, the convolution matrix is written as the product of two lower rank matrices, thus lowering the computational complexity of applying the space-varying convolution.

Algorithms Tailored to Specific PSFs: If the convolution algorithm is not required to be applicable to any generic PSF, the features of the target space-varying PSF might be exploited to lower the computational burden. For example, Popkin et al. [27] propose a computationally efficient method tailored to Gaussian PSFs. Their method applies to smoothly space-variant Gaussian PSFs and uses a specialized filter bank optimized for Guassian PSFs. Tan et al. [28] have adapted an efficient recursive filter approximations of Gaussian blurring.

Hierarchical Techniques: Hierarchical techniques construct and employ a pyramid of blurred versions of the original image at differing levels of resolution and are highly efficient. The blended Gaussian Pyramid approach [29, 30] is an example of this method which performs a smoothly space-variant approximation of the Gaussian blurring.

Coordinate Transformation: Another technique is to apply a spatial coordinate transformation, such as log polar mapping [31, 32], and then apply space-invariant convolution with FFT followed by the reverse coordinate transformation. The coordinate transformation method are problem specific and finding a generic method for obtaining the appropriate transform is not possible.

3.2 The Proposed Method

The space-varying convolution method proposed in this thesis may be considered as an extension of the subspace methods discussed in the previous section. The major difference is that we follow a methodical approach toward finding the appropriate basis PSFs as discussed below.

Consider a simple test case of the space-varying convolution problem where the impulse response is piece-wise space-invariant. For instance, the PSF for the left half image may be different from the PSF for the right half image, but the PSF in each half image is space-invariant. Clearly, one might perform the convolution efficiently by using an FFT on each half plane and then combining the results together. Denoting the number of pixels in the image by N_P and the number of PSF grid points by N_K, the time complexity will be $O(N_p \log N_p)$. Compared to a complexity of $O(N_K N_p)$ for a direct convolution, and keeping in mind that $N_p > 400$ for typical problems, the FFT-based method will lead to a significant speed up on the order of $N_k / \log N_p$. Since $N_k \ll N_p$ for typical imaging problems, the speed-up may be further improved by using an overlap–add method to avoid calculating the FFT of the extensively zero–padded PSF.

In reality, the PSFs of space-varying optical systems are far from piece-wise linear and typically vary smoothly with space. As a result, the method discussed in the previous paragraph may not be directly applied to such problems. However, a deeper investigation reveals that a different perspective on the piece-wise space-invariant regime is applicable to the smoothly varying PSF systems. In particular, instead of extracting the partitions from the image and applying the FFT and then combining the results, the same results may be obtained by masking the image with two binary masks, one setting to zero the left half plane and the other one zeroing the right half plane. Finally, each of the two full-size masked images are filtered with an FFT and the results are simply added together. If an overlap-add methods is used, applying the FFT to areas of the image with negligible signal power may be skipped to lower the computational complexity.

Mathematically, denoting the input image by $X(x, y)$, the left- and right-plane PSFs by K_l and K_r, and the corresponding masks with M_l and M_r, the output image is written as:

$$\begin{aligned} Y(x,y) = &\int\int dudv X(u,v) M_l(u,v) K_l(x-u, y-v) \\ &+ \int\int dudv X(u,v) M_r(u,v) K_r(x-u, y-v) \end{aligned} \tag{3.3}$$

where the integral variables u and v run over the entire image plane.

To extend the idea to the general non-piece-wise space-varying kernel, note that the output image may also be refactored as:

$$Y(x,y) = \int\int dudv X(u,v) \Big[M_l(u,v) K_l(x-u, y-v) + M_r(u,v) K_r(x-u, y-v) \Big] \tag{3.4}$$

If the term inside the brackets is interpreted as a decomposition of the space varying kernel $K(u, v, x-u, y-v)$, we may suggest a general decomposition into the space of several space-invariant PSFs:

$$K(u, v, x-u, y-v) = \sum_{i=1}^{N_b} M_i(u,v) K_i(x-u, y-v) \tag{3.5}$$

where N_b is the number of space-invariant basis kernels used for the decomposition. Then, the output may be obtained by:

$$Y(x,y) = \int\int dudv X(u,v) \Big[\sum_{i=1}^{N_b} M_i(u,v) K_i(x-u, y-v) \Big] \tag{3.6}$$

which may be refactored as:

$$
\begin{aligned}
Y(x,y) =& \sum_{i=1}^{N_b} \int\int dudv \Big[X(u,v)M_i(u,v)\Big] K_i(x-u, y-v) \\
=& \sum_{i=1}^{N_b} \int\int dudv X_i(u,v) K_i(x-u, y-v)
\end{aligned}
\tag{3.7}
$$

where X_i is the i'th masked version of the input image obtained by multiplying the input image by the i'th mask K_i. Note that each double integral inside the summation is now a shift-invariant convolution which may be implemented efficiently with an FFT.

Ignoring the offline calculation of the PSF basis set $\{K_i\}$, the time complexity of the on-line calculation imposed by this scheme consists of two main parts:

1. Multiplying the image by the masks. This requires N_pN_b multiplications, leading to an order of $O(N_pN_b)$ assuming that multiplication is the computational bottleneck. Note that since the masks are pre-calculated, the areas of the mask with negligible values may be segmented and omitted from the multiplication, leading to a lower computational complexity at the cost of lower precision.

2. Performing N_b FFTs for each masked image. The 2D FFT consists of 2 1D FFTs; thus, the computational complexity of an individual FFT is $O(\sqrt{N_p}\log\sqrt{N_p})$, and the overall complexity is $O(N_b\sqrt{N_p}\log\sqrt{N_p})$. If overlap–add or overlap–save method is used to apply the convolution, the complexity may be further reduced.

The overall complexity is then $O(N_pN_b)$, which is linear in the image pixel count. Note that the multiplication by the masks is now more complicated than the filtering itself.

The efficiency of the algorithm boils down to how well the basis set can explain the space-varying PSF. In the context of this problem, given a fixed number of basis functions, a basis set A is intuitively said to be more efficient than another basis set B if it can approximate the space-varying PSF with a higher precision.

This suggest a constructive approach towards creating an optimum basis set: define the first element of the basis set such that it captures the highest possible variance in the observed space-varying PSF. Then, after removing the contribution of the first element, find the next element following the same principle: the second element should explain as much variance as possible in the residue of the space-varying PSF. The remaining elements follow the same procedure.

An implicit assumption of this scheme is that the effect of the first element can be perfectly removed from the space-varying PSF residue fed to the calculation of the next element. This requirement suggests that the elements of the basis set are orthogonal.

The collection of these requirements suggests that Principal Component Analysis (PCA) may be used to find the optimal basis set. The transformation performed by PCA is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA can be thought of as fitting an n-dimensional ellipsoid to the data, where each axis of the ellipsoid represents a principal component. If some axis of the ellipsoid is small, then the variance along that axis is also small, and by omitting that axis and its corresponding principal component from our representation of the dataset, we lose only a small amount of information.

To use different PSFs as the sample vectors in the PCA, each 2D PSF at a given location of the image may be vectorized and passed as a data point in an N_g dimensional space, where N_g is the number of grid points used to sample the space-varying PSF.

