Title
Lensless Computational Imaging using Random Optics

Author
Antipa, Nicholas Alexander

Publication Date
2020

Peer reviewed|Thesis/dissertation

Lensless Computational Imaging using Random Optics

by

Nicholas A Antipa

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Associate Professor Laura Waller, Chair
Assistant Professor Ren Ng
Associate Professor Hillel Adesnik

Summer 2020

Lensless Computational Imaging using Random Optics

Abstract

Lensless Computational Imaging using Random Optics

by

Nicholas A Antipa

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Associate Professor Laura Waller, Chair

Efficiently capturing high-dimensional optical signals, such as temporal dynamics, depth, perspective, or spectral content is a difficult imaging challenge. Because image sensors are inherently two-dimensional, direct sampling of the many dimensions that completely describe a scene presents a significant engineering challenge. Computational imaging is a design approach in which imaging hardware and digital signal processing algorithms are designed jointly to achieve performance not possible with partitioned design schemes. Within this paradigm, the sensing hardware is viewed as an encoder, coding the information of interest into measurements that can be captured with conventional sensors. Algorithms are then used to decode the information. In this dissertation, I explore the connection between optical imaging system design and compressed sensing, demonstrating that extra dimensions of optical signals (time, depth, and perspective) can be encoded into a single 2D measurement, then extracted using sparse recovery methods. The key to these capabilities is exploiting the inherent multiplexing properties of diffusers, pseudorandom free-form phase optics that scramble incident light. Contrary to their intended use, I show that certain classes of diffuser encode high-dimensional information about the incident light field into high-contrast, pseudorandom intensity patterns (caustics). Sparse recovery methods can then decode these patterns, recovering 3D images from snapshot 2D measurements. This transforms a diffuser into a computational imaging element for high-dimensional capture at video rates. Efficient physical models are introduced that reduce the computational burden for image recovery as compared to explicit matrix approaches (the computational cost remains high, however). Lastly, analysis and theory is developed that enables optimization of customized diffusers for miniaturized 3D fluorescence microscopy.

Dedicated to my family for not only making me, but making sure that I turn out ok.

Contents

List of Figures

List of Tables

Acknowledgments

I need to start by acknowledging all the people in my life who supported me through grad school. It was a strange time in my life: I had just moved back to California after an ill-fated attempt to remote work in Boston. The plan was to keep working at LLNL, and possibly go to Cal should I get in. Literally my first day back at LLNL, I received my acceptance and told my supervisors I'd be leaving again. To their credit, they were extremely understanding and even happy for me. Laura Kegelmeyer was one of my main mentors at LLNL, taking time to teach me the basics of image processing, and to promote my work within the organization. If it weren't for her help with getting my career started and cultivating my interest in image processing, I likely would never have identified computational imaging as the right field for me. Throughout my time at Berkeley, I've been in contact with Laura and Philip Kegelmeyer frequently, including their famous board game parties. Their ongoing mentorship has been crucial in my success at LLNL and beyond.

I also want to thank Sylvia for her patience, advice, and general maturity. She has patiently listened to my crackpot research ramblings on countless hikes, car rides, and breakfasts. These conversations have had significant impact on the course of my work. Her guidance in navigating the faculty application process was also crucial–how a person can have as much talent, humor, personality, and patience in one package is beyond me.

My Mom, Linda, and my Dad, Alex, have always supported my career decisions. When I said I wanted to drop engineering for music, they supported me (clearly, I never followed through on that). When deciding if grad school was right for me, they had zillions of patient talks with me, helping me think through the pros and cons. And when my appendix exploded in semester 1 of my PhD, they were there (along with Sylvia, whom I'd been dating for all of 2 months!) to help. As grad school progressed, they continued to be available to help me think through the many weird ways in which I had to grow to make this PhD thing work. Without them, that growth might not have happened, and I might never have written this dissertations.

I will avoid naming friends individually, as the number of amazing people who are voluntarily in my life has grown over the years. All of my friends played instrumental roles in my decision to come to grad school, and to finish it.

My advisors, Laura and Ren, have always pushed me to be better. I still remember the look of shock and horror on their faces when I showed them my first draft of my first conference talk. I really had no idea what level of communication was expected at a place like Berkeley, but they helped me learn, for which I'm grateful. I've been in an environment that has fostered my creativity in research, as both advisors have encouraged me to explore, try weird things, but to always do so critically and with an understanding of *why* a concept is worth exploring.

Of course, help from unofficial advisors has been crucial as well. Eric Jonas, Reinhard Heckel, and Emrah Bostan are all, as a group, responsible for my competence in the math that underlies computational imaging. Thank you for teaching me what an inverse problem is and how to solve one! A special thanks is owed to Eric for his countless zoom calls helping me through the faculty process. Without Eric, I couldn't have done it!

I want to also thank the administrative and support staff that make EECS run. Logan Baldini for keeping Cory Hall standing (what a feat!!!). Similarly, the rest of the building staff have been extremely responsive on facility issues–you rock! The office staff have always helped me with last-minute printing and scanning! David Au, for his undying support of the EE123 lab work–this adds an incredible value to the class, and without you and the rest of EECS ESG, we absolutely could not provide the rich lab experience that EE123 students receive. Finally, the advisory staff deserve a big thank-you. Shirley Salanio has been an outstanding advisor throughout my time here, not only patiently helping me navigate the bureaucracy, but also providing support in personal ways. When she asks how I'm doing, I know she genuinely wants to know because she cares.

I also want to acknowledge all of my amazing coauthors: Grace Kuo, with whom the original 3D DiffuserCam was jointly conceived, Emrah for helping us make an algorithm that actually worked, Reinhard for getting some more theory in our work, and Ben Mildenhall for making it all go fast (and getting us that best demo award at ICCP 2017!). The tireless work of Patrick Oare on the rolling shutter project convinced me, a constant skeptic, that it was actually worth pursuing. The miniscope project was the hardest research project I have ever done, and without Kyrollos Yanny's tireless hardware iteration, hacking, and 2 am prototyping, we'd never have finished. Will Liberti, computational imaging's biggest neuroscience fan, was incredibly supportive and motivating in pushing through the tough times with this project, providing samples, ground truth, and ultimately making us feel like our tech could actually be useful! I also want to thank Kristina Monakhova and Linda Liu for all the great discussion and whiteboard time in working out the design principles for random pupil coding. Finally, Michael Kellman, for all the fantastic discussions, helping me learn about deep learning, and for not getting run over in Japan. My undergraduate mentees: Sylvia Necula for helping get the original light field code working, Camille Biscarrat and Shreyas Parthasarathy for the amazing work in open-sourcing lensless cameras, Jon Fung for working toward getting learned algorithms working with real hardware, Essence Hansberry for laying the groundwork for aligning multi-sensor diffuserCams, Jon Silberstien for prototyping the first light field camera alongside me, and Gerardo Gutierrez for being my first remote-only intern, and for getting preview mode ready for prime-time.

This work was supported in part by the Defense Advanced Research Projects Agency (DARPA), Contract No. N66001-17-C-4015, Gordon and Betty Moore Foundation Data-Driven Discovery Initiative (Grant GBMF4562), National Institute of Health (NIH) grant 1R21EY027597-01, the National Science Foundation Grant No. 1617794, and an Alfred P. Sloan Foundation fellowship.

Chapter 1

Replacing lenses with random phase masks

Compressed sensing in optics

The vast majority of optical lens design has focused on one question: do the optics produce a sharp, bright image in the right place? This allows efficient capture of 2D images using film or 2D photodiode arrays. With the advent of digital sensors, these improvements to color processing, denoising, motion blur, and more became possible using signal processing applied to captured 2D images. While this has made a massive impact on all forms of imaging, the results are still largely limited to capturing 2D images.

However, the underlying dimensionality of optical information is far greater than 2. In general, we will consider the volumetric spectral radiance. We will denote this as $\boldsymbol{v}$, with units of $\frac{\mathrm{J}}{\mathrm{s{\cdot}sr{\cdot}m^3{\cdot}nm}}$. This can be thought of as energy as a function of time, space, direction, area, and wavelength: $\boldsymbol{v}(t, x, y, z, \theta, \phi, \lambda)$; this is also known as the *plenoptic function*[2]. Note, this neglects polarization, assuming unpolarized light, and does not account for interference effects. Hence, designing optical systems that can access these extra dimensions is an active area of development in imaging; this is sometimes called *high-content imaging*. Because sensors are typically only 1D or 2D, designing optics that approximate identity mappings from higher dimensions onto a 2D grid comes at the cost of limited sampling, requiring either high pixel-count sensors, which sacrifices resolution, or scanning optics which limits temporal sampling speeds. The overarching goal of this dissertation is to develop optical designs based on *compressed sensing*, which enables accessing high-dimensional image data from a single 2D exposure captured using a conventional digital imaging sensor.

Overview of contributions

My work has focused on designing optical imaging encoding hardware such that sparse recovery methods can faithfully recover high-dimensional optical signals. Specifically, the focus is on encoding extra dimensions of images into a single 2D exposure capture with a conventional photodiode array. I demonstrate this capability in proof of concept imaging

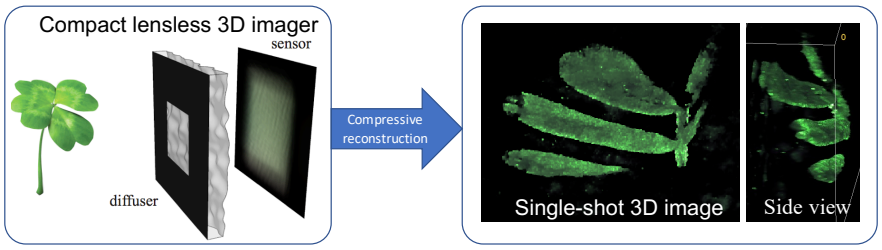


Figure 1.1: DiffuserCam: a simple lensless camera using compressed sensing to image 3D data from a single frame. Right, a sample 3D reconstruction of a small plant, computed from a single exposure.

systems that enable lensless 2D imaging using a random phase diffuser Chapter 2, snapshot volumetric imaging without lenses (Chapter 4), and encoding a high-speed video into a single rolling-shutter exposure (Chapter 3). The key insights developed in these proof-of-concept systems is synthesized into an optical design framework in Chapter 6(c) video-rate 3D imaging of fluorescence signals, such as neurons, in a device weighing under 3 grams. I utilized ideas from compressed sensing to develop the theory and practice of using pseudorandom phase masks for the capture of high dimensional optical signals. These compact hardware prototypes are possible because their design incorporates algorithms in the image formation pipeline, together with recent advances in free-form optics and rapid prototyping. Finally, Chapter 7 will cover using random phase diffusers to capture light fields.

Chapter 2 will introduce and discuss the use of random diffusers to replace lenses in conventional sensors. The result is a simple camera made up of a diffuser and a conventional CMOS sensor, which we have dubbed the *DiffuserCam*. Chapter 4 demonstrates how this camera architecture can be extended to 3D image capture. While imaging in 3D often requires multi-shot scanning, which significantly limits temporal resolution, the DiffuserCam uses compressed sensing to overcome this tradeoff by encoding volumetric images into single 2D acquisitions. This relies on two assumptions: first, the input must be sparsely representable, and second, the measurement system must map each input point to a distributed, noise-like basis function. This intuition motivates the use of diffusers, which we show can successfully encode 3D information in a single 2D measurement. Because each point within a volume maps to a unique, distributed, noise-like pattern of caustics on the sensor, sparse recovery methods reconstruct over 20 million voxels from a single 1 million-pixel 2D measurement (Fig. 1.1). Because the optics are so simple, a DIY guide for building DiffuserCam prototypes using Raspberry Pi hardware and simple optics (developed jointly with Grace Kuo, Camille Biscarat, and Shreyas Parthasarathy) is shown in Appendix 8.2.

Extending this work beyond the spatial dimensions, Chapter 3 demonstrates the innate compressive video properties of DiffuserCam for capturing temporal information in a single exposure (see Fig. 1.2). Because image sensor chips have a finite bandwidth with which to read out pixels, recording video typically requires a trade-off between frame rate and pixel count. This project demonstrates how this tradeoff can be broken using random multiplexing optics and compressed sensing. Using a random microlens diffuser, light is spread across the sensor, encoding information about the whole scene into each sensor row, which is read

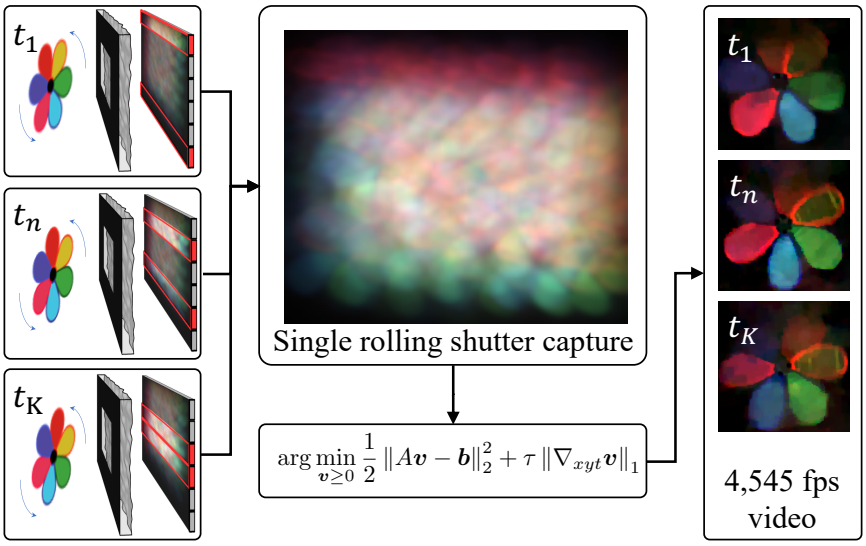


Figure 1.2: Coupling rolling shutter with DiffuserCam enables high speed video from a single exposure.

quickly using a rolling shutter CMOS sensor (note, a lens-based system cannot do this). This enables recovery of 140 video frames at over 4,500 frames per second from a single rolling shutter capture. Our lensless, proof-of-concept system uses easily-fabricated diffusers paired with an off-the-shelf sensor.

To study neural circuitry in living animals, calcium imaging has emerged as a powerful technique wherein fluorescent proteins are introduced into the neurons in living animals. These proteins change fluorescent strength as neurons fire. To record these signals over a large volume of brain tissue in a freely moving animal, compact (<3 grams) video-rate 3D fluorescent imagers with single-cell (3-10 µm) resolution are needed. This project integrates concepts from DiffuserCam into the open-source Miniscope platform by replacing the tube lens with a multifocal, nonuniform lenslet diffuser at the aperture stop. Rather than relying on randomness, I combined theory from compressed sensing and optics to optimize the lenslet design, 3D printing the design using 2-photon polymerization (Nanoscribe). By optically bonding the resulting diffuser—only tens-of-microns thick—to the back surface of an off-the-shelf lens, we transformed the miniscope into a computational camera that records 3D fluorescence at video-rates. Current prototypes are being tested on zebrafish, and I am collaborating with UC Berkeley Professor Jose Carmena's neuroscience lab to test our system in freely moving rodents. We have demonstrated time-resolved 3D neural capture of a 900×700×350 µm volume of GCaMP-tagged neurons in fixed mouse brain with single-neuron resolution (Fig. 1.3).

Background on Compressed Sensing

Compressed Sensing (CS) [26] is a modern sampling paradigm that moves beyond these limits. The key result from CS is that a sparse signal, $\mathbf{v}$, can be faithfully sampled using

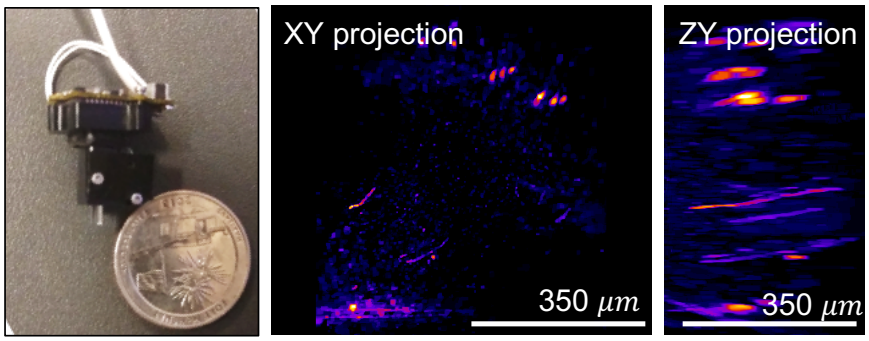


Figure 1.3: Left, the Miniscope3D prototype with 3D printed multifocal lenslet diffuser, weighing under 3 grams (US quarter for scale). Middle and right, projections of a 3D reconstruction of cleared mouse brain showing individual neuron cell bodies, as well as dendrites.

fewer samples than are required to sample the sparse signal. This requires two ingredients: a measurement system, $\mathbf{A}$, that comprises multiplexed, linear projections of the signal, and that $\mathbf{v}$ can be sparsely represented. As an optimization problem, recovering the sparsest input given linear observations $\mathbf{b} = \mathbf{A}\mathbf{v}$ can be formulated as an optimization problem:

$$\begin{aligned} \hat{\mathbf{v}} &= \underset{\mathbf{v}}{\operatorname{argmin}} \|\Psi\mathbf{v}\|_0 \\ \text{s.t. } & \mathbf{A}\mathbf{v} = \mathbf{b}, \end{aligned} \tag{1.1}$$

where $\|\mathbf{v}\|_0$ is the number of nonzeros in $\mathbf{v}$, and Ψ is a function that maps $\mathbf{v}$ to a space in which it is represented with a small number of nonzeros. This is a combinatorial problem, so a key takeaway from the CS literature is that this problem can be relaxed to a convex formulation:

$$\hat{\mathbf{v}} = \underset{\mathbf{v}\geq 0}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{A}\mathbf{v} - \mathbf{b}\|_2^2 + \tau \|\Psi\mathbf{v}\|_1 \tag{1.2}$$

where $\|\mathbf{v}\| = \sum_n |\mathbf{v}[n]|$, and $\tau > 0$ is a tuning parameter. For details, see Candes' and Wakin's tutorial on CS [26]. Note, I have included the added constraint $\mathbf{v} \geq 0$, which is not common to all CS problems, but is true for all imaging inverse problems where energy (or intensity) is the goal, as this can never be negative. Broadly, this type of inverse problem is commonly called *sparse recovery*. The term $\|\mathbf{A}\mathbf{v} - \mathbf{b}\|$, referred to as the *data fidelity*, enforces consistency between the estimated image and the measurements. Note, this is under the assumption of zero-mean additive Gaussian noise of variance σ^2 corrupting the measurements so that the measurement model is $\mathbf{A}\mathbf{v} = \mathbf{b} + \mathbf{n}$ where $\mathbf{n}[n] \sim \mathcal{N}(0, \sigma)$. The second term enforces the sparsity prior. Hence, a higher value of τ will increase the weight of the sparsity prior. A well studied choice for Ψ in imaging is $[\nabla_{d1}...\nabla_{dN}]^\intercal\mathbf{v}$, where ∇_{di} computes the finite difference of the $\mathbf{v}$ along the di axis. Penalizing the ℓ_1 norm of this enforces a sparse gradient prior, preserving edge detail in the signal. This is called *total variation* regularization [109], with the total variation semi-norm being defined as $\|\mathbf{v}\|_{TV} := \|[\nabla_{d1}...\nabla_{dN}]^\intercal\mathbf{v}\|_1$. Other common choices for Ψ include using wavelets, or $\Psi = \mathbf{I}$ for enforcing native sparsity.