Fig. 3.1 shows PSF examples for a sample wide-angle lens in Zemax 2009 database. Each PSF is calculated for a given point source elevation at a given wavelength. The elevation of a source is defined as $\arctan \frac{\sqrt{x^2+y^2}}{z}$, where x, y, and z are the coordinates of the point in a coordinate system where the optical system is centered at the origin and the optical axis is aligned with the z axis. The space-varying nature of the PSF is evident by comparing the figures horizontally as the source elevation changes. Although not shown here, the PSFs simply rotate around their center if the point source is rotated around the z-axis at a fixed elevation. In other words, the PSFs are rotationally symmetric.

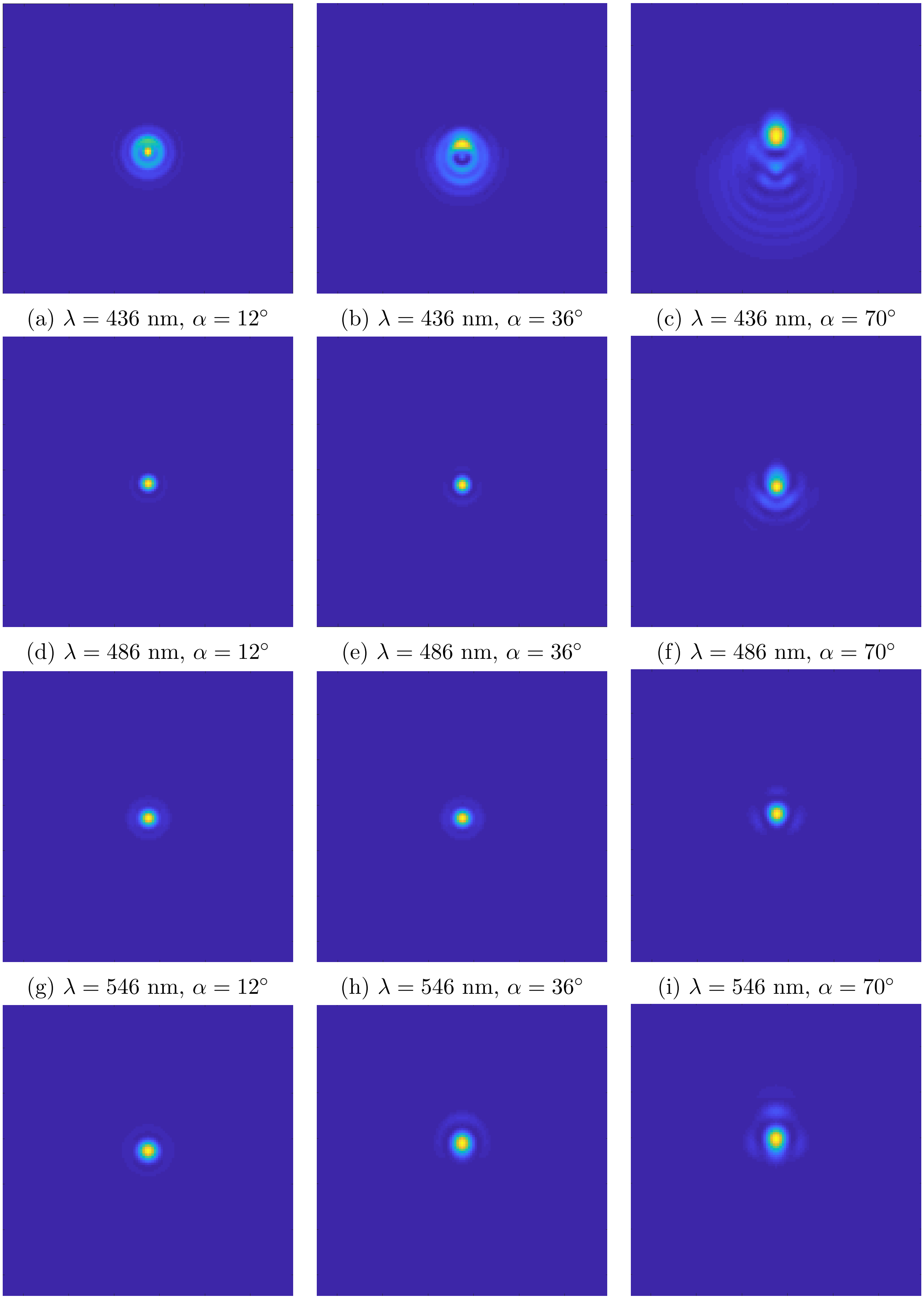
Figs. 3.2 and 3.3 illustrate the application of PCA to the PSFs of Fig. 3.1 at a wavelength of 436 nm for a fixed source azimuth. It is observed that more than 99% of the variance of PSFs can be explained by keeping only the first 6 principal components. In other words, all PSFs at all points can be expressed with a 99% accuracy as a weighted sum of the first 6 principal components shown in Fig. 3.2.

3.3 Demonstration

To demonstrate the capability of the proposed space-varying filtering method, we consider the monochrome 2228×3840 image shown in Fig. 3.4 as the scene. Each pixel of the image is interpreted as a scene patch that lies on a plane perpendicular to the optical axis. The distance of the scene to the optics is such that the scene corners have an elevation of 70° to match the PSF data in Fig. 3.1 for the 436 nm wavelength. The size of the PSFs compared to the image frame are exaggerated for illustration. Note that the same method may be applied to each wavelength of a non-monochrome scene. Here, we consider only one wavelength (436 nm) for the purpose of illustration.

The PSFs data used to generate Fig. 3.1 at 436 nm is rotated per image point to implement the rotational symmetry of the PSF. The rotated PSFs are then applied to the perfect image through a basic implementation of the convolution, resulting in the blurred image in Fig. 3.5

To calculate the PCA-based basis PSFs, we should ideally consider all $2228 \times 3840 \approx 8 \times 10^6$ PSFs as the data points to be fed into PCA, which is computationally prohibitive. Although online PCA algorithms such as the one proposed in [33] may be used, here we



(a) $\lambda = 436$ nm, $\alpha = 12°$

(b) $\lambda = 436$ nm, $\alpha = 36°$

(c) $\lambda = 436$ nm, $\alpha = 70°$

(d) $\lambda = 486$ nm, $\alpha = 12°$

(e) $\lambda = 486$ nm, $\alpha = 36°$

(f) $\lambda = 486$ nm, $\alpha = 70°$

(g) $\lambda = 546$ nm, $\alpha = 12°$

(h) $\lambda = 546$ nm, $\alpha = 36°$

(i) $\lambda = 546$ nm, $\alpha = 70°$

(j) $\lambda = 656$ nm, $\alpha = 12°$

(k) $\lambda = 656$ nm, $\alpha = 36°$

(l) $\lambda = 656$ nm, $\alpha = 70°$

Figure 3.1: The space-varying PSFs for a sammple wide-angle lens from Zemax 2009 database.