The CS sampling paradigm is part of an exciting frontier in imaging system design in which the goal is to abstractly consider the role of the sensing hardware as an encoder, rather than as a direct signal approximator. This concept already had impact in fields like MRI [92] and computed tomography, accelerating scan speeds by reducing the number of samples required. As an optical design framework, the question becomes: how do we design optics that encode extra dimensions of optical images such that sparse recovery succeeds in faithfully recovering the image? Hence the goal is to study the properties of $\mathbf{A}$ that enable solving of Equation 1.2 to succeed, and translate these understandings into design requirements that drive optimization of practical optics and sensors.

A good CS sensing matrix is underdetermined, possessing more columns than rows. For sparse recovery to succeed, is should also have low *matrix coherence*, μ, defined as the maximum inner product between any two columns of the sensing matrix, $\mathbf{A}$: $\mu = \max_{\{i,j\}} \langle A_i, A_j \rangle$, where $\langle \cdot, \cdot \rangle$ denotes inner product[26]. A key consequence of this is that each column of $\mathbf{A}$ must be distributed, having many nonzero entries. This property is known as multiplexing, as it maps one input to many outputs. In the framework of optical systems design, this suggests that one-to-many imaging systems are necessary for CS in optical imaging. This is in clear opposition to the goal of ideal one-to-one mapping that is the purview of classic lens design. In the matrix-vector framework, most of lens design corresponds to engineering optics that approximate the identity system matrix, $\mathbf{A} \approx \mathbf{I}$. Another key takeaway is that the columns of $\mathbf{A}$ should be mutually orthogonal. For an underdetermined matrix, this is not possible, but can be approximately true for matrices made up of a subset of columns from $\mathbf{A}$. A common practical way of achieving this is the use of random matrices[1], in which each entry is independently chosen from a random distribution. Common choices are Gaussian and Bernoulli matrices[26]. Hence, the optical design goal is to develop multiplexing optics that produce distributed, random (or pseudorandom) measurements.

Convolutional matrices and optical systems

Efficiently solving Equation 1.2 will be discussed in Chapter 4 (Sections 4.2 and 4.6). These will all entail iterative solvers, which requires repeated application of $\mathbf{A}$ and its adjoint $\mathbf{A}^*$. For imaging problems, $\mathbf{A}$ is large, on the order of millions-by-millions when using megapixel sensors. Hence, explicit matrices are computationally expensive to work with. To alleviate this issue, good random matrices include those that not only multiplex, but also have structure facilitating fast computation. A popular choice for this is random circulant (convolutional) matrices can be efficiently diagonalized using Fast Fourier Transforms. This is a reasonable goal for optical systems which are often approximated as linear shift-invariant systems [47], characterized by a single impulse response function, termed the *point spread function* (PSF). Note that, while convolutional approaches are more efficient than explicit matrices in many cases, the CS paradigm inherently incurs a significant computational burden when compared to direct-sampling approaches. This remains a challenge for this field, limiting the application spaces where such heavy-weight processing is appropriate.

[1] such a matrix should be called *pseudorandom*, as it passes tests for randomness but is a fixed, deterministic mapping

General image formation model

The goal is efficient sensing of high-dimensional signals using practical optics and image sensors and signal processing. In essence then, the idea is to sense this high dimensional signal by measuring integrated energy on a conventional 2D photodiode array (CMOS or CCD), then using inverse problem approaches to recover the image. This method depends crucially on an accurate mathematical model of the image formation process, termed the *forward model*, which describes quantitatively the expected measurement given a known scene and fixed sensing hardware. In general, for a spectral radiance $\boldsymbol{v}'(t, x', y', \theta, \phi, \lambda)$ at the sensor, the total energy deposited at pixel $[i, j]$ (called *radiant exposure*) is

$$\mathbf{b}[i,j] = \int_0^\infty \mathrm{d}\lambda S_\lambda \iint_{-\pi}^{\pi} \mathrm{d}\theta\, \mathrm{d}\phi S_{\theta,\phi} \iint_{-\infty}^{\infty} \mathrm{d}x'\, \mathrm{d}y' S_{x',y'} \int_{-\infty}^{\infty} \mathrm{d}t S_t \boldsymbol{v}(t, x', y', \theta, \phi, \lambda), \tag{1.3}$$

where x' and y' are the lateral spatial coordinates at the sensor. Here, $S_{var}(i, j)$ denotes the response of pixel indexed by (integer) coordinates (i, j) a function of variable var.[2] For instance, S_λ is the spectral sensitivity of a pixel, and would include effects of wavelength-varying quantum efficiency in the photodiodes, as well as any color filters on the sensor. In practice, a number of assumptions will be made to simplify this. First is that we will assume slowly varying spectra and broadband sensors, treating S_λ as flat over our scenes. Additionally, pixel angular responses S_θ and S_ϕ will be assumed to be uniform, with any deviations from uniform being absorbed into the calibration or recovered images. We will assume an $M \times N$ array of pixels, indexed by row $i \in [0, M]$ and column $j \in [0, N]$. Each pixel has a square active area of size Δ, so $S_{x,y} = \mathrm{rect}\left(\frac{x'-j\Delta}{\Delta}, \frac{y'-i\Delta}{\Delta}\right)$. In other words, the pixel integrates anything that falls within its area, and ignores anything else.[3] Hence, by considering only the irradiance at the sensor, $\tilde{v}(x', y', t)$ Equation 1.3 simplifies to :

$$\mathbf{b}[i,j] = \int_{\Delta(i-1/2)}^{\Delta(i+1/2)} \mathrm{d}y' \int_{\Delta(j-1/2)}^{\Delta(j+1/2)} \mathrm{d}x' \int_{-\infty}^{\infty} \mathrm{d}t S_t \tilde{v}(t, x', y') \tag{1.4}$$

By assuming the pixel area, Δ, is small relative to the spatial frequencies in $\tilde{v}$, integration over the pixel area can be modeled as sampling, so

$$\int_{\Delta(i-1/2)}^{\Delta(i+1/2)} \mathrm{d}y' \int_{\Delta(j-1/2)}^{\Delta(j+1/2)} \mathrm{d}x' f(x', y') \approx f(i\Delta, j\Delta)$$

Hence it is convenient to consider discrete representations of all variables and operators

$$\mathbf{b}[i,j] \approx \sum_{k=0}^{T-1} S_t[k; i, j] \mathbf{v}'[i, j, k],$$

[2] because of the inherent grid-based nature of commercial sensors, the i-j dependence is implicit in these definitions. It will be introduced explicitly when needed.

[3] In practice, pixels have complex spatio-angular dependence, but this can largely be ignored for angles below 30 degrees or so, which is the case in most work presented here. This can easily become an issue at high angles, however.

where a finite exposure time of length $T\Delta_t$ s is assumed, with sampling period Δ_t. [4] Because this is a linear process, it can be abstracted as a matrix-vector multiply:

$$\mathbf{b} = \mathbf{A}_t \mathbf{v}',$$

where, conceptually, $\mathbf{v}'$ and $\mathbf{b}$ are column-vector representation of the irradiance at the sensor and the captured exposure, respectively; $\mathbf{A}_t$ is the matrix that maps from the irradiance at the sensor to the final exposure. This neglects effects of quantization and noise in the digitization of $\mathbf{b}$. Note that the temporal behavior of the sensor is a powerful tool for encoding high speed dynamics, as discussed in Chapter 3, so the structure of $\mathbf{A}_t$ is an important design variable. However, for all other chapters, S_t will be assumed to be constant over the exposure time, so the measurement process is subject to conventional Nyquist sampling assumptions: if the signal is band limited to $f_c = \frac{1}{2\Delta_t}$ Hz, it can be faithfully captured, and anything dynamics outside of f_c will alias or blur the measurement. Under the temporal band-limited assumption, the measured energy will be proportional to the mean irradiance over time $t = [0, T\Delta_t]$, denoted $\mathbf{v}'[i, j; t]$ (units W/m^2) at the sensor:

$$\mathbf{b}[i, j; t] = T\Delta_t \mathbf{v}'[i, j; t].$$

Note, when $\mathbf{v}'$ directly approximates the quantities of interest, $\mathbf{b}$ can be directly used. However, this is a very restrictive sensing scheme, limiting practical systems to capturing 2D images[5]. As outlined above, energy within an optical scene is a function of many more than 2 variable. Hence, my work that follows studies how to infer $\mathbf{v}$ as a function of 3 or more variables from a single exposure, $\mathbf{b}$, recorded with a conventional photodiode array. The goal then is to consider the optics and sensor as an *encoder*, encoding the extra degrees of freedom such that they are recoverable (via computation) from 2D discrete observations.

To this end, the process of transforming the high dimensional optical signal into the 2D time-varying intensity at the sensor is crucial. As in the exposure model outlined above, I will assume that a discrete representation of $\boldsymbol{v}(t, x, y, z, \theta, \phi, \lambda)$ suffices, with the understanding that scenes that vary faster than the grid used for any particular dimension will be incorrectly represented. For the remainder of this dissertation, it will be implicit that when the arguments are excluded following $\mathbf{v}$, that variable has minimal impact on the measurements and can safely be neglected. For example, $\mathbf{v}[x, y]$ implies a 2D spatial image, and is suitable for imaging a still, 2D scene. Additionally, when square brackets are used, it is assumed that the arguments are discrete integers in $[0, N-1]$ where N samples are assumed along that dimension of $\mathbf{v}$. In other words, the arguments are indices, not physical units. Continuous variables will be denoted with round brackets, $\boldsymbol{v}(\mathbf{r})$, where $\mathbf{r}$ is the N-dimensional continuous coordinate. When recovering discrete images on a unitless grid, they will be converted to physical units as needed after computation.

[4] The notation $f[y; x]$ denotes that function f is evaluated on variable y, but is parameterized by parameters x.

[5] Filters can be placed atop each pixel to provide independent measurement across other dimensions, but when coupled with one-to-one imaging systems, this severely limits the final image resolution, and relies on interpolation to fill in gaps. A conventional Bayer pattern is an example of this, but the same concept can be used for polarization. See [113] for more.

The final piece to the forward model puzzle is to compute how the quantities of interest, $\mathbf{v}[n]$, maps to the time-varying irradiance at the sensor, $\mathbf{v}'[n]$. Note, n may represent the multiple dimensions necessary to index $\mathbf{v}$. In general, n will contain multiple indexing variables. I will assume everything is spatially and temporally incoherent, such that energy only adds at the sensor plane[6]. This is suitable for a wide range of scenes such as natural photography and fluorescence imaging. Hence, each point in $\mathbf{v}'$ can is a linear combination of points in $\mathbf{v}$, with weights determined by the optical system. More explicitly:

$$\mathbf{b}[i,j] = \sum_n \mathbf{v}[n]h[i,j;n]$$

Here, $h[i,j;n]$ is the energy striking pixel $[i,j]$ after light from scene point indexed by n passes through the optical system and is exposed on the sensor. This linear map from discretely represented image, $\mathbf{v}$, to final measurement $\mathbf{b}$, will be written in matrix-vector form frequently:

$$\mathbf{b} = \mathbf{A}\mathbf{v}. \tag{1.5}$$

For the remainder of this work, I will assume that an image can be denoted as a vector, but its underlying shape does not need to be 1D, as is conventional for vectors. Depending on the underlying structure of the vector, it may be convenient to index with multiple indices, but this can easily be converted to a single index if needed. For example, in 2D imaging, it is natural to refer to $\mathbf{v}[x,y]$ due to the two underlying spatial dimensions inherent in the signal. This structure matters when choosing signal priors, as locality exists in the 2D signal that is not as obvious in a 1D representation. Note, however, that for any linear operator that finds local structure in the 2D representation, and equivalent operator could be computed in the 1D representation that would be exactly equivalent. See Appendix 8.1 for more details. A second point worth mentioning is the confusion over the word *dimension*. In the vector-space sense, this refers to the underlying number of basis vectors necessary to describe a vector space. However, it is common terminology in imaging to use the word *dimension* to describe the number of basis vectors needed to span the image's domain. In other words, *dimension* refers to the number of arguments needed to index an image; for example, a 2-dimensional image is a function of a 2-dimensional space, even if the vector representation of the image is of far higher dimensionality in the vector sense. Lastly, when norm notation is used on image vectors, for example the 2-norm $\|\mathbf{v}\|_2$, this implies a vector norm, not a matrix norm (even if the image is 2D).

Previous Work

Romberg [108] proposed the use of convolutional random matrices for CS in imaging, pointing out that random phase is a good choice. Coded aperture phase has also been explored by Willett and Roummel [54] in simulation. Random erasures at the sensor plane have also

[6]This means no destructive interference is possible between scene points. Interference effects can be present in the impulse response, however, so wave optics is not totally neglected in this framework

been proposed [89]. However, as Romberg points out, pointwise erasure is a poor choice when the signal is sparsely represented using spatially compact basis functions, suggesting that random phase is a better choice for imaging.

Lensless cameras, in which the camera comprises only a mask and a sensor, offer a convenient package for exploring multiplexed optical imaging, as they inherently map scene points to distributed patterns at the sensor plane. Additionally, Lensless cameras are attractive for their potentially small form factor, with broad investigation in applications of 2D photography. Unlike traditional cameras, in which a point in the scene maps to a point on the sensor, lensless cameras map a point in the scene to many points on the sensor, requiring computational reconstruction. While this can be a hindrance for 2D imaging, it offers an opportunity when viewed through the encoder-decoder framework of CS.

Related works in 2D lensless imaging

The roots of lensless imaging come from imaging in domains where lenses are impossible to build due to the lack of materials with strong refractive index, such as x-ray and 3He based neutron detection[28]. This idea was ported to optical imaging with the goal of creating thin, low cost imagers by replacing a lens with an encoding element placed directly in front of the sensor. 2D lensless cameras have demonstrated passive incoherent imaging using amplitude masks [13], diffractive masks [123, 46], random reflective surfaces [40, 124], and modified microlens arrays [125]. Due to the unique form factors, wide field-of-view (FoV) cameras have been proposed for thermal imaging[46], and the joint use of opposing image sensors acting as the other's coding element [97]. Whereas these approaches rely largely on absorbing masks, which are light inefficient, lensless 2D imaging has also been demonstrated using inline holography and scattering masks [53, 27, 119, 120]. These approaches require illuminating the object with coherent lighting, precluding their use in natural scenes or for imaging fluorescence. The mask-based lensless camera system described in Chapter 2 and 4, dubbed DiffuserCam, utilizes a random phase masks designed to create sufficient contrast with incoherent scenes to enable light efficient lensless imaging for natural scenes.

Related works in single-shot 3D imaging

Single-shot 3D imaging, in which a 3D image is encoded into a single 2D measurement, has been demonstrated in a variety of architectures. Light field cameras, also called integral imagers, passively capture 4D space-angle information in a single-shot [98], which can be used for 3D reconstructions. This concept can be built into a thin form factor with microlens arrays [58] or Fresnel zone plates [63]. Lenslet array-based 3D capture schemes have also been used in microscopy [76], where wave-optical [24, 83] or scattering [105, 83] effects can be included. All of these systems, however, must trade resolution (or field-of-view) for single-shot capture, limiting the number of useful voxels. DiffuserCam improves upon this tradeoff, capturing large 3D volumes with high voxel counts in a single exposure.

Coherent 3D lensless imaging has been demonstrated as well [23, 75, 18, 38, 117], but these methods require active (coherent) illumination, limiting applications. Further, many

coherent methods do not generate unambiguous 3D reconstructions, but rather use digital refocusing to estimate depth. DiffuserCam, on the other hand, exhibits actual depth sectioning (in the absence of occlusions) for "true 3D".

Since methods for imaging through scattering often use diffusers as a proxy for general scattering media [68, 36, 118], our mathematical models will be similar. However, instead of trying to mitigate the effects of unwanted scattering, here we use the diffuser as an optical element in our system design. We choose a thin, optically smooth diffuser that refracts pseudorandomly, producing high contrast patterns under incoherent illumination. Such diffusers have been used in light field imaging [8] and coherent holography [75, 67]. Coherent multiple scattering has been demonstrated as an encoding mechanism for 2D compressed sensing [87], but necessitates a transmission matrix approach that does not scale well past a few thousand pixels. We achieve similar benefits without needing coherent illumination, and we reconstruct 3D objects, rather than 2D. Finally, an important benefit of our system over previous work is the simple calibration and efficient computation that allow for 3D reconstruction at megavoxel scales with superior image quality.

Related works in compressive video

To capture high-speed dynamics with conventional sensors, one must overcome the bandwidth limit of digital imaging chips. Compressive video works by spatio-temporally coding the video data prior to capture. Rather than capture a video, then compress it to exploit redundancies, compressive video does the compression step in hardware and captures only relevant data. For example, Hitomi *et al.* proposed a compressive video acquisition scheme that reconstructs a high-speed video from a single image ($9-18\times$ temporal upsampling at 1000 fps) [141]. The approach relied on pixel-wise programmable exposure timing to modulate the recorded image temporally during the acquisition. Reconstruction was performed through a dictionary of space-time signal patches that is learned offline. Experimentally, the approach used a spatial light modulator (SLM) and global shutter sensor, but could theoretically be implemented on-chip in a CMOS architecture. On-chip random downsampling has been implemented in circuitry using a $\Sigma-\Delta$ approach, which reduces the burden on optical design, but requires modification of sensor fabrication process–this is far more difficult to scale than optical redesigns [102]. Using strobed exposure with unique sequences, Veeraraghavan *et al.* reconstructed a high-speed video of periodic events at 2000 fps from a video captured by a camera operating at 25 fps [132]. Another technique, proposed by Llull and Yuan *et al.*, achieved high-speed video reconstruction (22 frames at 660 fps) from a single-shot coded-aperture image that is obtained by translating binary amplitude masks within the focal plane of a global shutter sensor [89, 143]. Koller *et al.* later improved the mask design [69] and Liu *et al.* proposed a reconstruction that exploits the low-rank structure of the underlying scene [86]. The commonality between these setups is that each pixel is *temporally* modulated during the exposure, and all require bulky and expensive hardware. Our technique, in contrast, uses simple optics and *spatial* multiplexing rather than temporal.

Rolling shutter can induce undesirable artifacts when imaging dynamic scenes. Removal of such artifacts is an active field of study. Liang *et al.* characterized and corrected the

geometric distortions [77]. Saurer *et al.* considered extensions for stereo imaging and registration with rolling shutter cameras [112]. When camera motion exists, Su and Heidrich [110] proposed an approach to reconstruct a sharp image by simultaneously removing the motion blur and rolling shutter distortions.

Rather than undoing the effects of rolling shutter sensors, we seek to leverage them for performance. Gu *et al.* have proposed controlling the readout timing and exposure length for each row [50] such that the exposure time discrepancy in subsequent rows enables one to flexibly sample the 3D space-time volume of the dynamic scene. In simulations, their architecture-level proposal was beneficial for computational photography applications such as high dynamic range (HDR) imaging and auto-exposure, but did not successfully resolve video using sparse recovery methods. Oieke and Gamal proposed another architecture that used spatial multiplexing at the chip-level, which allowed them to reach 1920 fps data rate for 256×256 pixel count. Another method uses digital micro-mirror devices (DMDs) for aperture coding and streak cameras with femtosecond speeds to reconstruct ultrafast videos (10 trillion fps) from a single image [74, 60]. Liu *et al.* considered similar ideas and used a galvanometer to perform streaking (*i.e.* temporal shearing of the scene) [85]. While this concept is similar to ours in spirit, they do not consider spatial multiplexing and they rely on complex, costly hardware. Finally, Sheinin *et al.* recently used rolling shutter and spatial multiplexing to detect and de-mix the contributions from flickering light bulbs in a scene, providing useful information about the power grid. The authors observed that spatial-multiplexing via a diffuser enabled observation of spatio-temporal information, but they do not considering high-speed imaging directly [115].

Spatially-multiplexed image capture has been a key ingredient for compressive imaging [35]. Using amplitude masks, Salman *et al.* realized such ideas on a lensless and compact system [13]. Diffuser (*i.e.* phase mask)-based lensless cameras have been shown to be capable of 2D imaging [72], and single-shot 3D imaging [11]. Here, we show that diffusers are useful optical elements for compressive video systems, allowing each frame of video to be sampled from a small subset of sensor pixels. Our system can be calibrated from a single image, fabricated using simple lab equipment, and reconstructed using computationally-efficient convolution-based algorithms.

Related works in compact 3D microscopy

Because of the combination of single-shot 3D and miniature form factor, CS-based imaging systems are an attractive option for developing compact 3D microscopes. While many volumetric microscopy methods capture 3D structure using scanning (e.g. two-photon, light sheet), this which is difficult to miniaturize and trades temporal resolution against field-of-view (FoV). Two-photon microscopes have been implemented in small form factors [145, 55], giving high resolution at a cost of motion artifacts [106], limited FoV, and expensive hardware. Miniaturized light sheet microscopes achieve faster capture [37], but also depend on scanning which causes motion artifacts and increases the complexity and size of the hardware.

Unlike scanning approaches, single-shot methods [142, 82, 11, 73, 1, 14, 76, 24] offer faster capture speeds, with temporal resolution limited only by camera speed. These methods illuminate a fluorescent sample with excitation light, the optically encode the volumetric emission into a single 2D measurement, computationally reconstructing the 3D information. Single-shot 3D fluorescence capture has been demonstrated using a lensless architecture [1, 73], but lacked the integrated illumination that is required for in-vivo imaging. In addition, such mask-only systems have no magnifying optics, and so are limited to low effective numerical aperture (NA) resulting in poor lateral and axial resolutions. Other recent work combines coding elements with multi-fiber endoscopes to achieve single-shot non-fluorescence 3D, with relatively low resolution [116]. Recently, the miniature light field microscope (MiniLFM) [121] demonstrated an integrated 3D fluorescence system with computationally-efficient temporal video processing for neural activity tracking [99]. This system adds a standard microlens array (regularly-spaced, unifocal) to the image plane of a miniature compound microscope, giving it single-shot 3D capabilities at the cost of degraded lateral resolution and a larger and heavier device. As discussed in Chapter 6, replacing the tube lens with a pseudorandom phase mask more efficiently captures 3D information. The key to this is utilizing CS theory to optimize the phase mask for improved performance over what can be achieved with conventional microlenses.

Related works in single-shot light field capture

Single-shot capture of in-camera light fields using microlenses is an active area of research, with a long history dating back to the early 1900s [81, 59, 103, 3, 96, 98, 91, 79, 139]. The technique is variously referred to as integral, plenoptic and light field imaging. The main tradeoff is reduced image resolution in order to sample angular information. Microlens approaches have also been applied in microscopy [76], where a wave-optics model can be used to exploit diffraction effects for higher-resolution reconstruction at some distances [24]. This approach recognizes that microlens arrays become subject to wave-optics effects as their sizes shrink. In this paper, we extend this line of thinking to a new framework in which the phase-encoded surface need not be a pre-designed periodic array of lenslets, but can be any phase (or amplitude) mask, even with irregular surface relief and diffractive properties.

Attenuation masks are another approach for encoding light fields in a single-shot [133]. Recent work has shown that these systems can be modeled using a matrix method that is amenable to compressive sensing, possibly overcoming the resolution trade-offs inherent in 4D imaging with 2D sensors [94, 64]. These techniques require significantly more computational resources than microlens systems in order to infer the light field from the sensor measurements, and the masks attenuate a significant amount of light. In this paper, we use phase encoding rather than attenuation encoding, thereby avoiding light loss. However, our system is similar to the mask-based methods in that it implements a multiplexed-type measurement that is able to exploit a priori information (e.g. sparsity) for possible resolution benefits.

Microlens arrays and attenuation masks have also been used without a main imaging lens to create very flat cameras for 2D imaging [126, 14], since the lens function can be achieved

in computation. Very small 2D cameras have also been created with diffractive optics and computation [122] using a purely wave-optics model. Although it is not shown in this paper, our approach of using diffusers could also be extended to flat imaging device designs, but with volumetric reconstructions resulting from the light field data.

Other approaches to light field capture include angle sensitive pixels [136, 56], aperture scanning with time multiplexing [78], macroscopic lens arrays [44], and camera arrays [140, 134]. Various attempts at obtaining higher image resolution have been proposed, for example depth-aware splatting [41] and hybrid imaging by also using a high-resolution 2D camera [19].

Designed phase plates have been used in 2D imaging to extend depth of field [34, 29] and improve signal for phase retrieval [6, 93]. Random planar refractive masks have been used to reconstruct 2D images and estimate depth [40]. Recent efforts have attempted to image objects through unknown random diffusers [68, 53, 7], which suggests possible future applications for our work.

The work demonstrated in Chapter 7 considers the use of a diffuser to encode light fields in a single snapshot. The goal of this work was to realize benefits of compressed sensing in light field capture, alleviating the sampling limit common to state-of-the-art plenoptic cameras. However, this was not successfully demonstrated for reasons discussed at the end of the chapter.

Chapter 2

Capturing 2D images with a diffuser

This is work done jointly with Grace Kuo, Camille Biscarrat, Ren Ng, and Laura Waller and is based on [72].

2.1 Introduction

This chapter introduces the lensless imaging architecture and demonstrates the ability to capture 2D photographs using a camera comprised only of a diffuser and a sensor. Mask-based lensless imagers [14, 125, 122] are a new class of computational cameras that use an optical mask rather than a traditional lens. Unlike a lens, a mask does not directly produce an image on the sensor. Rather, the mask indirectly encodes object irradiance, which must then be algorithmically recovered from the sensor data. Mask-based cameras have several advantages, including thin form factor, low weight, potential for scaling to larger sensor formats, and ability to capture depth information.

We propose a lensless camera comprised only of a diffuser (a pseudo-random phase mask) placed a small distance away from an image sensor. The diffuser is a piece of clear polymer with a smooth, slowly-varying surface, and it is the only optical element in our system (Figure 2.1a). When illuminated by a point source, the convex bumps on the diffuser concentrate rays, creating a high contrast caustic pattern at the sensor (Figure 2.1b) [8]. The diffuser shape need not be known *a priori*, as the system will be calibrated from a single image of a point source.

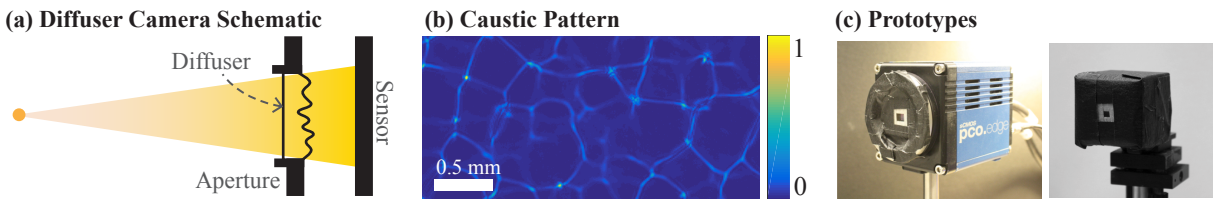


Figure 2.1: (a) Our lensless camera is simply a diffuser placed a fixed distance from a sensor. (b) Experimentally measured caustic pattern. (c) Prototype systems (PCO left, Point Grey right).

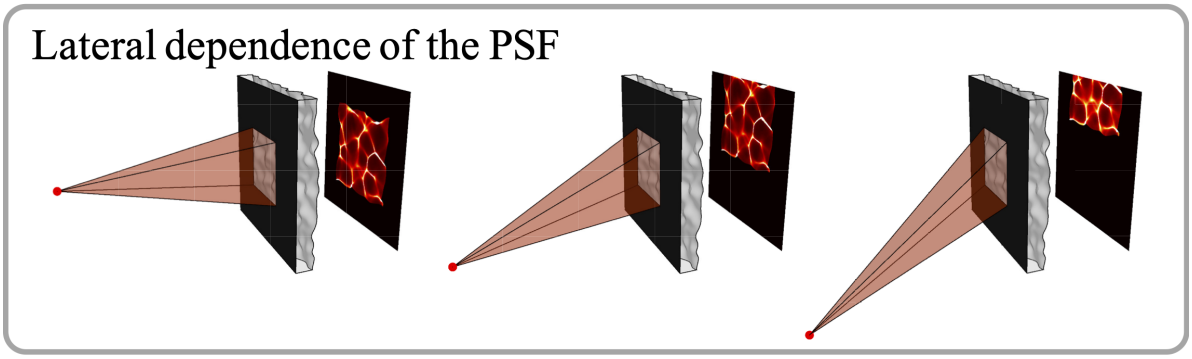


Figure 2.2: Shift invariance of the Diffuser PSF. Translating a point at a fixed depth leads to a translation (in the opposite direction) of the PSF on the sensor. This behavior is validated

The advantages of our system compared to other lensless cameras are the use of off-the-shelf parts, high light throughput compared to amplitude masks, simple calibration, digital auto-focus, and an efficient computational model. As a step toward providing design tools for general mask-based cameras, we present a framework for analyzing the resolution, field-of-view, and depth-of-field of such systems. We built two experimental prototypes (Fig. 2.1c) and show high-quality image reconstructions by using an optimization-based deconvolution algorithm.

Convolutional Forward Model

Because the diffuser surface is slowly varying, a point source located in the far-field can be treated paraxially. Recovering an image requires knowing the system matrix, $\mathbf{A}$, which is extremely large (on the order of millions-by-millions). Measuring or storing the full $\mathbf{A}$ would be impractical, requiring millions of calibration images and operating on multi-Terabyte matrices. Instead, we use the convolution model outlined below to drastically reduce complexity of both calibration and computation.

We describe the object, $\mathbf{v}$, as a set of point sources located at $[x, y, z]$ on a non-Cartesian 2D grid at distance z from the camera. The relative radiant power collected by the aperture from each source is $\mathbf{v}[x, y, z]$. The caustic pattern at pixel $[x', y']$ on the sensor due to a unit-energy point source at $[x, y, z]$ is the PSF, $h[x', y'; x, y, z]$. Thus, $\mathbf{b}[x', y']$ is the sum of all 2D sensor measurements for each non-zero point in $\mathbf{v}$ after propagating through the diffuser and onto the sensor. This lets us explicitly write the matrix-vector multiplication $\mathbf{Av}$ by summing over all scene points in the FoV:

$$\mathbf{b}[x', y'] = \sum_{[x,y]} \mathbf{v}[x, y, z] h[x', y'; x, y, z]. \tag{2.1}$$

This can be simplified by treating the diffuser as paraxial, which leads to a shift invariant PSF. This is similar to the *infinite memory effect* [68, 36], and is a valid model because the diffuser has a slowly varying, smooth, and relatively shallow surface. Consider the caustics created by point sources at a fixed distance, z, from the diffuser. A lateral translation of

the source by $(\Delta x, \Delta y)$ leads to a lateral shift of the caustics on the sensor by $(\Delta x', \Delta y') = (m\Delta x, m\Delta y)$, where m is the paraxial magnification. We validate this behavior in both simulations (see Fig. 4.2) and experiments (see Sec. 4.34.3). For notational convenience, we define the on-axis caustic pattern at depth z as $h[x', y'] := h[x', y'; 0, 0]$, and assume $m = -1$, applying correct scaling after image recovery. Thus, the off-axis caustic pattern is given by $h[x', y'; x, y] = h[x' - x, y' - y]$. Plugging into (2.1), the sensor measurement due to a 2D scene at depth z is:

$$\begin{aligned} \mathbf{b}[x', y'] &= \sum_{[x,y]} \mathbf{v}[x, y, z] h[x' - x, y' - y] \\ &= \mathbf{C}\left[\left(\mathbf{v}\left[x, y\right] \overset{(x,y)}{*} h\left[x, y\right]\right)\left[x', y'\right]\right]. \end{aligned} \tag{2.2}$$

Here, $\overset{(x,y)}{*}$ represents 2D linear discrete convolution over (x, y), which returns arrays that are larger than the originals. Hence, we crop to the original sensor size, denoted by the linear operator $\mathbf{C}$ (see Supplementary Fig. S5 for more details). For an object discretized into N_z depth slices, the number of columns of $\mathbf{A}$ is N_z times larger than the number of elements in $\mathbf{b}$ (*i.e.* the number of sensor pixels), so our system is underdetermined.

This model as a number of benefits. First, it allows us to compute $\mathbf{Av}$ as a linear operator in terms of only 1 image, rather than instantiating $\mathbf{A}$ explicitly. We evaluate the cropped convolutions are evaluated using circular 3D convolution, implemented with 2D FFTs, which scales well. Second, this model coupled with the distributed, pseudorandom PSF connects well to thery of random matrices in compressed sensing [70]. The third benefit is ease of calibration, requiring only one calibration image at the desired imaging depth. Note that depth-dependence of this problem is explored in Chapter 4. This also motivates the addition of the aperture on the diffuser, which ensures that a single image captures the entire PSF.