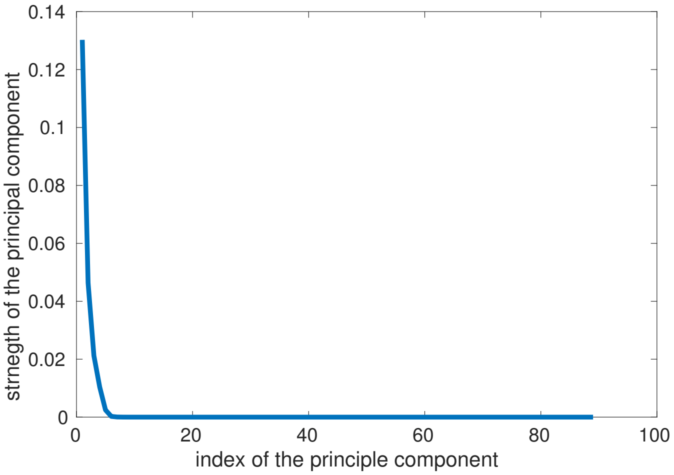


Figure 3.2: The strength of principal components of the PSFs in Fig. 3.1 at 436 nm wavelength. More than 99% of the variance can be explained by only 6 principal components.

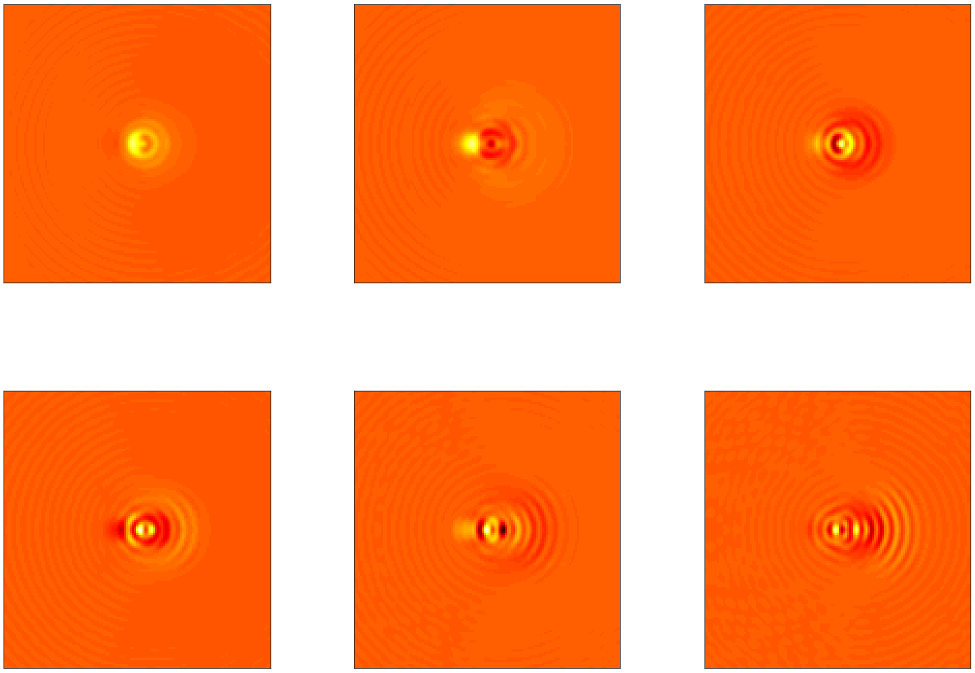


Figure 3.3: The 6 significant principle components of the PSFs of Fig. 3.1 at 436 nm wavelength. Descending order of significance: top left to top right, then bottom left to bottom right.

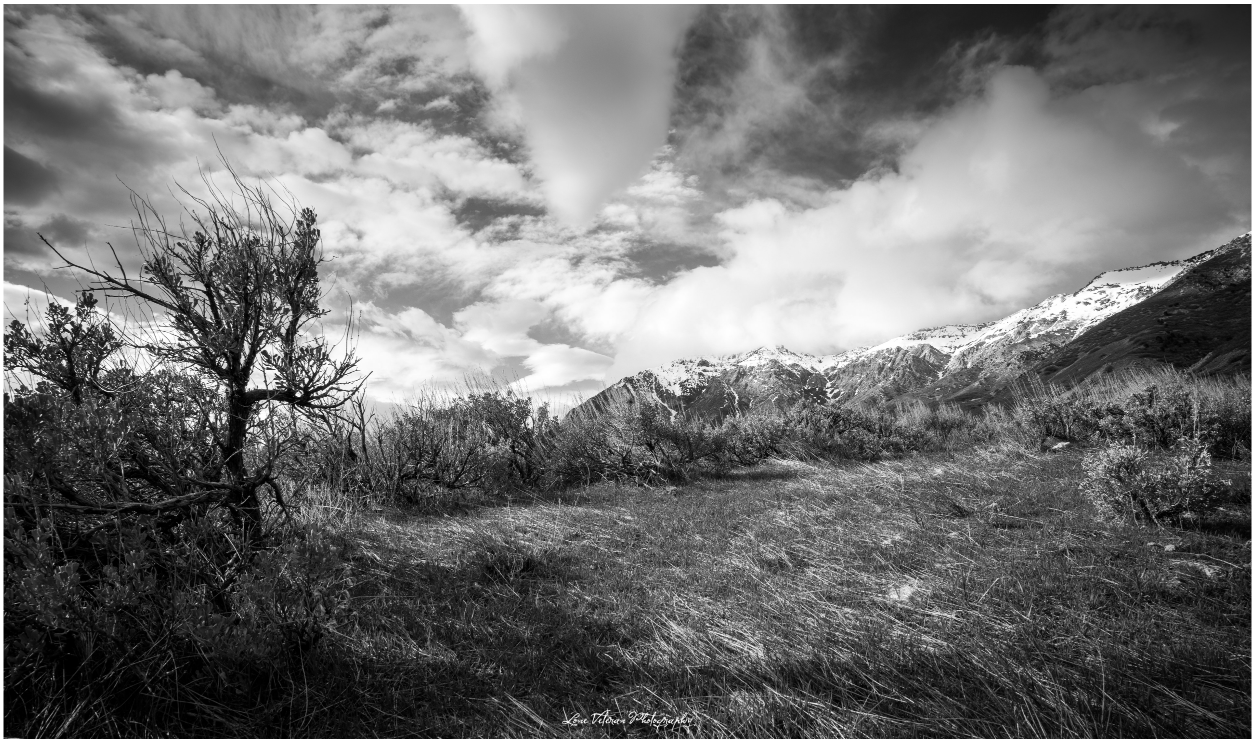


Figure 3.4: The input scene used to demonstrate the space-varying filtering.