2.2 Methods

Calibration amounts to collecting a single image of the PSF, $\mathbf{h}[x', y']$, by illuminating with an LED located 2 m from the sensor. The PSF exhibits slight variations with depth, but this can be modeled under the paraxial approximation, making digital autofocusing after image acquisition straightforward (see Section 2.2). In contrast, [14] is calibrated by projecting a series of Hadamard patterns on the camera, and calibration must be repeated for objects outside the depth-of-field.

Recovering the image amounts to solving Equation 1.2. Methods such as Fast Iterative Shrinkage Thresholding Algorithm (FISTA)[15, 101], or Alternating Direction Method of Multipliers (ADMM) [22, 21, 135] suffice, using Equation 2.2 as the forward model. While this form of inverse problem is necessary for CS recovery problems, in this case the sparsity prior serves only to regularize the problem, reducing the impact of noise on the reconstructions. Methods for solving this efficiently with total variation are discussed in Chapter 4.

Resolution

If the PSFs of neighboring point sources are very similar, it is challenging to distinguish between the sources. This causes poor reconstruction quality, lowering the SNR and resolution. To quantify this, consider two point sources located (Δ_x, Δ_y) apart with PSFs $\mathbf{h}_1[x', y']$ and $\mathbf{h}_2[x', y']$. Leveraging the shift invariance of the problem, we can define $h_1[x', y'] := \mathbf{h}[x', y'; 0, 0]$, the on-axis PSF, and $\mathbf{h}_2[x', y'] := \mathbf{h}[x', y'; \Delta_x, \Delta_y]$. We define *similarity* between these two PSFs, denoted $\mu(\Delta\boldsymbol{\xi})$, as the inner product of the PSFs:

$$\begin{aligned}\mu(\Delta_x, \Delta_y) &= \langle \mathbf{h}[x', y'; 0, 0], \mathbf{h}[x', y'; \Delta_x, \Delta_y] \rangle \\ &= \langle \mathbf{h}[x', y'], \mathbf{h}[x' - \Delta_x, y' - \Delta_y] \rangle\end{aligned} \tag{2.3}$$

The second line comes from the shift-invariance assumption, and is the 2D autocorrelation of $\mathbf{h}$. For normalized PSFs, $\mu(0, 0) = 1$ by definition, and ideally, μ should decrease quickly as $\left\|[\Delta_x, \Delta_y]\right\|$ increases. We quantify the autocorrelation sharpness by looking at its half-width at 70% of maximum, which empirically matches our data (see Sec. 2.3 and Fig. 2.3b). Because $\mathbf{h}$ is nonnegative, μ can only be zero when two PSFs occupy a completely disjoint set of pixels.

Field-of-view

Theoretically, the angular field of view (FOV) of our camera is determined by the maximum illumination angle that contributes to the sensor measurement. Since the diffuser bends light, we take into account the diffuser's maximum deflection angle, denoted β. Based on the geometry shown in Fig. 2.3a, we calculate that the angular FOV α satisfies $l + w = d \tan(\alpha - \beta)$ where $2l$ is the sensor width, $2w$ is the width of the PSF support, and d is the distance from the diffuser to sensor. Finally, real-world sensor pixels cannot detect light from arbitrarily high angles, so we include their maximum angle of acceptance, α_c, in our final FOV equation:

$$\alpha = \beta + \min\left[\alpha_c, \tan^{-1}\left(\tfrac{l+w}{d}\right)\right] \tag{2.4}$$

Depth-of-field

Consider two on-axis point sources at different depths, z_1 and z_2. We define the depth-of-field (DOF) to be the minimum detectable separation, $\Delta z = z_1 - z_2$. Treating the diffuser paraxially, the corresponding PSFs, $\mathbf{h}[\mathbf{x}; z_1]$ and $\mathbf{h}[\mathbf{x}; z_2]$, are related by a coordinate scaling with parameter s: $\mathbf{h}[s\mathbf{x}; z_1] = \mathbf{h}[\mathbf{x}; z_2]$. Plugging this into the similarity definition in Eq. 2.3, we can determine the depth sensitivity of the camera in terms of a single PSF measurement and s using $\mu(s) = \langle \mathbf{h}[s\mathbf{x}], \mathbf{h}[\mathbf{x}] \rangle$. Similar to Section 2.2, we determine the values of s for which μ is sufficiently low. Then, we relate s geometrically to the corresponding DOF.

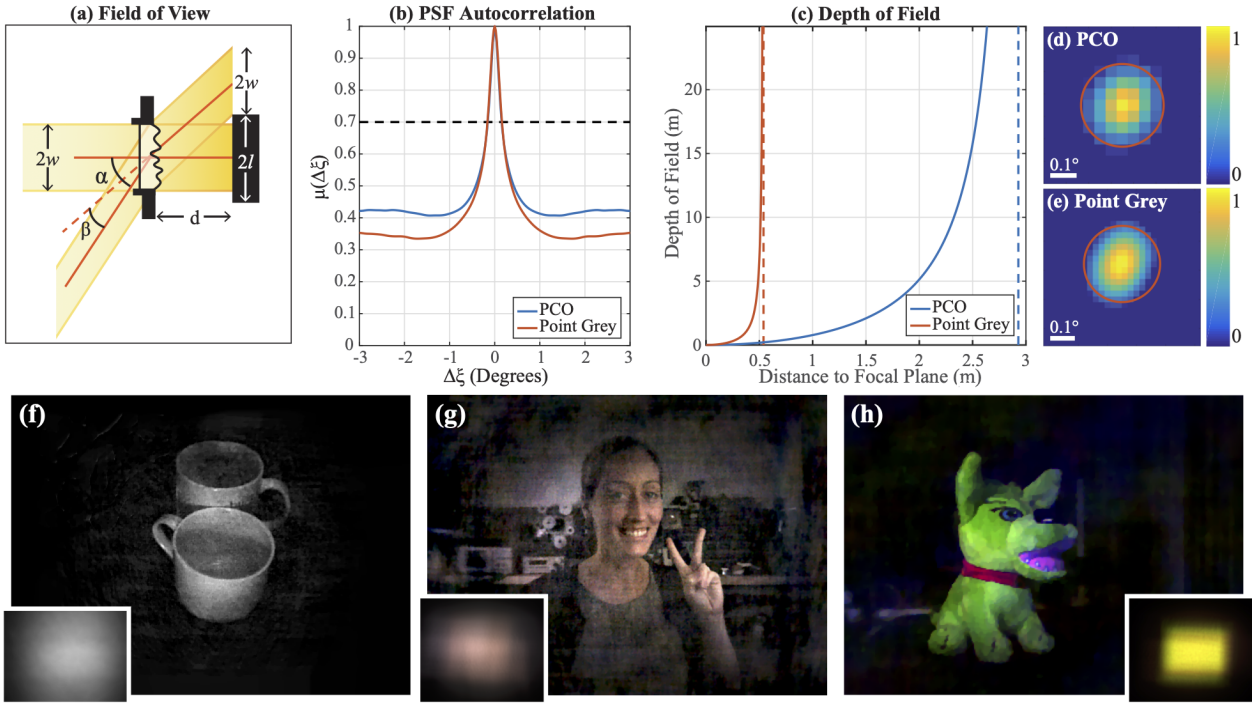


Figure 2.3: Analysis and results of DiffuserCam. (a) Schematic showing geometric effects that contribute to field-of-view. (b) Autocorrelation of diffuser PSF for the two prototypes, which sets optical resolution limits.(c) Depth of field (solid) and hyperfocal distance (dotted) for the two prototypes. (d)-(e) Zoom-in on the reconstruction of a single point source captured with each camera to illustrate resolution. The red circles represent estimated spot size based on autocorrelation width at 70% of maximum. (f) Reconstructed image from the Point Grey prototype (300 × 400 pixels). Raw data shown in inset. (g)-(h) Reconstructed image from the PCO prototype (640 × 540 pixels). Raw data shown in inset.

2.3 Experimental Results

To demonstrate the ease with which our method can be adapted to any existing sensor and how sensor parameters affect imaging characteristics, we built two prototype cameras. One uses a PCO Edge 5.5 Color camera, and the other a Point Grey Flea3 with Sony IMX036 monochrome CMOS chip. We placed a 0.5° Luminit Light Shaping Diffuser at $d = 8.8$ mm $d = 6.4$ mm, respectively (Figure 2.1c). The surface of the 0.5° diffuser has a maximum angle of about 1.5°, making the paraxial approximation valid. We measured the experimental PSF of each prototype using an LED, and additionally measured the angular acceptance of each sensor, α_c, by translating the calibration LED. We found that using a cutoff of 20% of the brightness at normal incidence predicts the FOV we observe experimentally.

Using the measured PSFs in conjunction with the analysis presented in Sections 2.2-2.2, we compute the theoretical system parameters for each prototype. For the PCO camera, the resolution, defined by the autocorrelation peak half-width, is 0.16°, and the half FOV is 37°. A line from each autocorrelation is shown in Fig. 2.3b. The DOF of the PCO camera is shown in Fig. 2.3c (blue), with a hyperfocal distance of 2929 mm. For the Point Grey prototype, the resolution is also 0.16°, and the half FOV is 27.5°. The DOF is plotted in

Fig. 2.3e (red), with a hyperfocal distance of 571 mm. Note that the smaller sensor size of the Point Grey camera results in significantly larger depth of field. However, since both prototypes use the same diffuser, the angular resolution is the same, despite differences in pixel size.

To validate resolution, we took images of a single point source with each prototype, and reconstructions of each are shown in Figure 2.3d-e. The spot-size radius matches our theoretical resolution for each camera. We also took images of several objects (Figure 2.3f-h). All objects were kept within one DOF, and the observed FOV matches our theory. Instructions for building a camera are in the Appendix, 8.2.

2.4 Compressed sensing

The end goal of using a multiplexing random phase mask is to realize benefits from compressed sensing. To test this, we attempt to recover an image after discarding a fraction of sensor pixels. This is equivalent to erasure random rows from $\mathbf{A}$, as is common in MRI[92]. The mathematical model for this is $\mathbf{b}[i,j] = \mathbf{\Theta C}\left(\mathbf{v} * \mathbf{h}\right)$, where $\mathbf{\Theta}$ is a pointwise erasure operator that zeros pixels we wish do discard. We find that an image can be recovered succesfully from a small number of pixels, as shown in Figure 2.4(a), which shows reconstructions with 100%, 10%, and 1.5%. Figure 2.4(b) compares the reconstruction using all pixels to two methods of recovering from erausre. The middle erases 98.5% of the pixels from the reconstruction on the left then uses bicubic interpolation to get back to the high resolution grid. The image on the right show erasure of 98.5% of pixels from the raw data prior to reconstruction. Arguably, the righthand image using CS is perhaps slightly more detailed, but it is also noisier than the bicubic method. Hence, for static 2D imaging from subsampled sensor measurements, we observe that the CS theory works: we can recover an image from a subset of sensor pixels. However, for static scenes, the degradation in quality is similar to what would be lost by simply using a lower resolution sensor and upsampling with bicubic interpolation. However, in the next chapter, we will explore utilizing this concept to enable high speed imaging by coupling a diffuser with a rolling shutter sensor, which can be viewed as a pixel erasure problem. They key to overcoming the lost quality shown here will be to consider jointly multiple time points, relying on temporal regularization to regain detail in the recovered images.

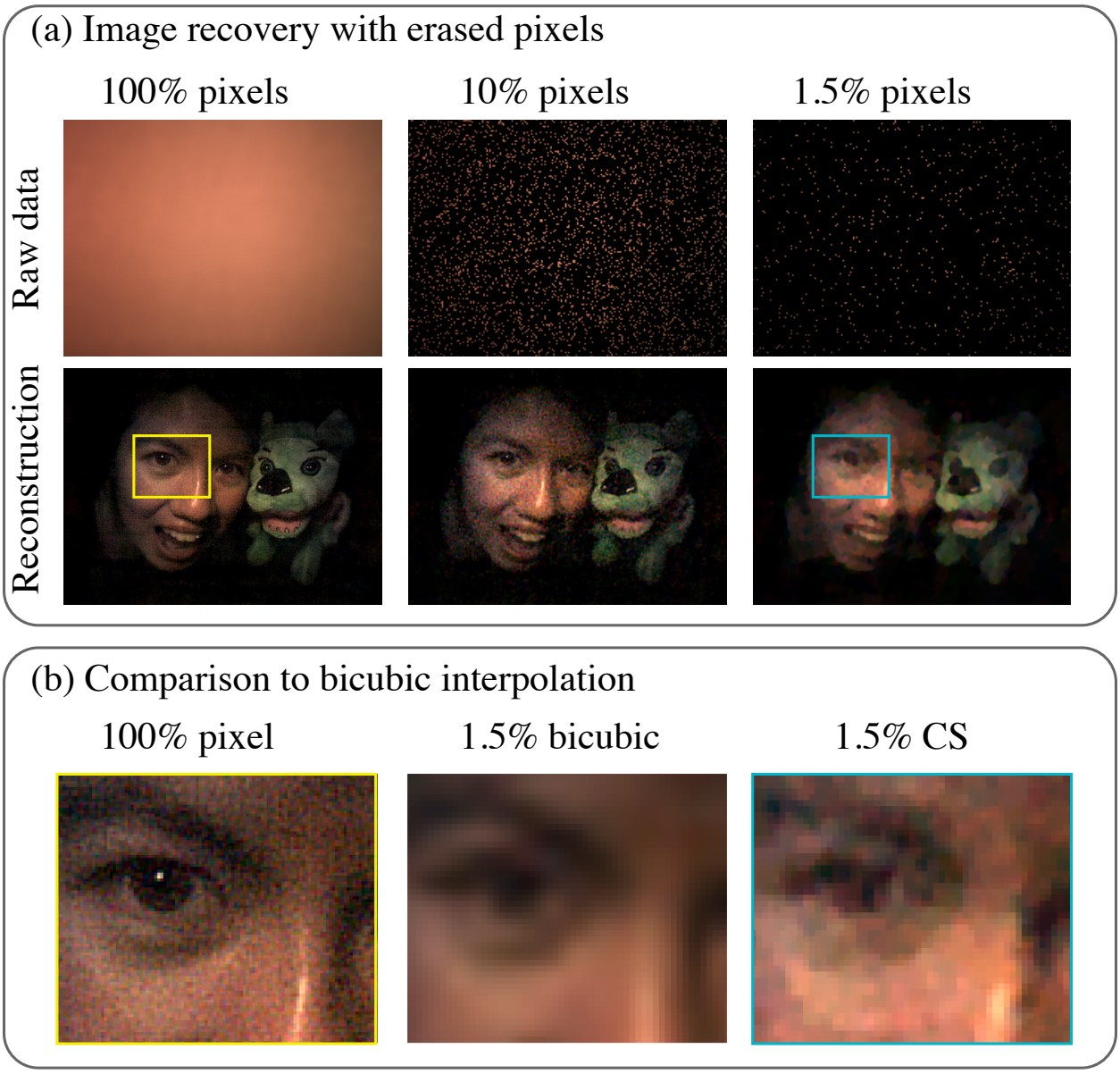

(a) Image recovery with erased pixels
100% pixels
10% pixels
1.5% pixels
Raw data
Reconstruction
(b) Comparison to bicubic interpolation
100% pixel
1.5% bicubic
1.5% CS

Figure 2.4

Chapter 3

Compressive High Speed Video in Lensless Cameras

This is work done jointly with Patrick Oare, Emrah Bostan, Ren Ng, and Laura Waller and is based on [12].

3.1 Introduction

All digital imaging sensors have a finite bit rate for exporting the digital measurement. This limited bit rate restricts the space-time bandwidth of the system, forcing a trade-off between temporal and spatial resolution. Traditionally, increasing the frame rate while maintaining pixel count requires increasing the chip bandwidth, which is expensive. Compressive video approaches seek to break this trade-off by spatio-temporally compressing the video data prior to exporting the bits, effectively encoding more information into the limited bandwidth. While most work in compressive video has focused on redesigning the readout architecture of CMOS chips, we instead propose a compressive video scheme based on optical multiplexing using a diffuser. We demonstrate the concept using a simple lensless camera with an off-the-shelf rolling shutter sensor. Our system effectively encodes 140 frames into a single still image.

Increasing the frame rate of a sensor with fixed bandwidth can be achieved by reading a subset of pixels at each frame. However, when using one-to-one imaging optics (*i.e.* lenses) that map each scene point to a point on the sensor, information is lost from parts of the sensor that are not sampled. Figure 3.1(a) illustrates a sensor with a narrow band of pixels actively recording, placed at the image plane of a lens, with a simple scene consisting of two point sources. The cyan source falls outside of the active exposure band and is therefore not measured. To solve this problem, we propose using spatial-multiplexing optics such that even a small subset of sensor pixels (e.g. one row of a 2D array) contain information from most scene points. Our approach consists of replacing the lens with a pseudorandom

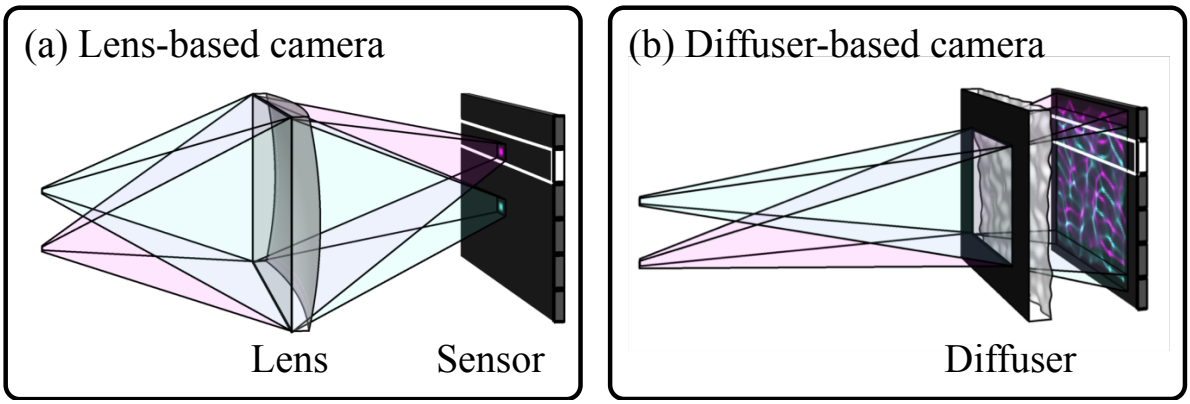


Figure 3.1: Diffuser-encoded pseudorandom multiplexing ensures that every row in the sensor measurement contains information from nearly every scene point. (a) A lens-based camera maps each scene point to a point on the sensor. If the sensor samples a subset of rows at a time (outlined in white), as with rolling shutter, only one row of the scene is visible. For example, the cyan point is completely missed in this case. (b) Multiplexing optics, such as a diffuser, spread information across the sensor, allowing the entire scene to be sampled by the subset of rows illustrated here. This effect enables our lensless system to recover a video at a frame rate set by the sensor line scan rate.

phase diffuser placed near the sensor, which maps each point to a distributed, high-contrast pattern of caustics on the sensor. As shown in Fig. 3.1(b), the information from every scene point falls on *nearly all* sensor pixels, and is therefore present in the band of rows being read. Recovering a video from a sequence of row measurements then requires solving an underdetermined inverse problem. Because the diffuser produces pseudorandom noise-like measurements, we interpret this as a compressive sensing system, reconstructing the video using sparsity-constrained nonlinear optimization.

To implement this idea, we leverage the ubiquity of *rolling shutter* CMOS sensors. During capture of a single image, rolling shutter sensors expose each row of pixels over a unique time window. This encodes temporal information into the 2D measurement. By randomly multiplexing the scene onto such a sensor, we can recover a video of a dynamic scene wherein each frame corresponds to a row of the rolling shutter capture.

Our experimental prototype recovers 140 frames of video at $4,545$ frames-per-second (fps) from a single 2D rolling shutter capture. The system is built using a dual-shutter sCMOS sensor (Fig. 3.2). We analyze the spatial and temporal resolution of the system and show that, for sparse scenes, the spatial resolution significantly surpasses that of much more expensive global shutter approaches at comparable frame rates.

3.2 Forward model and inverse problem

In this section, we outline a forward model for the optics and the rolling shutter exposure, as well as the inverse problem approach. We will use this model to analyze the temporal resolution of the system in Section 3.4.

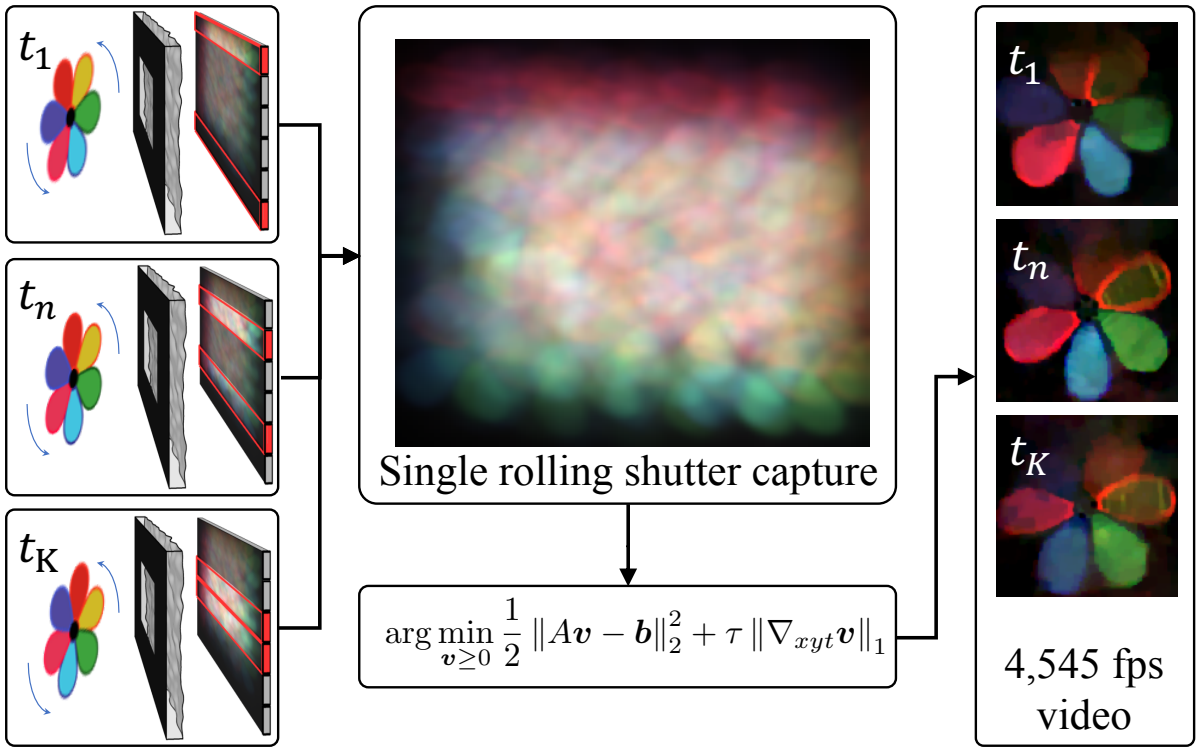


Figure 3.2: High-speed video from a single-shot rolling shutter image captured by a lensless computational camera. Each row of the recorded image, **b**, is captured at a unique time and contains information about nearly all scene points due to the inherent multiplexing of our lensless imager. The optics and exposure process can be described by a linear forward model, **A**, which is used to solve for the time sequence of 2D images (video), **v**, via non-negative least squares with a 3D gradient sparsity penalty, $\|\nabla_{xyt}\mathbf{v}\|_1$, weighted by τ. Each frame of the raw 33 fps recording is expanded to 140 frames giving an effective frame rate of 4,545 fps.

Rolling shutter model

In general, the exposure at each point on the sensor, $L(x, y)$, can be modeled as a temporal integral,

$$L(x,y) = \int_0^\infty S_t(t|x,y) \cdot \tilde{v}(x,y,t)dt, \tag{3.1}$$

where $\tilde{v}(x, y, t)$ represents the time-varying optical intensity on the sensor, and $S_t(t|x, y) \in \{0, 1\}$ is a 3D indicator, the *shutter function*, that encodes the temporal exposure window at each (x, y) position. While our approach could be generalized to different exposure patterns, we focus on rolling shutter due to its ubiquity. Rolling shutter is a column-parallel approach in which each row of pixels exposes for T_e seconds, beginning at a delay, T_l, after the previous row began (typically tens-of-microseconds). Because rolling shutter records row-by-row, we drop the x-dependence of the shutter function, denoting it as $S_t(t|y)$ for the remainder of the paper. At any given instant, a small band of $N_l = T_e/T_l$ rows is actively recording

photons. For a sensor with pixel size Δ, this is depicted in Fig. 3.3, with red indicating where $S_t(t|y) = 1$. Our goal is to spatially multiplex scene information into the exposure band at each time point, which enables each band to produce a frame of the final video, achieving frame rates equal to $1/T_l$ fps.

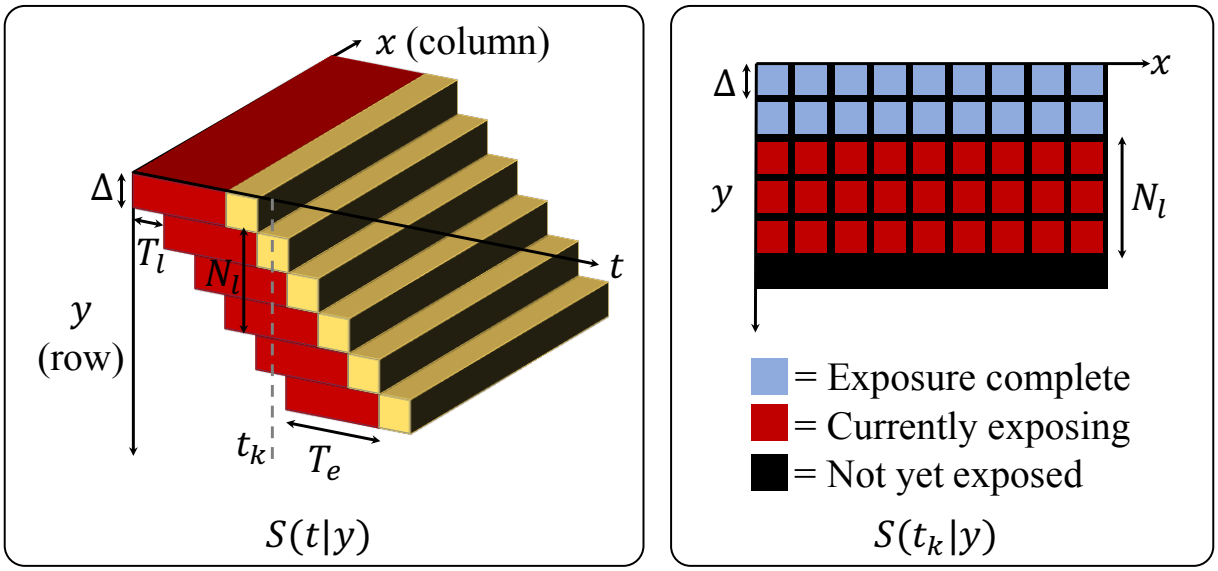


Figure 3.3: (Left) Spatio-temporal illustration of the rolling shutter function $S_t(t|y)$ for a sensor with pixel size Δ and exposure time T_e. Red depicts active exposure, and gold is the readout time. (Right) A slice through S_t at time t_k. Each row begins exposing T_l seconds after the previous row begins, with red representing actively exposing rows, and blue representing completed rows. The number of rows simultaneously exposed is $N_l = T_e/T_l$, which in this example is 3. For simplicity, we choose T_e such that N_l is an integer.

Lensless imaging model

In order to achieve the desired multiplexing, we use a simple lensless architecture (see Fig. 3.4) that employs a diffuser – a pseudorandom phase optic – as a computational imaging element [11, 8]. The system comprises a diffuser placed a distance d_0 from the rolling shutter sensor, with the scene at distance d_i from the diffuser. An aperture placed on the diffuser ensures that the resulting Point Spread Function (PSF) is shift-invariant, and enables simple calibration [11, 8]. For magnification $m = d_i/d_0$, the sensor plane intensity can be modeled by convolving the magnified scene intensity, $\boldsymbol{v}(x/m, y/m, t)$, with $\mathbf{h}(x, y)$, the on-axis PSF [47]:

$$\tilde{v}(x, y, t) = \boldsymbol{v}\left(\frac{x}{m}, \frac{y}{m}, t\right) \overset{(x,y)}{*} \mathbf{h}(x, y), \tag{3.2}$$

where $\overset{(x,y)}{*}$ denotes linear convolution over (x, y). The diffuser's PSF fills nearly the entire sensor with a pseudorandom caustic intensity pattern that is unique for each shift. This high

degree of spatial multiplexing is key to how our system works, enabling any horizontal slice of $\tilde{v}(x, y, t)$ to contain information about nearly all (x, y) positions in the scene.

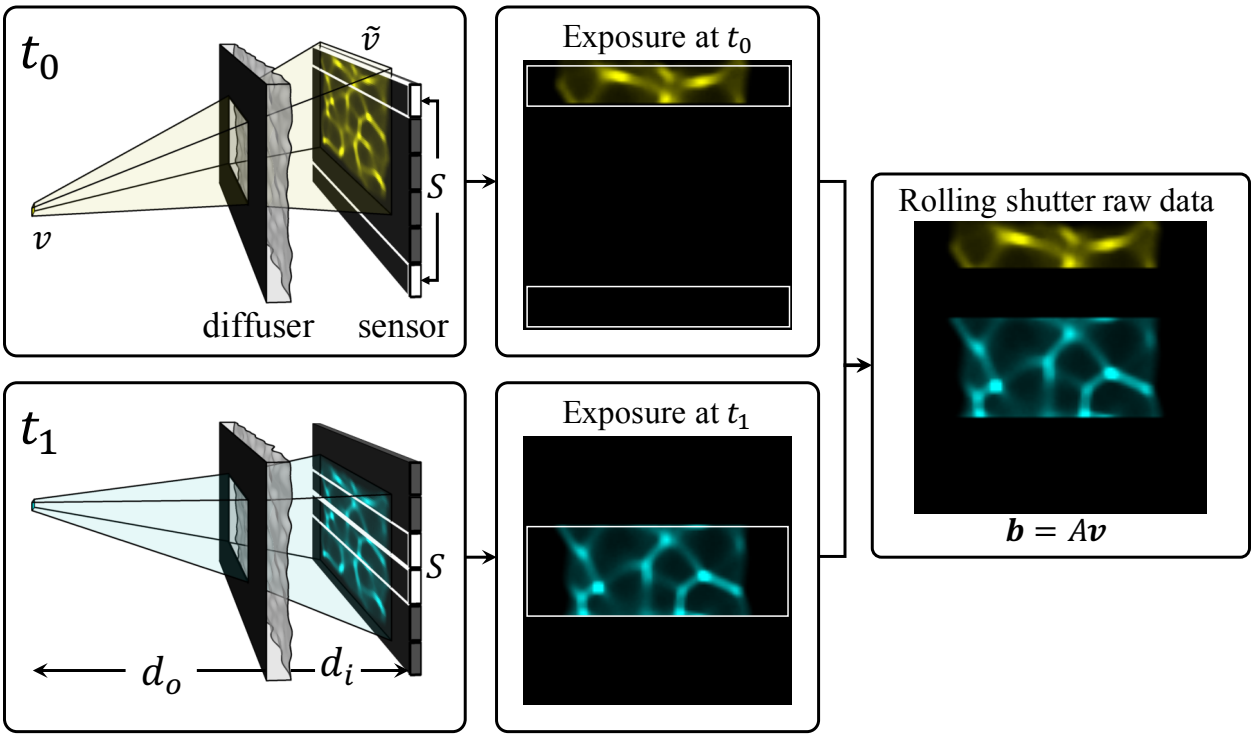


Figure 3.4: Image formation for a time-varying scene with two point sources (one yellow, one blue) flashing at unique y locations and times t_0 and t_1. (Left) Data measurement at times t_0 and t_1, with the time varying optical intensity, $\tilde{v}(x, y, t_i)$ rendered on the sensor, and dual shutter function $S_t(t_i|y)$ outlined in white. (Middle) The instantaneous exposure $S_t(t_i|y) \cdot \tilde{v}(x, y, t_i)$, is shown for each point source. (Right) The captured rolling shutter image is their sum. Due to the spatially-multiplexed optics, nearly all scene points project information into $S_t(y, t)$. This provides enough information to recover a video from a single image by solving an inverse problem.