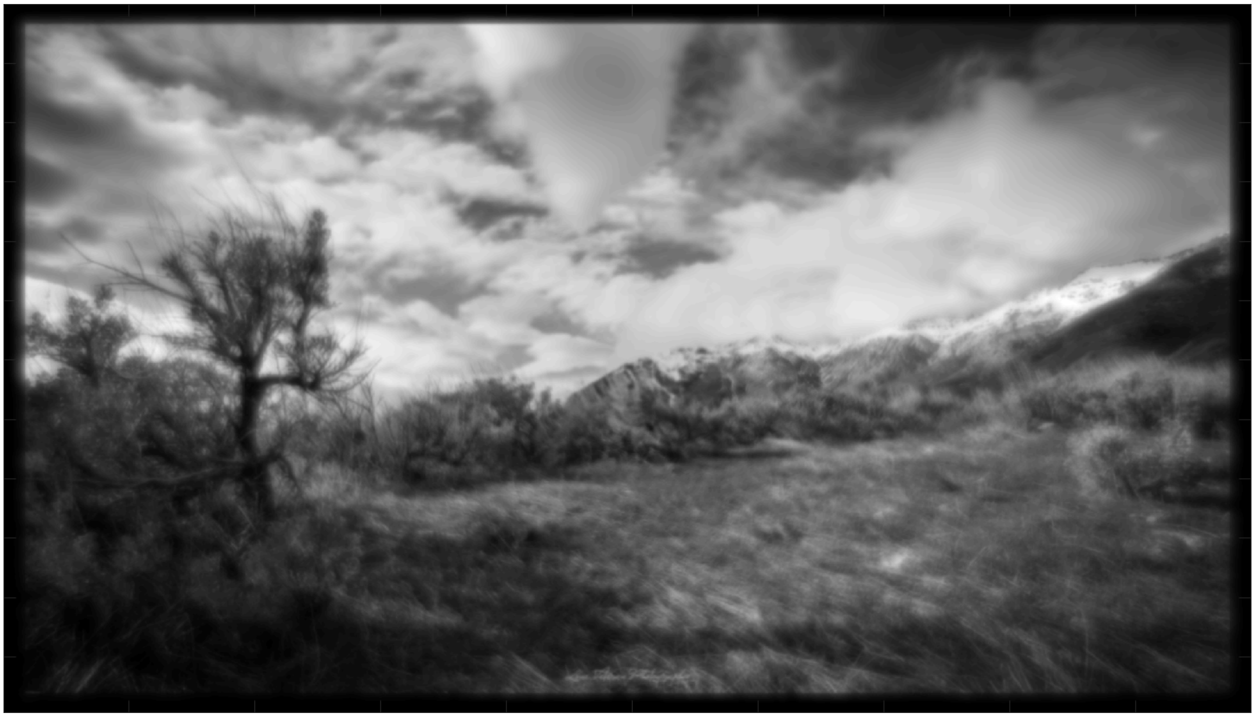


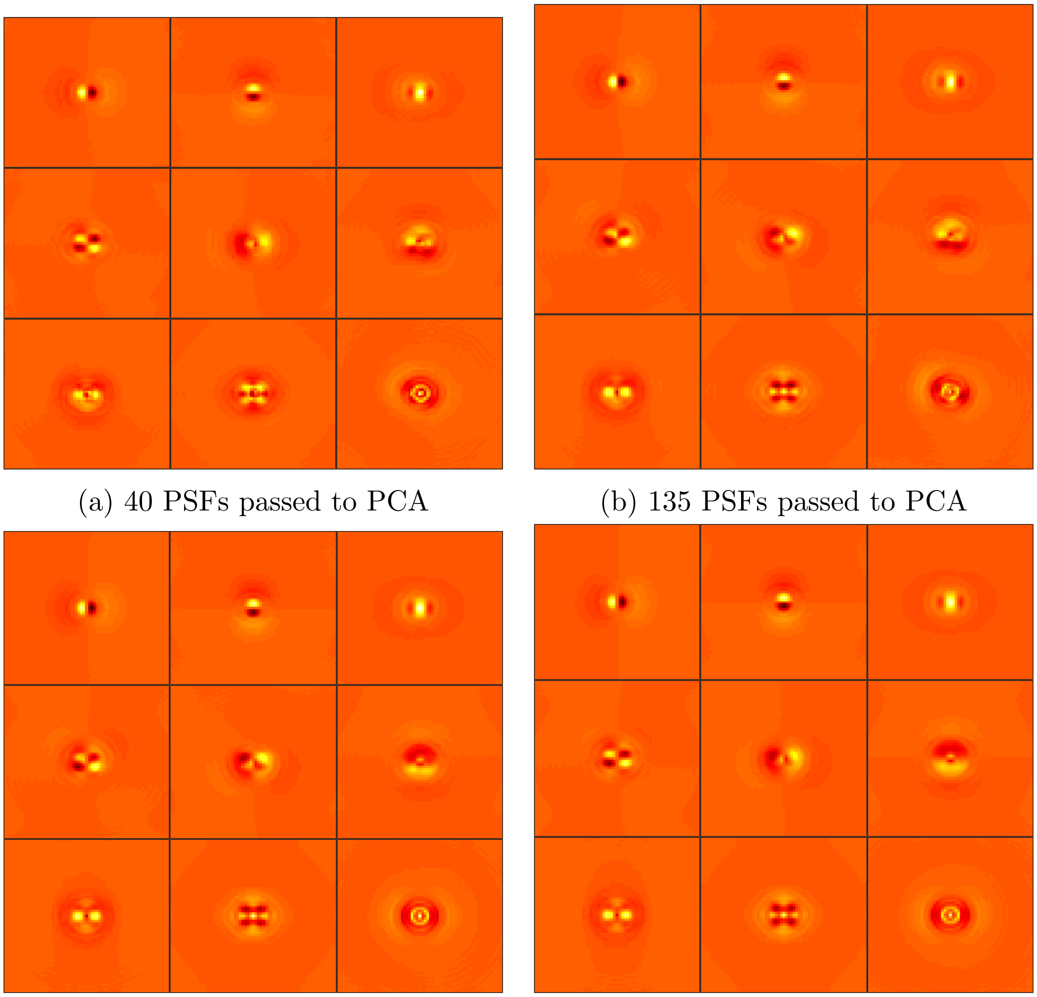
Figure 3.5: Convolution of the input scene with the space-varying kernel using a basic implementation of convolution.

consider feeding a smaller number of PSF samples to the PCA and we will show that the PCA results will not improve much after a certain number of data points have been added.

For each of the subfigures in Fig. 3.6, a different number of PSFs have been fed into the PCA algorithm. The subfigures correspond to the principal components explaining 95% of the PSF variation after downsampling by a factor of 256, 128, 64, and 32. For instance, when a downsampling of 256 is used, the PSF samples are centered at grid points that are the results of downsampling the image on each dimension by a factor of 256. The higher the downsampling factor, the fewer is the number of data points passed to the PCA. In the two extremes, the 256 and 16 downsampling factors lead to 40 and 2160 samples respectively. Despite an order of 2 magnitude difference in the amount of available data in the two extremes, it is observed that the shape and number of required principal components are relatively insensitive to the amount of data.

Once the PCA scores over the sample grids are calculated by the PCA algorithm, the scores are interpolated in the 2 dimensional space to obtain the masks to be multiplied by the image. Figs. 3.7 through 3.15 illustrate the interpolated masks and the masked images. Importantly, it is observed that PCA can neatly take care of the rotational symmetry of the PSF by providing rotational patterns in the masks.

Finally, the masked images are filtered in the frequency domain with the corresponding (zero-padded) space-invariant PSF, and the results are added together to obtain the filtered image in Fig. 3.16. On MATLAB, the brute-force space-varying convolution takes about 3 minutes on an Intel Core i-5-7200U CPU (2.5 GHz × 4). On the other hand, the PCA based method takes about 1 second on the same system and leads to a pixel-averaged error of 1.5%. Note that the results have been obtained by keeping only the first 6 principal components to account for 95% of PSF variation; in this example, keeping 12 principal components explains more than 99% of the data and leads to an average error of .3% while only doubling the computational complexity to 2 seconds.