Combining lensless and rolling shutter models

To solve for the video, we need a discrete forward model. We treat the measurement as a vector of samples taken from the continuous exposure $L(x, y)$: $\mathbf{b}[i, j] = L(j\Delta, i\Delta)$, where i and j index the sensor rows and columns, respectively. This leads to a discretized (magnified) scene, denoted $\mathbf{v}$, on a 3D spatio-temporal grid with lateral spacing Δ. The temporal spacing is T_l, as discussed in section 3.4. This leads to the linear discrete forward model:

$$\mathbf{b} = \sum_{k=0}^{K-1} \overline{S_t}[i, k] \cdot \left(\mathbf{h}[i, j] \overset{[i,j]}{*} \mathbf{v}[i, j, k] \right) \tag{3.3}$$

$$= \mathbf{A}\mathbf{v} \tag{3.4}$$

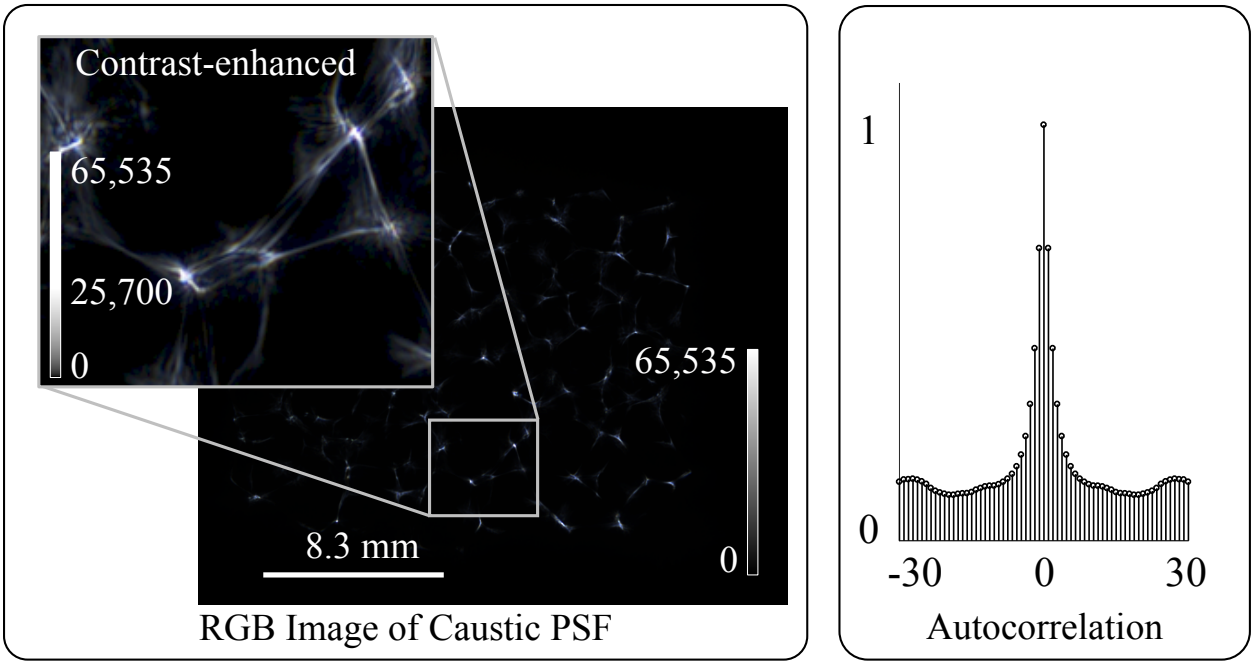


Figure 3.5: Left: 16-bit RGB image of the diffuser's caustic point spread function (PSF) for a white LED point source a distance 830 mm from the diffuser. A contrast stretched crop ($\gamma = 0.5$) is shown inset to show the structure of the caustics. Right: A slice from the normalized autocorrelation of the green channel showing a sharp main peak and relatively low side lobes, making this pattern suitable for compressed sensing.

where $\overset{[i,j]}{*}$ represents discrete linear 2D convolution over the spatial dimensions, $\overline{S_t}[i,k] = S_t(kT_l|i\Delta)$ is the discrete shutter function, and K is the number of recovered frames. Note that for global shutter, this would be a cropped convolution identical to [11, 72], but here we absorb the crop into the definition of $\overline{S_{tk}}[i]$. This linear forward model, denoted $\mathbf{A}$ in matrix form, is depicted in Fig. 3.4.

Video Recovery

To recover a video from a single rolling shutter measurement, we must solve an underdetermined linear inverse problem. For a dual-shutter camera such as ours, each symmetric pair of rows in the measurement corresponds to a frame in the reconstruction, so we recover approximately $K = M/2$ frames from a single $M \times N$ capture. The diffuser produces pseudorandom noise-like measurements, so our system fits within the framework of compressed sensing (as demonstrated in [11]). Hence we can solve the underdetermined problem for sparsely-represented scenes using $\ell 1$ minimization. If, as in Chapter 2, we attempt to recover each frame independently using a simple pixel erasure model, we get complete failure in reconstruction. Figure 3.6 shows the failure of reconstructing an image from two bands, comprising 5% of the total sensor pixels. This performs far worse than using morehtna 3× fewer pixels with random erasure. To resolve this, we solve jointly for the entire video, which enables us to impose additional temporal priors. Specifically, we use a weighted 3D total

variation (3DTV) prior on the scene, so the reconstructed video, $\mathbf{v}^*$, can be written as the solution to:

$$\mathbf{v}^* = \arg\min_{\mathbf{v}\geq 0} \frac{1}{2} \|\mathbf{A}\mathbf{v} - \mathbf{b}\|_2^2 + \tau \left\|\nabla_{xyt}\mathbf{v}\right\|_1, \tag{3.5}$$

where $\nabla_{xyt} = \begin{bmatrix}\nabla_x & \nabla_y & \alpha\nabla_t\end{bmatrix}^\intercal$ is the matrix of forward finite differences in the x, y, and t directions. We include an additional tuning parameter, α, that weights the temporal gradient sparsity penalty relative to the spatial dimensions (typically set between 5 and 30). We use FISTA [15] with the weighted anisotropic 3DTV proximal operator, implemented using parallel proximal methods according to [66]. For computational efficiency, we never instantiate the matrix $\mathbf{A}$ explicitly, but instead compute the matrix-vector products $\mathbf{A}(\cdot)$ and $\mathbf{A}^H(\cdot)$ using a combination of zero-padding, FFT-based convolutions, and cropping. Each color channel of the video is processed separately, using the corresponding color from the calibrated PSF. This inherently compensates for much of the chromatic aberration in the system.

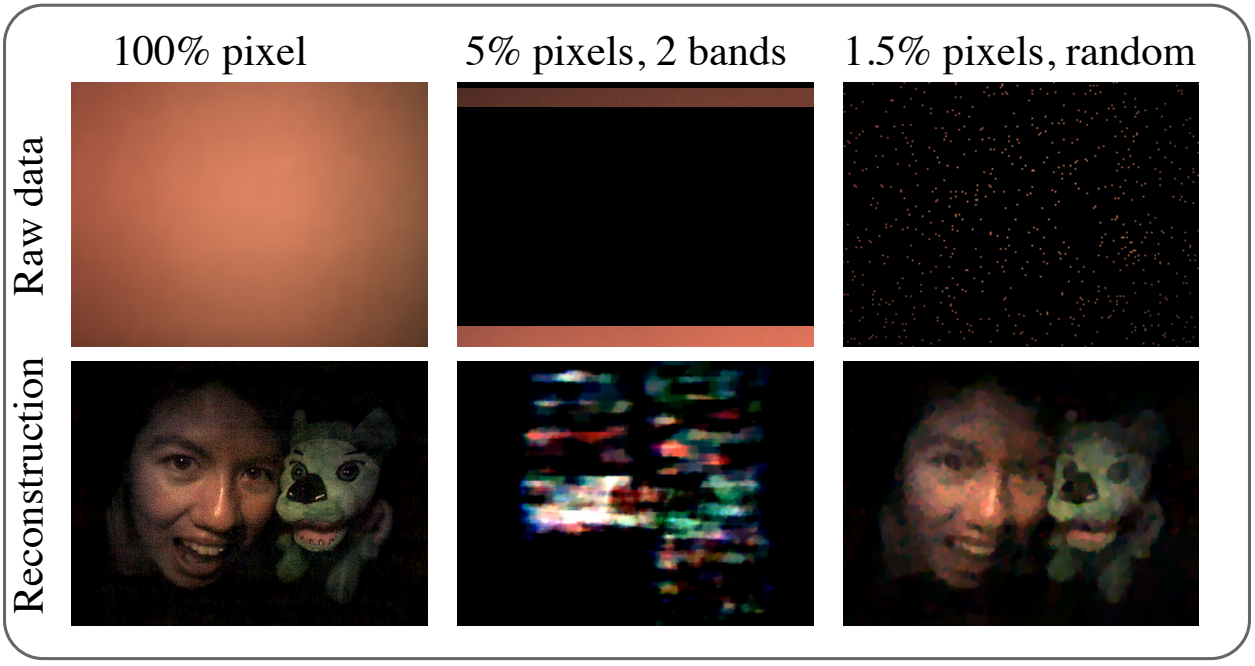


Figure 3.6: Comparing erasure patterns. With structured erasure such as is present in a rolling shutter camera, a simple frame-by-frame pixel erasure model fails compared to randomly erasing pixels.

3.3 Experiments

System Design

We built our prototype around a PCO Edge 5.5 sCMOS sensor, set to slow-scan rolling shutter mode. The dual shutter reads simultaneously from the top and bottom of the sensor.

Our homemade diffuser consists of randomly spaced lenslets. Because the lenslets concentrate light into sharp points, random lenslets have been shown to perform well in low-light situations [71], as is typical with high-speed imaging. Additionally, the uniformly random lateral placement of the lenslets ensures that each scene point produces a unique pattern on the sensor, and contributes a similar amount of light to each exposure band. This is not true near the edge of the sensor, as discussed in Section 3.4.

We fabricate our random lenslet diffusers using the molding process outlined in Section 3.4. Each lenslet comprising the diffuser has a focal length of 12.7 mm, yielding an approximately 30° by 40° (width-by-height) half field-of-view (FoV), which is reasonable for photographic scenes [72]. The system is calibrated using a single image of a white point source placed in the scene. Figure 3.5 shows a 16-bit color image of the PSF along with its 2D autocorrelation.

Experimental results

To test our system, we captured a variety of dynamic scenes. The raw data is downsampled by either 4× or 8× to match the expected temporal bandwidth (see Section 3.4). Videos are reconstructed at $640 \times 540 \times 140$ voxel grid for 4× downsampling, or $320 \times 270 \times 140$ for 8×. In both cases the video spans 30.8 milliseconds. Two example reconstructions are shown in Fig. 3.7. The first is a tennis ball dropping into a hand. The second is a green foam dart ricocheting off of an apple placed on a text book. In both cases, motion is clearly visible with good temporal detail present (see Supplementary Videos [9]). Due to system geometry, the outer sensor rows are relatively insensitive to the center of the object, degrading the quality of the first 30-40 frames. This is not a fundamental limit of our approach, but is rather a consequence of our implementation (see Sec. 3.4 for more discussion).

3.4 Analysis and Discussion

In this section, we analyze the temporal behavior of the system, showing that the temporal frequencies are band-limited by the exposure time. This motivates the design choices of our prototype, including the diffuser, exposure time, and use of binning (downsampling).

Temporal resolution

Next, we analyze the temporal frequency content of the measurements to validate temporal resolution. Intuitively, short exposure times are required to achieve high temporal resolution. We will show that, because our system is only compressive in space, its temporal resolution is

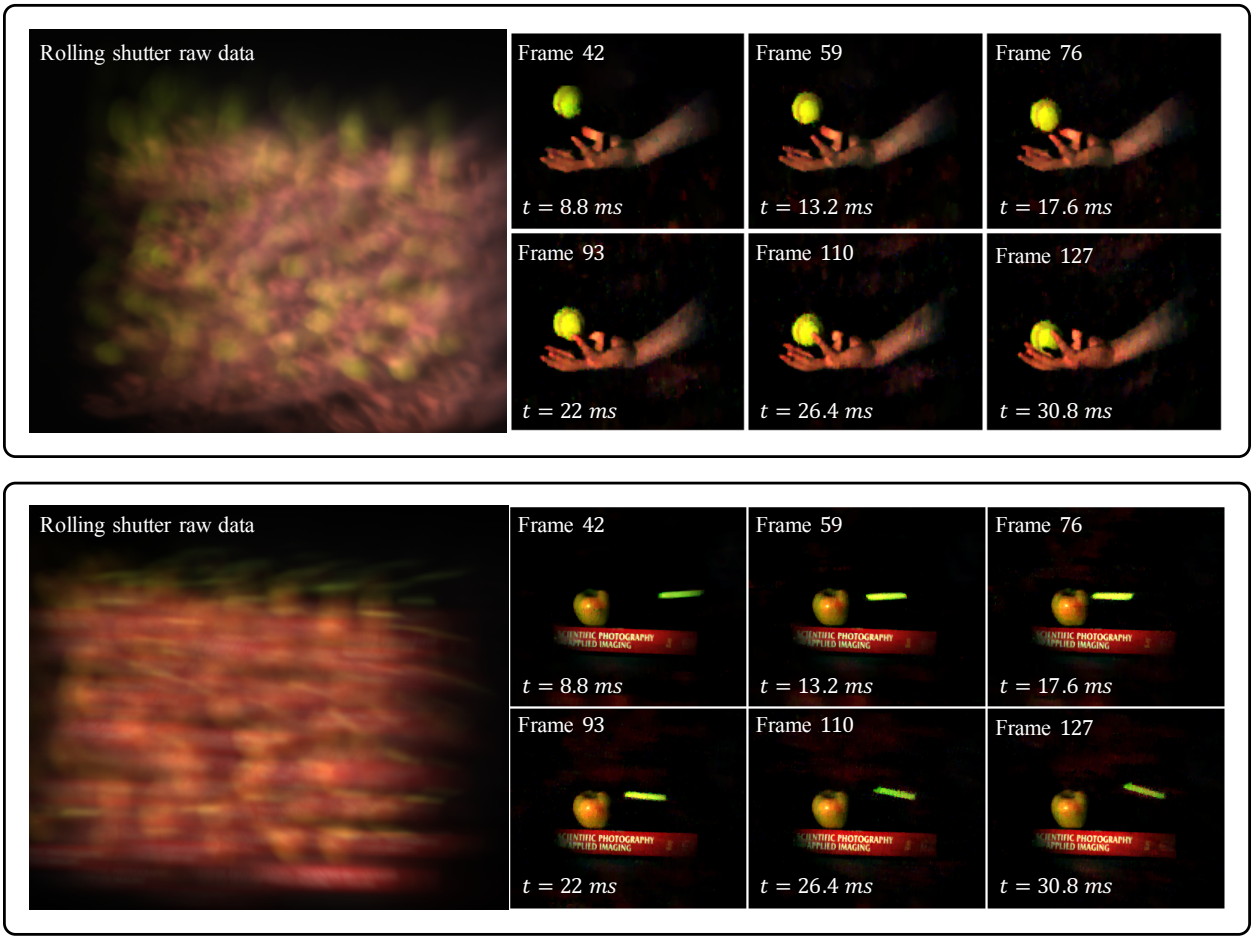


Figure 3.7: Experimental videos reconstructed from single-shot images (with 660 μs exposure). The top example shows a tennis ball falling into a hand, reconstructed with with $8x$ downsampling, and cropped to the center 135 $\times$ 160 pixels (see Supplementary Video 1 [9]). The bottom example shows a green foam dart ricocheting off an apple with 4$\times$ downsampling, cropped to 270 $\times$ 320 (see Supplementary Video 2 [9]). In both, the raw captured data is shown on the left, with a few frames from the reconstructed video shown at right. The final result contains 140 frames.

Nyquist limited, with an inherent band-limit set by the exposure time T_e, and the sampling rate determined by the line time, T_l. To show this, we begin by writing an expression for $S_t(t|y)$. As depicted in Fig. 3.3, $S_t(t|y)$ is a 1D temporal rectangular window of width T_e seconds, offset by T_l seconds per row:

$$S_t(t|y) = \text{rect}\left[\frac{t - \frac{T_e}{2} - \lfloor y/\Delta \rfloor T_l}{T_e}\right], \tag{3.6}$$

where $\lfloor y/\Delta \rfloor$ represents the row index. Substituting this into the continuous model for rolling

shutter acquisition, Eq. 3.1:

$$L(x,y) = \int_{-\infty}^{\infty} \text{rect}\left[\frac{t - t_c(y)}{T_e}\right] \tilde{v}(x,y,t)dt, \tag{3.7}$$

where we define $t_c(y) := \frac{T_e}{2} + \lfloor y/\Delta \rfloor T_l$ for compactness. Upon inspection, we see that this is a 1D convolution in the time dimension between the time-varying intensity at the sensor, $\tilde{v}(x,y,t)$, and a rectangular window of width T_e. The result of the convolution is evaluated along the slice of 3D space-time defined by $(x,y,t) = (x,y,t_c(y))$:

$$L(x,y) = \left[\tilde{v} \overset{t}{*} \text{rect}\left(\frac{t}{T_e}\right)\right]\Bigg|_{\left(x,y,t_c(y)\right)}. \tag{3.8}$$

This captures both the temporal band-limiting inherent in the exposure process as well as the mapping from time to row. Next we substitute Eq. 3.2, the expression for the spatially-multiplexed video, into Eq. 3.8:

$$\begin{aligned} L(x,y) &= \left\{\left[\mathbf{h} \overset{(x,y)}{*} v_g\right] \overset{t}{*} \text{rect}\left(\frac{t}{T_e}\right)\right\}\Bigg|_{\left(x,y,t_c(y)\right)} \\ &= \left\{\mathbf{h} \overset{(x,y)}{*} \left[v_g \overset{t}{*} \text{rect}\left(\frac{t}{T_e}\right)\right]\right\}\Bigg|_{\left(x,y,t_c(y)\right)}, \end{aligned} \tag{3.9}$$

where $v_g = \boldsymbol{v}(x/m, y/m, t)$ and the convolutions have been reordered, associating the temporal low-pass filter with the input signal. This shows that, while we are multiplexing in space, the temporal information in the system is band-limited by the pixel exposure time.

Finally, we introduce sampling. As shown in Section 3.2, the measured image is generated by sampling $L(x,y)$ on a grid of spacing Δ. Applying this sampling to the arguments of Eq. 3.9, we get $t_c(y = i\Delta) = T_l\lfloor i\Delta/\Delta \rfloor + T_e/2 = T_l i + T_e/2$. In other words, due to the implicit mapping of time to space, the rolling shutter effectively samples at a rate of $1/T_l$ Hz. Hence we expect to avoid temporal aliasing when $T_e > 2T_l$, even if the scene contains faster dynamics. This is also why, as discussed in Section 3.2, we discretize the video on a temporal grid of spacing T_l.

For our sensor, the minimum exposure time is 500 μs, with a maximum line time of 27.5μs. This would result in significant temporal oversampling, which is computationally wasteful. Thus, in practice, we use a combination of lateral downsampling of the raw data and temporal binning of the reconstruction to maintain inter-frame times of 220μs (4,545 fps), which better matches the minimum exposure time. Hence we expect to observe dynamics up to 2 kHz at best. Note that our reconstruction is highly nonlinear, relying heavily on nonnegativity and 3DTV denoising. As a result, this analysis represents only an upper bound to the frequencies we can hope to recover. In practice, measurement noise, calibration error, and regularization reduce performance (see Fig. 3.8).