(a) 40 PSFs passed to PCA

(b) 135 PSFs passed to PCA

(c) 540 PSFs passed to PCA

(d) 2160 PSFs passed to PCA

Figure 3.6: Principal components accounting for at least 95% of the space-varying PSF variance. The shape and number of the principal components is insensitive to the number of data points passed to the PCA algorithm as long as the samples are taken over a uniform grid.

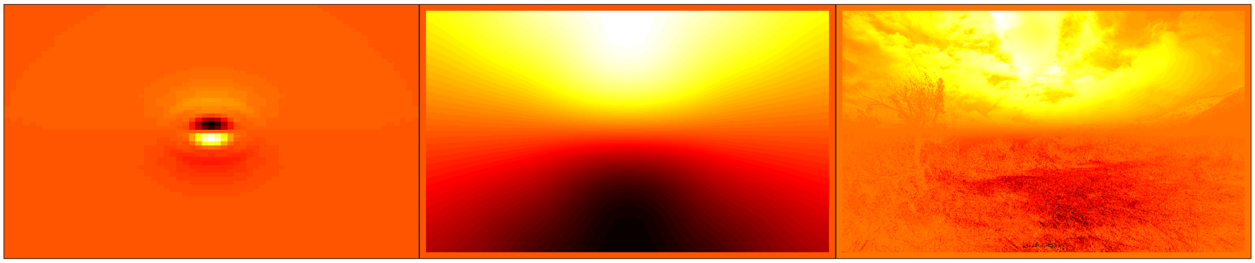


Figure 3.7: From left to right: The first principle component , the corresponding mask, and the masked image

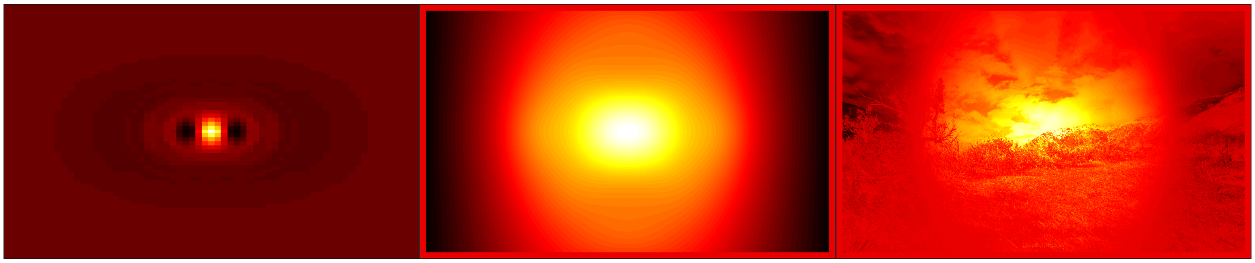


Figure 3.8: From left to right: The second principle component, the corresponding mask, and the masked image

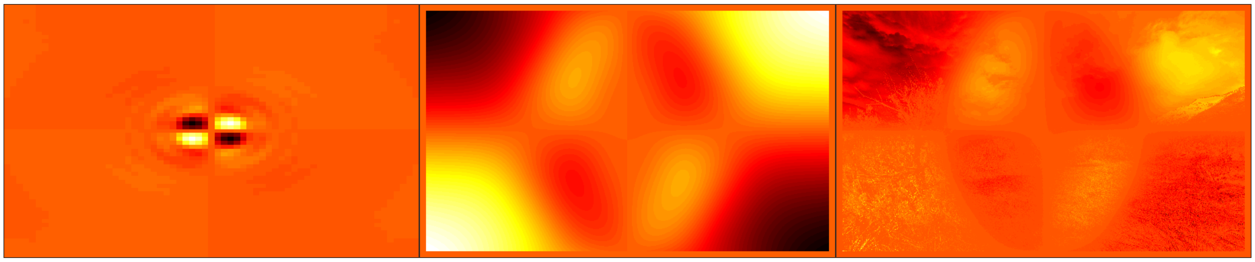


Figure 3.9: From left to right: The third principle component, the corresponding mask, and the masked image

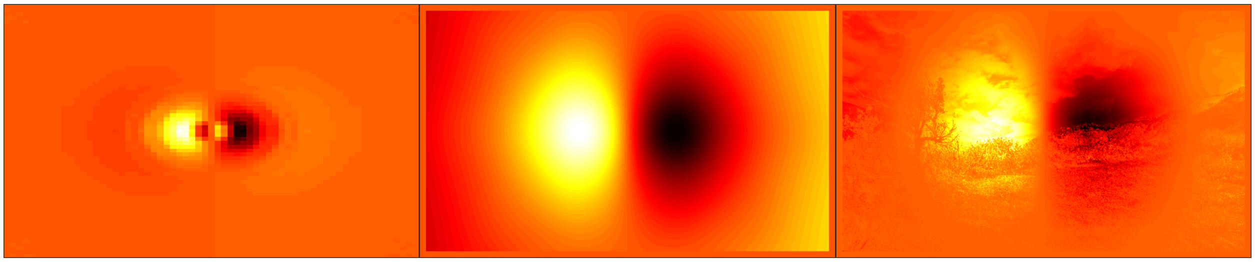


Figure 3.10: From left to right: The fourth principle component, the corresponding mask, and the masked image

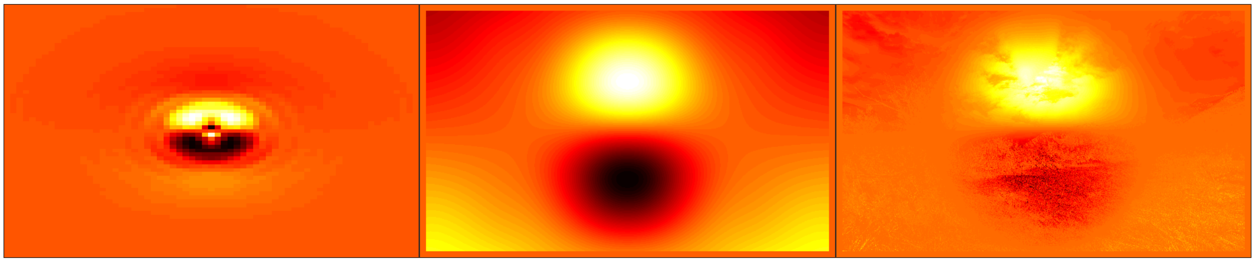


Figure 3.11: From left to right: The fifth principle component, the corresponding mask, and the masked image

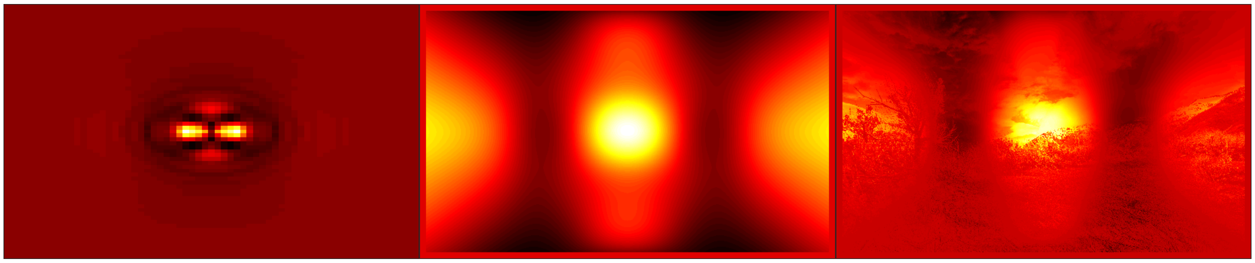


Figure 3.12: From left to right: The sixth principle component, the corresponding mask, and the masked image

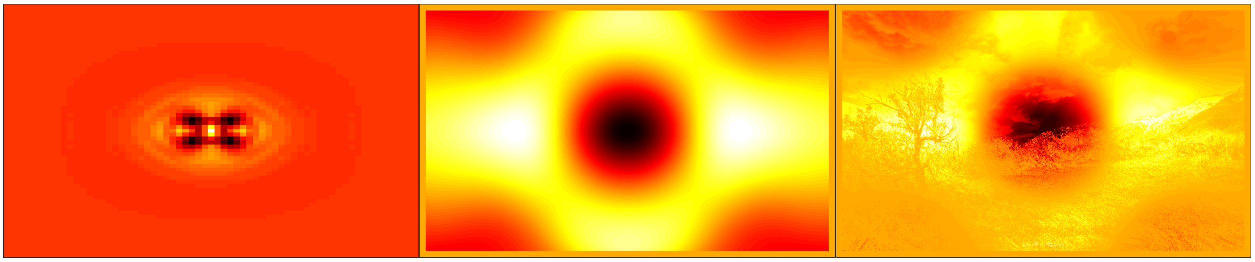


Figure 3.13: From left to right: The seventh principle component, the corresponding mask, and the masked image

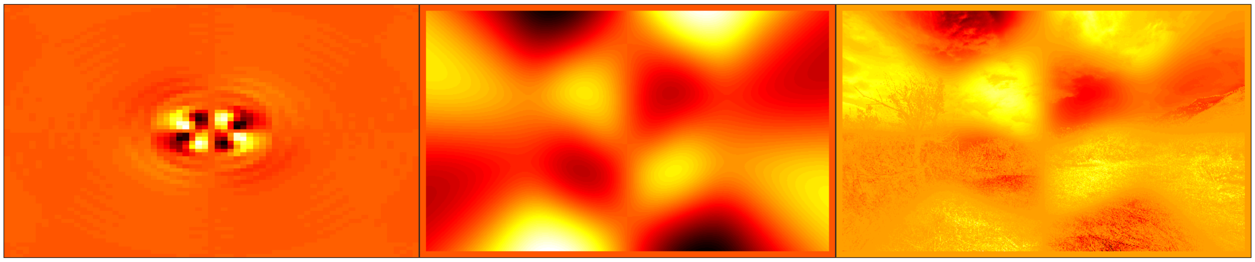


Figure 3.14: From left to right: The eighth principle component, the corresponding mask, and the masked image

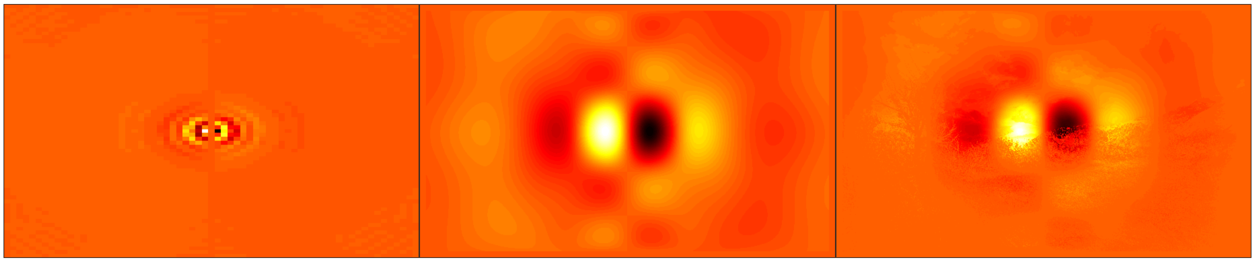
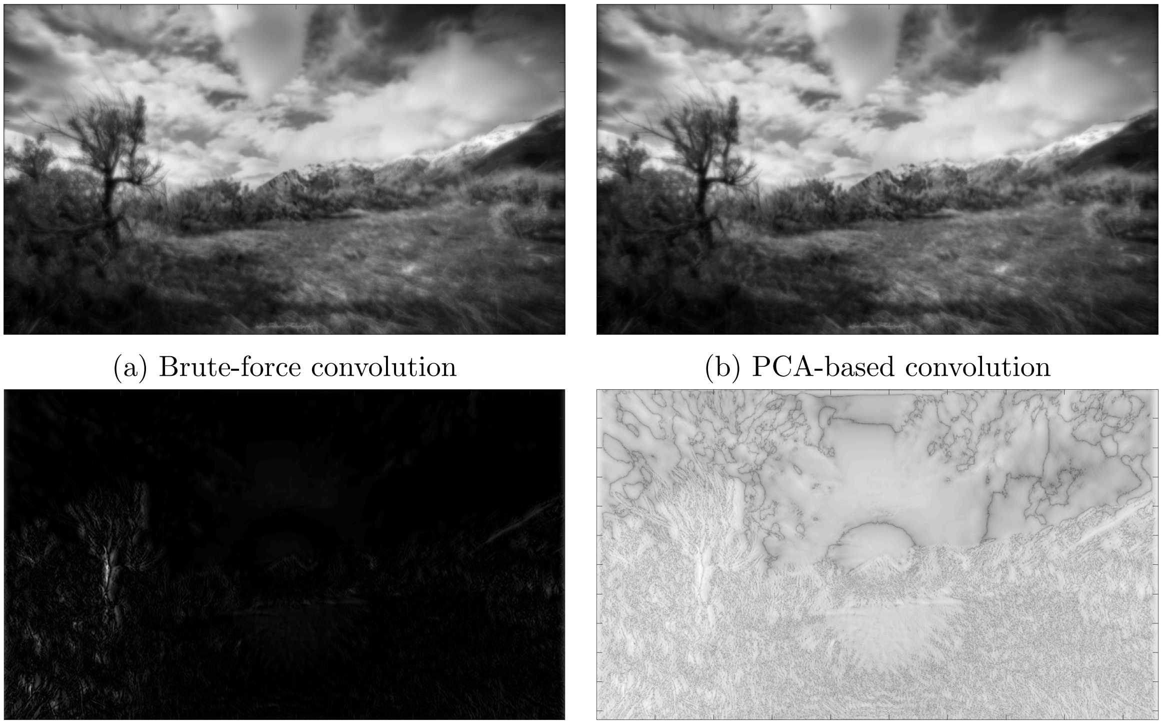


Figure 3.15: From left to right: The ninth principle component, the corresponding mask, and the masked image



(a) Brute-force convolution

(b) PCA-based convolution

(c) Graymap of relative error. The white point show a relative error of .1.

(d) Log of relative error. The darkest points correspond to 10^{-7}.

Figure 3.16: Comparison of PCA-based filtering with basic convolution.