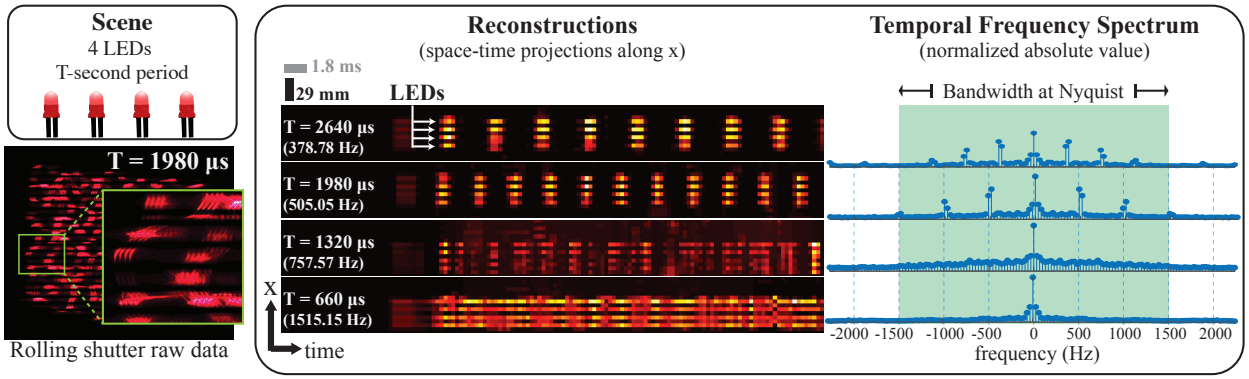


Figure 3.8: Resolution analysis using a sample consisting of a linear array of 4 LEDs, pulsed synchronously. We vary the pulse frequency of all four simultaneously. (Left) The raw data (660 μs exposure time) contains 4 copies of the caustic PSF pattern, each shifted in the horizontal direction according to each LED's spatial position, and the temporal patterns modulate the caustics in the y-direction. (Middle) $x-t$ projections of the reconstructed video. As expected, the performance degrades for the LEDs with shorter pulse periods, up to the theoretical limit of 660 μs predicted by Eq. 3.8. (Right) Temporal power spectra of the projections, clearly showing peaks in the time-direction moving as the LED frequency varies.

Resolution validation

As experimental validation of spatial and temporal resolution, we use a linear array of 4 LEDs flashing in unison with variable frequency square waves. We space the LEDs at the minimum separation resolvable by our system, which we establish empirically by varying the spacing until the LEDs are barely resolved in the reconstructions (6 mm separation at a distance 830 mm from the diffuser, or 0.4° angular resolution). We use an exposure time of $T_e = 660\mu s$, so $N_l = 3$ rows are exposing in each band. This should result in maximum frequency of $1,515$ Hz.

This dynamic scene can be expressed as $\boldsymbol{v}(x,y,t) = u(x,y)\cdot f(t)$, where $u(x,y)$ represents the 2D distribution of LEDs, and $f(t)$ is the modulating waveform. For such an object, the intensity inside the camera body will be $\tilde{v}(x,y,t) = f(t)\cdot(\mathbf{h}(x,y) * u(x/m, y/m))$. Plugging this into Eq. 3.8, we see that the continuous exposure at the sensor will be

$$L(x,y) = \left(\mathbf{h} \overset{(x,y)}{*} u_g\right) \cdot \left[f(t) \overset{t}{*} \text{rect}\left(\frac{t}{T_e}\right)\right]\Bigg|_{t=t_c(y)}, \tag{3.10}$$

where $u_g = u(x/m, y/m)$. Therefore we expect the measurement to look like the 2D scene convolved with the PSF and modulated in the y-direction by the low-pass filtered waveform. Figure 3.8 shows raw data from our experimental system. Because the 2D scene is 4 point sources in a line, this appears as 4 laterally shifted copies of the PSF, periodically modulated in the y-direction, as expected.

While our analysis provides a bound, experimental errors and nonlinear reconstruction can further deteriorate performance. To test how close we get to the limit, we recorded measurements with LED pulse rates varied from $2,640\mu s$ (378.78 Hz) to $660\mu s$ (1,515.15 Hz), the highest frequency predicted by the theory. The results are shown in Fig. 3.8. On

the left is a raw measurement with temporal period $T = 1,980\ \mu s$ (505 Hz). A strong envelope is clearly visible, modulating the measurement with a period of $T/T_l = 9$ pixels in the y-direction. In the reconstructions, we can clearly resolve all 4 LEDs spatially in all cases. At lower frequencies, the pulses are well resolved in time, with the harmonic structure of the square waves visible in the power spectra. As the period decreases, the temporal contrast reduces, with 660 μs period being totally unresolved.

For comparison, to record the same dynamic scene with LEDs pulsing at $T = 1980\mu s$ using global shutter would require 30 frames at greater than 1,010 fps. Within our system's sample budget of $270 \times 320 = 86,400$ samples, each frame from the corresponding global shutter system would only contain 49×58 pixels. This is a $6\times$ degradation in lateral resolution compared to what our compressive scheme achieves experimentally. Hence, at least for sparse scenes, the compressive approach surpasses a direct sampling scheme.

Diffuser fabrication

Based on simulations, we found that a diffuser consisting of randomly-spaced lenslets performed better than off-the-shelf diffusers [71]. To fabricate, we repeatedly indent a copper block with a ball bearing of radius 7 mm. The indentations are made at random spacing (by hand) over an area larger than the 14.04×16.64 mm size of the PCO Edge 5.5 sensor. The result is a mold that is piecewise spherical with curvature matching the ball bearing. We use this block as a mold for UV-cured epoxy (Norland 61), with microscope slide on the top surface to ensure flatness. We then cure the epoxy and separate it from the mold. The epoxy has refractive index 1.56, yielding a diffuser with random lenslets of approximate focal length $f = 12.5mm$. We mask the diffuser with a 13×15.5 mm rectangular aperture, then mount the diffuser approximately 12.4 mm from the sensor. This results in magnification of $-.015\times$ for objects placed 830 mm away.

Artifacts due to time-varying FoV

Given the structured sampling pattern of a rolling shutter sensor, we can reason about the system FoV geometrically. The set of scene points visible to each sensor pixel is determined by projecting rays from the pixel through the aperture. From this simple picture, we see that each pixel has a unique FoV. Because the rolling shutter pattern reads a band of rows simultaneously, this effectively means the FoV is varying with time: early in the exposure, the outer sensor rows are active, and cannot see the center of the FoV, while the inner rows (later frames) can. Because the sensor is blind to the on-axis points early in the exposure, these frames are determined via the regularizer. This explains the wiping artifact present in our videos in the early frames. If we were to use a single-shutter sensor, the effect would be more pronounced, as the FoV would sweep across the scene. This issue could be alleviated by distributing the active pixels more evenly across the sensor plane or by removing the aperture. In the current system, we simply discard the early frames of the video. In future builds, we could remove or enlarge the aperture, though this will preclude single-image calibration, and will lead to our shift-invariant lensless model breaking down at high angles.

Such artifacts are correctable [71], but lead to much slower processing times, and so we leave this for future work.

Limitations and future work

For our prototype, there are two main limiting factors: the quality of the optics, and the CMOS sensor dynamics. Because the sensor's minimum exposure time limits the maximum usable frame rate, sensors with shorter exposure will perform better. Additionally, to match the line time to the exposure time, we would like to freely adjust the sensor's line timing; however, our sensor does not allow this. This leads us to use spatial downsampling as a workaround to effectively increase the line time to better match the band-limit.

The second limiting factor is the quality of the diffuser. While our homemade diffusers are sufficient for proof-of-concept work, the resulting optics is fairly low quality, and the process is not well controlled. We can achieve the target focal length, but the focal spots (see Fig. 3.5) are extremely aberrated. This works well with the downsampling approach, as the caustics are not sharp enough to warrant using the full resolution sensor. However, to push our approach to the limit, we would need optics that can produce multiplexed PSFs with very sharp features. Coupled with a sensor capable of short exposures (on the order of the line time), our proposed architecture could achieve extremely high spatio-temporal resolution. For example, our current sensor can operate with line times as fast as 9.17 μs, or over 100,000 fps.

Another limiting factor is the reduced measurement signal-to-noise caused by the multiplexing. Pushing this system to 100,000 fps would require exposure times shorter than 10 μs. Because the light from each point is distributed across the sensor with only a few pixels being recorded in each frame, this would require extremely bright scenes. Additionally, the combination of multiplexing and regularized reconstruction generally reduces the dynamic range of the recovered image, further limiting the method to high contrast scenes.

As with most compressed sensing systems, it is difficult to validate the performance in general, since it is object dependent. We know from prior work [11] that the performance degrades with scene complexity, and we observe this effect. While it does work for dense scenes, we require higher regularization, effectively limiting the usefulness for scenes that do not fit a gradient sparsity prior well. Introducing more sophisticated priors could mitigate this issue.

Our reconstructions are computationally expensive relative to a direct sampling approach. Achieving extremely short exposures and the fastest line time possible would require not downsampling the measurement, leading to a computationally expensive 3D inverse problem at gigavoxel scale.

While we chose a dual-shutter camera for the experimental validation in this work, exploring the use of different programmable exposures could be extremely fruitful. Demonstrating the system with the more commonplace single shutter CMOS architecture would make it widely accessible, as the only other required equipment is a diffuser. Our current sensor also has a delay far longer than the line time between each sequential frame, preventing us from stringing together sequential frames into longer videos without a gap (see Supplementary

Video 4 [9]). A sensor that streamed continuously could alleviate this. It could also be useful to couple multiplexing optics with randomized sensor read patterns [138], as this will certainly lead to better video recovery.

Conclusion

In conclusion, we have demonstrated that a spatially-multiplexing lensless camera can turn rolling shutter from a detriment into an advantage. We built a proof-of-concept system that resolves $1,500$ Hz dynamics at a frame rate of $4,545$ frames per second. We derived a theoretical temporal resolution bound based on our forward model, and confirmed our theoretical predictions with experiment. Our system relies on compressed sensing to solve an extremely underdetermined problem. We successfully observed samples with space-time bandwidth product far exceeding what could be observed with a direct sampling approach. Finally, we demonstrated our approach on a variety of fast-moving scenes, reliably recovering high speed videos from single rolling shutter images.

Acknowledgements

The authors would like to thank The Moore Foundation, DARPA, and Bakar Fellows. This material is based upon work supported by the National Science Foundation under Grant No. 1617794. This work has also been supported by an Alfred P. Sloan Foundation fellowship. Emrah Bostan's research is supported by the Swiss National Science Foundation (SNSF) under grant P2ELP2 172278.

Chapter 4

Lensless 3D imaging

This is work done jointly with Grace Kuo, Reinhard Heckel, Emrah Bostan, Ben Mildenhall, Ren Ng, and Laura Waller and is based on [11]

4.1 Introduction

Because optical sensors are 2D, imaging 3D objects requires projection to 2D in such a way that the 3D information can be recovered. Scanning and multi-shot methods can achieve high spatial resolution 3D imaging, but sacrifice capture speed [33, 57]. In contrast, single-shot 3D methods are fast but may have low resolution or small field-of-view (FoV) [24, 105]. Often, bulky hardware and complicated setups are required. Here, we introduce a compact and inexpensive single-shot lensless optical system that is capable of 3D imaging. We show how it can reconstruct a large number of voxels by leveraging compressed sensing.

Our lensless imager, DiffuserCam, encodes the 3D intensity of volumetric objects in a

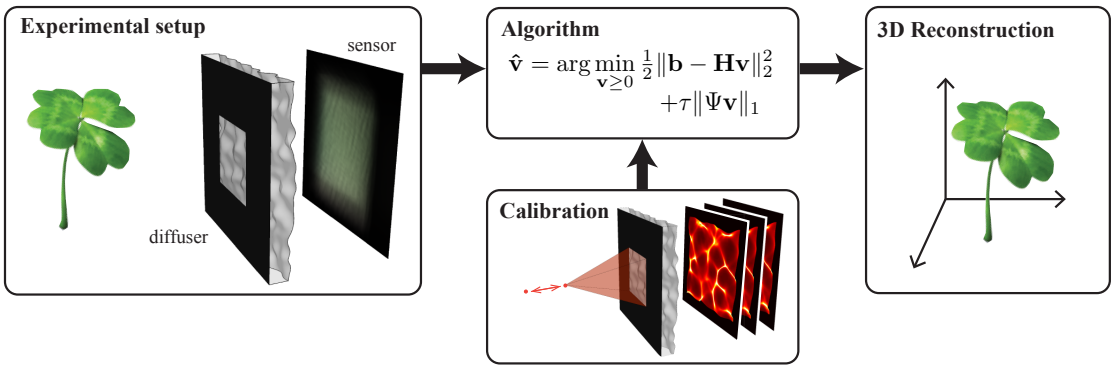


Figure 4.1: DiffuserCam setup and reconstruction pipeline. Our lensless system consists of a diffuser placed in front of a sensor (bumps on the diffuser are exaggerated for illustration). The system encodes a 3D scene into a 2D image on the sensor. A one-time calibration consists of scanning a point source axially while capturing images. Images are reconstructed computationally by solving a nonlinear inverse problem with a sparsity prior. The result is a 3D image reconstructed from a single 2D measurement.

single 2D image. The diffuser, a thin phase mask, is placed a few millimeters in front of an image sensor. Each point source in 3D space creates a unique pseudorandom caustic pattern that covers a large portion of the sensor. Because of this, compressed sensing algorithms can be used to reconstruct more voxels than pixels captured, provided that the 3D sample is sparse in some domain. We solve the inverse problem via a sparsity-constrained optimization procedure, using a physical model and simple calibration scheme to make the computation scalable. This allows us to reconstruct several orders of magnitude more voxels than related previous work [35, 87].

We demonstrate a prototype DiffuserCam system built entirely from commodity hardware. It is efficient to calibrate, does not require precise alignment, and is light efficient (as compared to amplitude masks). We reconstruct 3D objects on a grid of 100 million voxels (non-uniformly spaced) from a single 1.3 megapixel image. Our reconstructions show true depth sectioning, allowing us to generate 3D renderings of the sample.

Our system, like many computational cameras, uses a nonlinear reconstruction algorithm, resulting in object-dependent performance. To quantify, we experimentally measure the resolution of our prototype with different objects. We show that the standard two-point resolution criterion is misleading and should be considered a best-case scenario. To better explain the variable resolving power of our system, we propose a new local condition number analysis that is consistent with our experiments.

DiffuserCam uses concepts from lensless camera technology and imaging through complex media, integrated together via computational imaging design principles. Our proposed architecture and algorithm could enable high resolution, light efficient lensless 3D imaging of large and dynamic 3D samples in an extremely compact package. Such cameras will open up new applications in remote diagnostics, mobile photography and *in vivo* microscopy.

System Overview

DiffuserCam is part of the class of mask-based passive lensless imagers in which a phase or amplitude mask is placed a small distance in front of a sensor, with no main lens. Our mask (the diffuser) is a thin transparent phase object with smoothly varying thickness (see Fig. 4.1). When a temporally incoherent point source is placed in the scene, we observe a high-frequency pseudorandom caustic pattern at the sensor. The caustic patterns, termed Point Spread Functions (PSFs), vary with the 3D position of the source, thereby encoding 3D information.

To illustrate how the caustics capture 3D information, Fig. 4.2 shows simulations of the PSFs for a point source at different locations in object space. A lateral shift of the point source causes a lateral translation of the PSF [39, 39]. An axial shift of the point source causes (approximately) a scaling of the PSF. Hence, each 3D position in the volume generates a unique caustic pattern. The structure and spatial frequencies present in the PSFs determine our reconstruction resolution. By using a phase mask (which concentrates light better than an amplitude mask) and designing the system to retain high spatial frequencies over a large range of depths, DiffuserCam attains good lateral resolution across the volumetric field-of-view.

By assuming that all points in the scene are incoherent with each other, the measurement can be modeled as a linear combination of PSFs from different 3D positions. We represent this as matrix-vector multiplication:

$$\mathbf{b} = \mathbf{H}\mathbf{v}, \tag{4.1}$$

where $\mathbf{b}$ is a vector containing the 2D sensor measurement and $\mathbf{v}$ is a vector representing the intensity of the object at every point in the 3D FoV, sampled on a user-chosen grid. $\mathbf{H}$ is the forward model matrix whose columns consist of each of the caustic patterns created by the corresponding 3D points on the object grid. The number of entries in $\mathbf{b}$ and the number of rows of $\mathbf{H}$ are equal to the number of pixels on the image sensor, but the number of columns in $\mathbf{H}$ is set by the choice of reconstruction grid (discussed in Sec. 4.3). Note that this model does not account for partial occlusion of sources.