3.4 Comparison With Previous Work

Due to different fundamental assumptions about the space-varying PSFs, only a few of the methods mentioned in Sec. 3.1 can be compared with the proposed method. In this section, we compare our results with the subspace method of [16] and the space-partition method of [12].

The boundary artifacts are a well known and visually discernible problem of the space partitioning method. However, the subspace based methods do not result in such visually perceptible issues. Consequently, rather than considering the subjective visual quality of the outputs, we focus on the objective mean pixel-error between the output of the brute-force convolution and the target method as a performance metric.

To compare the three methods, we consider the number of basis functions, N_b, required for the PCA and the subspace method to be equal to the number of partitions for the partition method. The three algorithms are evaluated using a dataset of 40 RGB images of size 2000×3000 as the scene samples. The PSFs used for the simulation are similar to those in Fig. 3.1, and a 10 nm spaced grid for wavelength between 400 nm and 700 nm is considered.

Since the RGB values do not provide any scene spectral data, we converted each RGB pixel to a synthetic spectral information by a) considering the spectrum of R, G, and B pixels made by Nanosys quantum-dot enhancement film technology as shown in Fig. 3.17, and b) assuming that each R, G, and B value in the image scales the corresponding spectrum proportionately. Adding the three spectra for each pixel then results in a full synthetic spectrum of the target pixel, and the pixel may be used as a scene patch.

Fig. 3.18 shows the results of the simulation over the dataset of the 40 images. As is clear from the error curves, the PCA-method offers a consistently lower error compared to the other two methods while keeping the computational complexity linear in the number of pixels. Importantly, it is observed that the proposed method can yield relative mean errors as low as 10^{-3} even when a small N_b of 8 is used. For all three methods, the error curves start

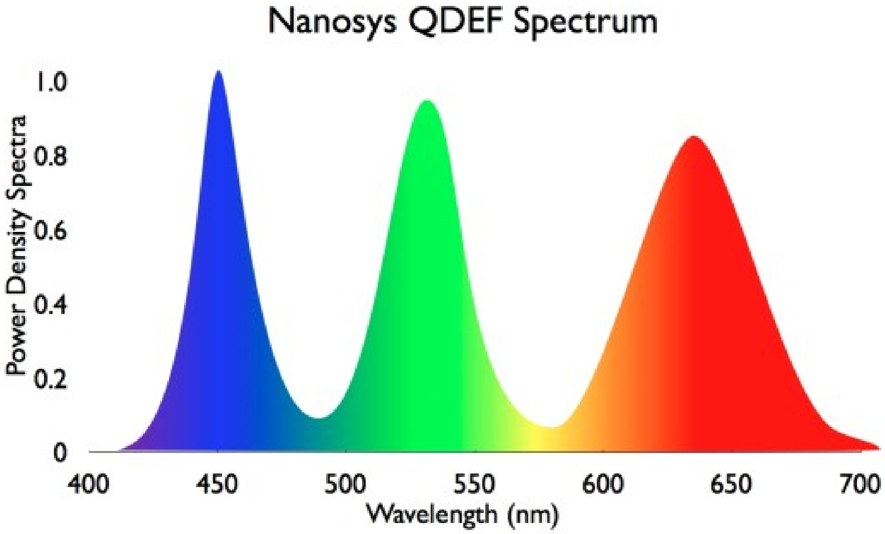


Figure 3.17: Spectrum from a Nanosys quantum-dot enhancement film (QDEF). The film contains green- and red- emitting phosphors and is stimulated by a blue GaN LED.

to plateau as we increase N_b. Compared to the PCA and the subspace method, the partition methods does not improve significantly with an increasing N_b; this may be attributed to the border artifacts that still persist even at a higher number of partitions. Finally, at large values of N_b, the performance gap between PCA and subspace method gets smaller, which is a natural consequence of using a larger number of basis functions in the subspace method. However, the high-N_b regime should be avoided because the time complexity of both PCA and the subspace methods scale linearly with N_b. The PCA method is superior to the subspace method mainly because it offers an improvement on the order of 2 in the desirable low-N_b regime with the same time-complexity.

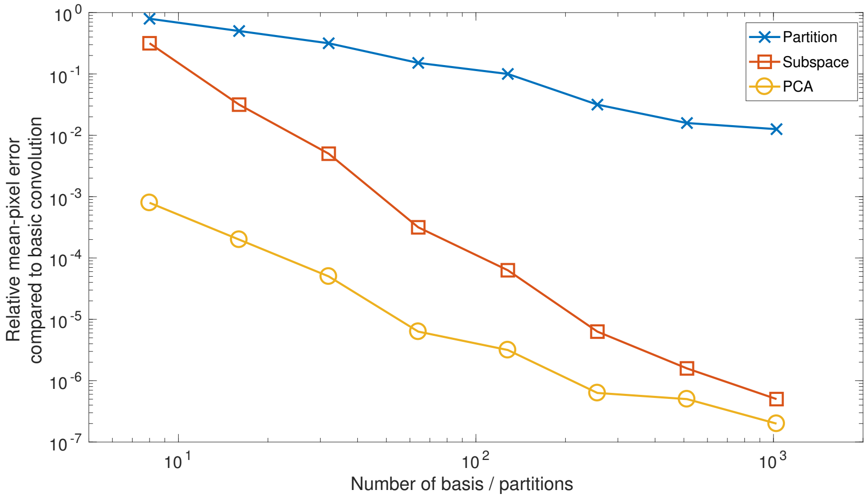


Figure 3.18: Performance of the proposed PCA-based filtering vs. the partition method [12] and subspace method [16]

Chapter 4

Conclusion

4.1 Summary

The theoretical background required to model the the elements of an imaging system was reviewed. The practical aspects of implementing each component, including time and space complexity, were investigated and several methods were suggested for a more efficient implementation.

To mitigate the computational cost of image formation in a shift-variant optical system, a new method based on Principal Component Analysis was suggested. The computational complexity and quality of the proposed method was compared to the previous work. It was empirically shown that the proposed method offers higher quality results compared to the previous work while requiring a time complexity that is only linear in the scene size.

4.2 Future Work

Although the proposed methods have been implemented in MATLAB using object oriented programming, the current framework is far from being extensible as discussed in the introduction. Attempts were made to make the framework extensible by developing a general architecture that is compatible with all components with all levels of complexity. For instance, the all scene models were equipped with a method to calculate the radiance received at patches of a given surface. Then, the optical systems were equipped with a method to convert the entrance pupil radiance to the irradiance received at an arbitrary measurement plane. Although this architecture is general and compatible with all model complexities, it does add extra overhead to the simple combinations such as a planar scene in front of a diffraction limited optical system with a measurement plane at the focal plane of the optics. Further work should be done to avoid the extra overhead on the simple models while keeping the architecture independent of the individual module implementations.

The scenes are currently modelled as a collection of objects, where each object is specified by a number of patches and each patch is specified by a set of three vertices. The radiation model of the scene is limited to incoherent light sources and opaque objects without considering the possibility of multiple scattering. Advanced 3D graphics rendering capabilities may be added to the toolbox in the future to overcome these limitations. As an alternative, the simulator may be extended to work with the standard output formats of 3D graphics rendering software such as Autodesk Maya or Autodesk 3D Max.

With the exception of diffraction limited optical systems, the mapping from optics entrance irradiance to PSF has been done using Zemax 2009 with the results stored in the proprietary binary format provided by Zemax. Although the current architecture can read the output of Zemax directly, it is more desirable to define standard storage formats and provide a mechanism for converting the proprietary outputs of different optical software into the standard formats. Alternatively, the optical modules may be written in MATLAB so that the optical calculations may be done natively without resorting to other commercial software.

Finally, the center of masses of the scene patches on the sensor plane may not finely resolve the features of the PSFs. This is a result of discretizing the scene without considering its potential effect on the accuracy of the image formed by the optics. Ideally, the scene sampling should take this effect into account. This raises the question of how the scene should be represented on a digital computer before the sampling is done. Alternatively, one might start with a PSF-agnostic sampling of the scene and then equip the scene with a resampling method to resample itself according to the PSF requirements. Adding this feature to the simulator can decouple scene representation at the time of scene creation from its appropriate representation at the time of image formation.

Bibliography

[1] Constantine A. Balanis. *Antenna Theory: Analysis and Design.* Wiley-Interscience, 2005.

[2] KURT Murat. A survey of bsdf measurements and representations. *Journal of Science and Engineering*, 20(58), 2018.

[3] Sherif Ghali. A survey of practical object space visibility algorithms. *SIGGRAPH tutorial notes*, 1(2), 2001.

[4] John F Hughes and James D Foley. *Computer graphics: Principles and Practice.* Pearson Education, 2014.

[5] JY Wang and D Es Silva. Wave-front interpretation with Zernike polynomials. *Applied optics*, 19(9):1510–1518, 1980.

[6] Jason Porter, Hope Queener, Julianna Lin, Karen Thorn, and Abdul AS Awwal. *Adaptive optics for vision science: Principles, practices, design and applications*, volume 171. John Wiley & Sons, 2006.

[7] Jason Daniel Schmidt. Numerical simulation of optical wave propagation with examples in MATLAB. SPIE Bellingham, Washington, USA, 2010.

[8] David Mas, Javier Garcia, Carlos Ferreira, Luis M Bernardo, and Francisco Marinho. Fast algorithms for free-space diffraction patterns calculation. *Optics Communications*, 164(4-6):233–245, 1999.

[9] Thomas G Stockham Jr. High-speed convolution and correlation. In *Proceedings of the April 26-28, 1966, Spring joint computer conference*, pages 229–233. ACM, 1966.

[10] James G Nagy and Dianne P O'leary. Fast iterative image restoration with a spatially varying PSF. In *Advanced Signal Processing: Algorithms, Architectures, and Implementations VII*, volume 3162, pages 388–400. International Society for Optics and Photonics, 1997.

[11] Johnathan Bardsley, Stuart Jefferies, James Nagy, and Robert Plemmons. A computational method for the restoration of images with an unknown, spatially-varying blur. *Optics express*, 14(5):1767–1782, 2006.

[12] Michael Hirsch, Suvrit Sra, Bernhard Schölkopf, and Stefan Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 607–614. IEEE, 2010.

[13] Stefan Harmeling, Hirsch Michael, and Bernhard Schölkopf. Space-variant single-image blind deconvolution for removing camera shake. In *Advances in Neural Information Processing Systems*, pages 829–837, 2010.

[14] Guotong Feng, Mohammed Shoaib, Edward L Schwartz, and M Dirk Robinson. Low-cost space-varying FIR filter architecture for computational imaging systems. In *Digital Photography VI*, volume 7537, page 75370I. International Society for Optics and Photonics, 2010.

[15] Erez Gilad and Jost Von Hardenberg. A fast algorithm for convolution integrals with space and time variant kernels. *Journal of Computational Physics*, 216(1):326–336, 2006.

[16] Loïc Denis, E Thiébaut, and Ferréol Soulez. Fast model of space-variant blurring and its application to deconvolution in astronomy. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2817–2820. IEEE, 2011.

[17] James G Nagy and Dianne P O'Leary. Restoring images degraded by spatially variant blur. *SIAM Journal on Scientific Computing*, 19(4):1063–1082, 1998.

[18] Muthuvel Arigovindan, Joshua Shaevitz, John McGowan, John W Sedat, and David A Agard. A parallel product-convolution approach for representing depth varying point spread functions in 3d widefield microscopy based on principal component analysis. *Optics express*, 18(7):6461–6476, 2010.

[19] Ralf C Flicker and François J Rigaut. Anisoplanatic deconvolution of adaptive optics images. *JOSA A*, 22(3):504–513, 2005.

[20] Jianing Wei, Charles A Bouman, and Jan P Allebach. Fast space-varying convolution using matrix source coding with applications to camera stray light reduction. *IEEE Transactions on Image Processing*, 23(5):1965–1979, 2014.

[21] Paul Escande, Pierre Weiss, and François Malgouyres. Image restoration using sparse approximations of spatially varying blur operators in the wavelet domain. In *Journal of Physics: Conference Series*, volume 464, page 012004. IOP Publishing, 2013.

[22] Samuel Farrens, FM Ngolè Mboula, and J-L Starck. Space variant deconvolution of galaxy survey images. *Astronomy & Astrophysics*, 601:A66, 2017.

[23] Adrienne Leonard, François Lanusse, and Jean-Luc Starck. Glimpse: Accurate 3d weak lensing reconstructions using sparsity. *Monthly Notices of the Royal Astronomical Society*, 440(2):1281–1294, 2014.

[24] Francois Lanusse, J-L Starck, Adrienne Leonard, and Sandrine Pires. High resolution weak lensing mass mapping combining shear and flexion. *Astronomy & Astrophysics*, 591:A2, 2016.

[25] Jerome Bobin, Florent Sureau, and J-L Starck. Polarized cosmic microwave background map recovery with sparse component separation. *Astronomy & Astrophysics*, 583:A92, 2015.

[26] Filip Sroubek, Jan Kamenicky, and Yue M Lu. Decomposition of space-variant blur in image deconvolution. *IEEE Signal Processing Letters*, 23(3):346–350, 2016.

[27] Timothy Popkin, Andrea Cavallaro, and David Hands. Accurate and efficient method for smoothly space-variant Gaussian blurring. *IEEE Transactions on image processing*, 19(5):1362–1370, 2010.

[28] Sovira Tan, Jason L Dale, and Alan Johnston. Performance of three recursive algorithms for fast space-variant Gaussian filtering. *Real-Time Imaging*, 9(3):215–228, 2003.

[29] Peter J Burt and Edward H Adelson. The Laplacian pyramid as a compact image code. In *Readings in Computer Vision*, pages 671–679. Elsevier, 1987.

[30] Jeffrey S Perry and Wilson S Geisler. Gaze-contingent real-time simulation of arbitrary visual fields. In *Human vision and electronic imaging VII*, volume 4662, pages 57–70. International Society for Optics and Photonics, 2002.

[31] Richard S Wallace, Ping-Wen Ong, Benjamin B Bederson, and Eric L Schwartz. Space variant image processing. *International Journal of Computer Vision*, 13(1):71–90, 1994.

[32] Sanghoon Lee and Alan C Bovik. Fast algorithms for foveated video processing. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(2):149–162, 2003.

[33] Manfred K Warmuth and Dima Kuzmin. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9(Oct):2287–2320, 2008.