In order to reconstruct the 3D object, $\mathbf{v}$, from the measured 2D image, $\mathbf{b}$, we must solve (4.1) for $\mathbf{v}$. However, if we solve on a 3D reconstruction grid that corresponds to the full optical resolution of our system (measured in Sec. 4.34.3), $\mathbf{v}$ will contain more voxels than there are sensor pixels. In this case, $\mathbf{H}$ has more columns than rows, so the problem is underdetermined and we cannot uniquely recover $\mathbf{v}$ simply by inverting (4.1). To remedy this, we rely on sparsity-based principles [26]. We exploit the fact that many 3D objects are sparse in some domain, meaning that the majority of coefficients are zero after a linear transformation. We enforce this sparsity as a prior and solve the ℓ_1 regularized nonnegativity-constrained inverse problem:

$$\hat{\mathbf{v}} = \underset{\mathbf{v}\geq 0}{\operatorname{argmin}} \tfrac{1}{2}\|\mathbf{b} - \mathbf{H}\mathbf{v}\|_2^2 + \tau\|\Psi\mathbf{v}\|_1. \tag{4.2}$$

Here, Ψ maps $\mathbf{v}$ into a domain in which it is sparse ($\Psi\mathbf{v}$ is mostly zeros), and τ is a tuning parameter that adjusts the degree of sparsity. For objects that are sparse in voxels, such as fluorescent particles in a volume, Ψ is the identity matrix. In our results we show reconstruction of objects that are not sparse in voxels but are sparse in the gradient domain. Hence, we choose Ψ to be the finite difference operator and $\|\Psi\mathbf{v}\|_1$ to be the 3D Total Variation (TV) semi-norm [109]. In general, any linear sparsity transformation may be used (e.g. wavelets), but we utilize only identity and gradient representations in this work.

Equation (4.2) is the basis pursuit problem in compressed sensing [26]. For this optimization procedure to succeed, $\mathbf{H}$ must have distributed, uncorrelated columns. Since our diffuser creates high spatial frequency caustics that spread across many pixels in a pseudorandom fashion, any shift or magnification of the caustics leads to a new pattern that is uncorrelated with the original one (quantified in Supplementary Fig. S4). As discussed in Sec. 4.24.2 and 4.24.2, these properties allow us to reconstruct 3D images via compressed sensing.

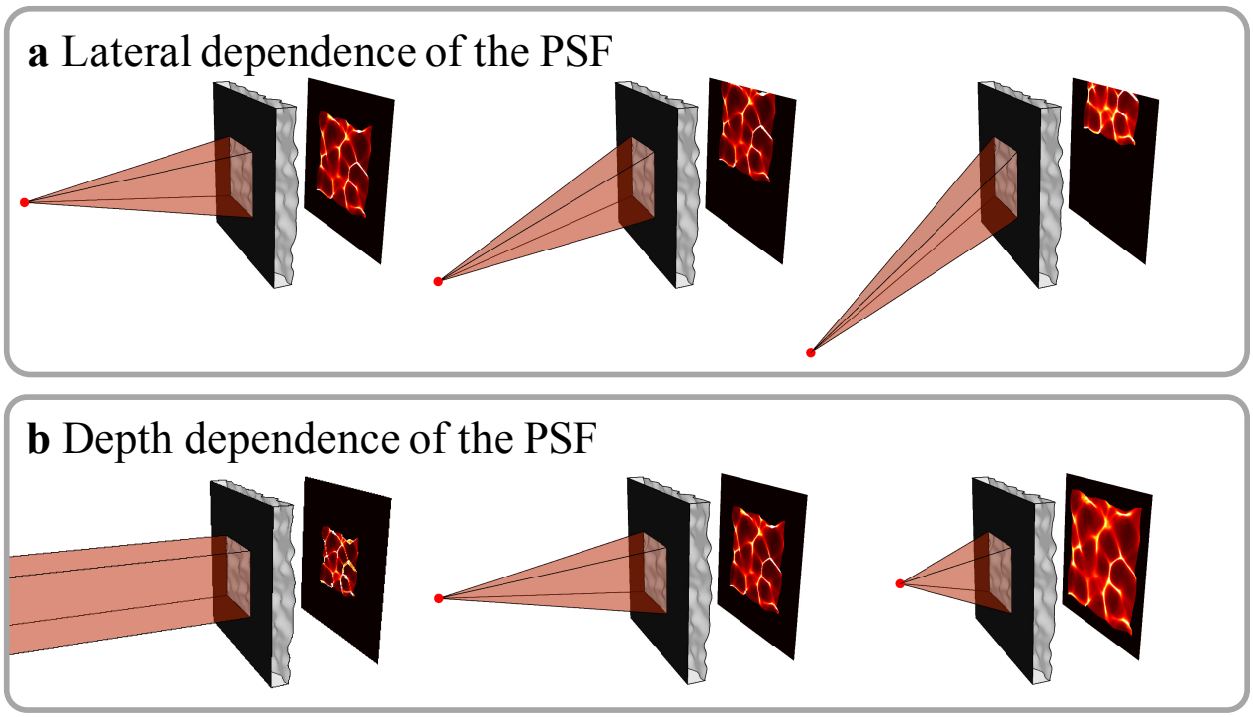


Figure 4.2: The caustic pattern shifts with lateral shifts of a point source in the scene and scales with axial shifts. (a) Ray-traced renderings of caustics as a point source moves laterally. For large shifts, part of the pattern is clipped by the sensor. (b) The caustics magnify as the source is brought closer.

4.2 Methods

System Architecture

The hardware setup for our prototype DiffuserCam (Fig. 4.3a) consists of an off-the-shelf diffuser (Luminit 0.5°) placed at a fixed distance in front a sensor (PCO.edge 5.5 Color camera, 6.5μm pixels). The diffuser has a flat input surface and an output surface that is described statistically as Gaussian lowpass-filtered white noise with an average spatial feature size of 140μm and average slope magnitude of 0.7° (see Supplementary Fig. S1). The convex bumps on the diffuser surface can be thought of as randomly-spaced microlenses that have statistically-varying focal lengths and f-numbers. The average focal length determines the distance at which the caustics have highest contrast (the *caustic plane*), which is where we place the sensor [8]. This distance, measured experimentally, is 8 mm for our diffuser. However, the high average f-number of the bumps (8mm/140μm=57) means that the caustics maintain high contrast over a large range of propagation distances. Therefore, the diffuser need not be placed precisely at the caustic plane (in our prototype, d=8.9mm). We also affix a 5.5×7.5mm aperture on the textured side of the diffuser to limit the support of the caustics.

Similar to a traditional camera, the sensor's pixel pitch should Nyquist sample the minimum features of the PSF. Since the f-number of the smallest bumps on the diffuser determine

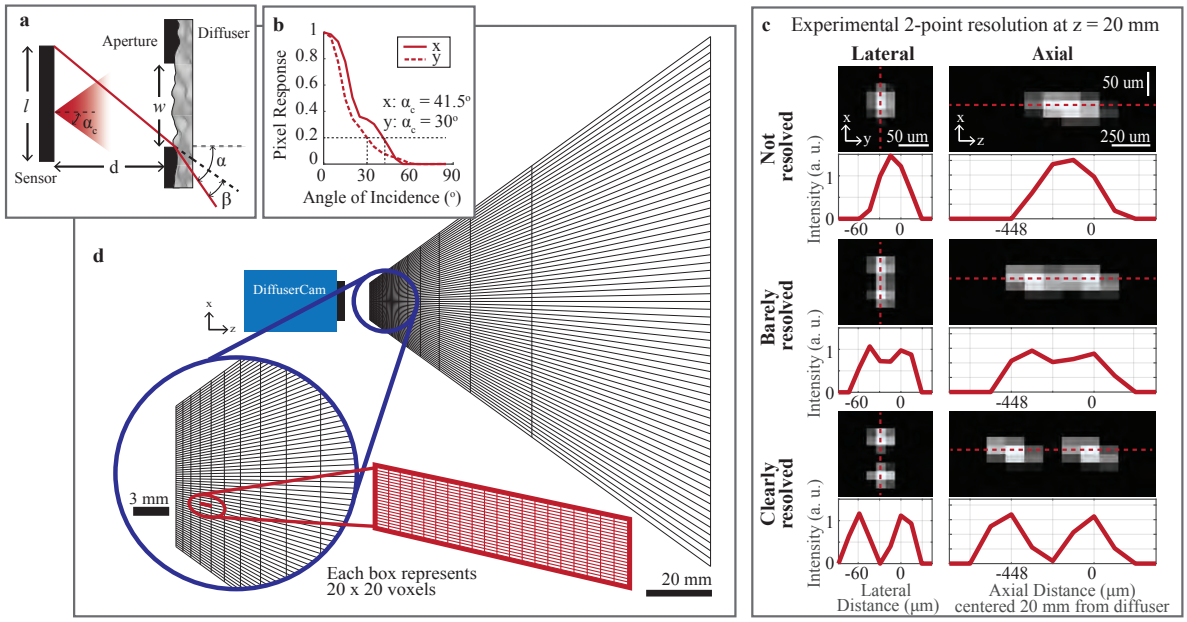


Figure 4.3: Experimentally determined field-of-view (FoV) and resolution. (a) System architecture with design parameters. (b) Angular pixel response of our sensor. We define the angular cutoff (α_c) as the angle at which the response falls to 20%. (c) Reconstructed images of two points (captured separately) at varying separations laterally and axially, near the z = 20 mm depth plane. Points are considered resolved if they are separated by a dip of at least 20%. (d) To-scale non-uniform voxel grid for 3D reconstruction. The chosen voxel grid is based on the system geometry and Nyquist-sampled two-point resolution over the entire FoV. For visualization purposes, each box represents 20×20 voxels, as shown in red.

the minimum feature size of the caustics, it will also set the lateral optical resolution. In our case, the smallest features generated by the caustic patterns are roughly twice the pixel pitch of our sensor, so we perform 2×2 binning on the data, yielding 1.3 megapixel images, before applying our reconstruction algorithm.

Convolutional Forward Model

Recovering a 3D image requires $\mathbf{H}$, the measurement matrix, which is extremely large. Measuring or storing the full $\mathbf{H}$ is infeasible, requiring millions of calibration images and multi-Terabyte-scale matrices. Instead, we rely on a depth-dependent convolution model outlined below to drastically reduce calibration and computational complexity.

The scene of interest is denoted $\mathbf{v}$, and is a set of point sources located at (x, y, z) on a non-Cartesian 3D grid. The relative radiant energy collected by the aperture from a source at $[x, y, z]$ is $\mathbf{v}[x, y, z]$, and the PSF from that voxel at pixel $[x', y']$ is denoted $h[x', y'; x, y, z]$. We model the measurement, $\mathbf{b}(x', y')$, as the sum of sensor contributions from every voxel within the 3D FoV:

$$\mathbf{b}(x', y') = \sum_{(x,y,z)} \mathbf{v}(x, y, z) h(x', y'; x, y, z). \tag{4.3}$$

This is equivalent to the matrix-vector multiplication $\mathbf{Hv}$ where each column of $\mathbf{H}$ is the PSF from the corresponding voxel. As before, a shift-invariance assumption greatly simplifies the evaluation of (4.3), which is illustrated in Fig. 4.2(a), and validated experimentally in Sec. 4.34.3. Defining the on-axis caustic pattern at depth z as $h(x', y'; z) := h(x', y'; 0, 0, z)$ and assuming magnification of $m = -1$, the off-axis caustic pattern is given by $h(x', y'; x, y, z) = h(x' - x, y' - my; z)$. Plugging into (4.3), the sensor measurement is then given by:

$$\begin{aligned} \mathbf{b}(x', y') &= \sum_z \sum_{(x,y)} \mathbf{v}(x, y, z) h(x' + mx, y' + my; z) \\ &= \mathbf{C} \sum_z \left[\mathbf{v} \left(\frac{-x'}{m}, \frac{-y'}{m}, z \right) * h \left(x', y'; z \right) \right]. \end{aligned} \tag{4.4}$$

This is the same cropped convolutional model used in Chapters 2 and 3, except now the contribution from every depth within the object contributes to the measurement. For an object discretized into N_z depth slices, the number of columns of $\mathbf{H}$ is N_z times larger than the number of elements in $\mathbf{b}$ (*i.e.* the number of sensor pixels), so our system is underdetermined.

The cropped convolution model provides three benefits. First, it allows us to compute $\mathbf{Hv}$ as a linear operator in terms of N_z images, rather than instantiating $\mathbf{H}$ explicitly (which would require petabytes of memory to store). In practice, we evaluate the sum of 2D cropped convolutions using a single circular 3D convolution, implemented with 3D FFTs, which scale well to large arrays (see Supplementary Material, Sec. 2C). Second, it provides a theoretical justification of our system's capability for compressed sensing; derivations in [70] show that translated copies of a random pattern provide close-to-optimal performance.

The third benefit of our convolution model is that it enables simple calibration. Rather than measuring the system response for every voxel (hundreds of millions of images), we only need to capture a single calibration image of the caustic pattern from an on-axis point source. Though the scaling effect described in Sec. 4.14.1 suggests that we could use only one image for calibrating the entire 3D space (by scaling it to predict PSFs at different depths), we obtain better results when we calibrate the PSF at each depth. A typical calibration thus consists of capturing images as a point source is moved axially. This takes minutes, but need only be performed once. The added aperture at the diffuser ensures that a point source at the minimum z distance generates caustics that just fill the sensor, so that the entire PSF is captured in each image (see Supplementary Fig. S2).

Inverse Algorithm

Our inverse problem is extremely large in scale, with millions of inputs and outputs. Even with the convolution model described above, using projected gradient techniques is extremely slow due to the time required to compute the proximal operator of 3D TV [16]. To alleviate this, we use the Alternating Direction Method of Multipliers (ADMM) [22] and derive a variable splitting that leverages the specific structure of our problem.

Our algorithm uses the fact that Ψ can be written as a circular convolution for both the 3D TV and native sparsity cases. Additionally, we factor the forward model in (4.4) into

a diagonal component, $\mathbf{D}$, and a 3D convolution matrix, $\mathbf{M}$, such that $\mathbf{H} = \mathbf{DM}$ (details in Supplementary Material). Thus, both the forward operator and the regularizer can be computed in 3D Fouier space. This enables us to use variable-splitting [5, 95, 4] to formulate the constrained counterpart of (4.2):

$$\begin{aligned} \hat{\mathbf{v}} = \underset{w\geq 0,u,v}{\operatorname{argmin}} \tfrac{1}{2}\|\mathbf{b} - \mathbf{D}v\|_2^2 + \tau\|u\|_1 \\ \text{s.t. } v = \mathbf{M}\mathbf{v}, u = \Psi\mathbf{v}, w = \mathbf{v}, \end{aligned} \tag{4.5}$$

where v, u, and w are auxiliary variables. We solve (4.5) by following the augmented Lagrangian arguments [100]. Using ADMM, this results in the following scheme at iteration k:

$$\begin{aligned} u^{k+1} &\leftarrow \mathcal{T}_{\frac{\tau}{\mu_2}}\left(\Psi\mathbf{v}^k + \eta^k/\mu_2\right) \\ v^{k+1} &\leftarrow (\mathbf{D}^\intercal\mathbf{D} + \mu_1 I)^{-1}\left(\xi^k + \mu_1\mathbf{M}\mathbf{v}^k + \mathbf{D}^\intercal\mathbf{b}\right) \\ w^{k+1} &\leftarrow \max\left(\rho^k/\mu_3 + \mathbf{v}^k, 0\right) \\ \mathbf{v}^{k+1} &\leftarrow (\mu_1\mathbf{M}^\intercal\mathbf{M} + \mu_2\Psi^\intercal\Psi + \mu_3 I)^{-1} r^k \\ \xi^{k+1} &\leftarrow \xi^k + \mu_1(\mathbf{M}\mathbf{v}^{k+1} - v^{k+1}) \\ \eta^{k+1} &\leftarrow \eta^k + \mu_2(\Psi\mathbf{v}^{k+1} - u^{k+1}) \\ \rho^{k+1} &\leftarrow \rho^k + \mu_3(\mathbf{v}^{k+1} - w^{k+1}), \end{aligned}$$

where

$$r^k = (\mu_3 w^{k+1} - \rho^k) + \Psi^\intercal\left(\mu_2 u^{k+1} - \eta^k\right) + \mathbf{M}^\intercal\left(\mu_1 v^{k+1} - \xi^k\right).$$

Note that $\mathcal{T}_\nu$ is a vectorial soft-thresholding operator with a threshold value of ν [137]. ξ, η and ρ are the Lagrange multipliers associated with v, u, and w, respectively. The scalars μ_1, μ_2 and μ_3 are penalty parameters which we compute automatically using the tuning strategy in [22]. A MATLAB implementation of our algorithm is available at [10].

Although our algorithm involves two large-scale matrix inversions, both can be computed efficiently and in closed form. Since $\mathbf{D}$ is diagonal, $(\mathbf{D}^\intercal\mathbf{D} + \mu_1 I)$ is itself diagonal, requiring complexity $\mathcal{O}(n)$ to invert using point-wise multiplication. Additionally, all three matrices in $(\mu_1\mathbf{M}^\intercal\mathbf{M} + \mu_2\Psi^\intercal\Psi + \mu_3 I)$ are diagonalized by the 3D discrete Fourier transform (DFT) matrix, so inversion of the entire term can be done using point-wise division in 3D frequency space. Therefore, its inversion has good computational complexity, $\mathcal{O}(n^3 \log n)$, since it is dominated by two 3D FFTs being applied to n^3 total voxels. We parallelize our algorithm on the CPU using C++ and Halide [107], a high performance programming language for image processing (see Supplementary Fig. S6 for runtime performance).

A typical reconstruction requires at least 200 iterations. Solving for 2048×2048×128=537 million voxels takes 26 minutes (8 seconds per iteration) on a 144-core workstation and requires 85 Gigabytes of RAM. A smaller reconstruction (512×512×128=33.5 million voxels) takes 3 minutes (1 second per iteration) on a 4-core laptop with 16 Gigabytes of RAM.

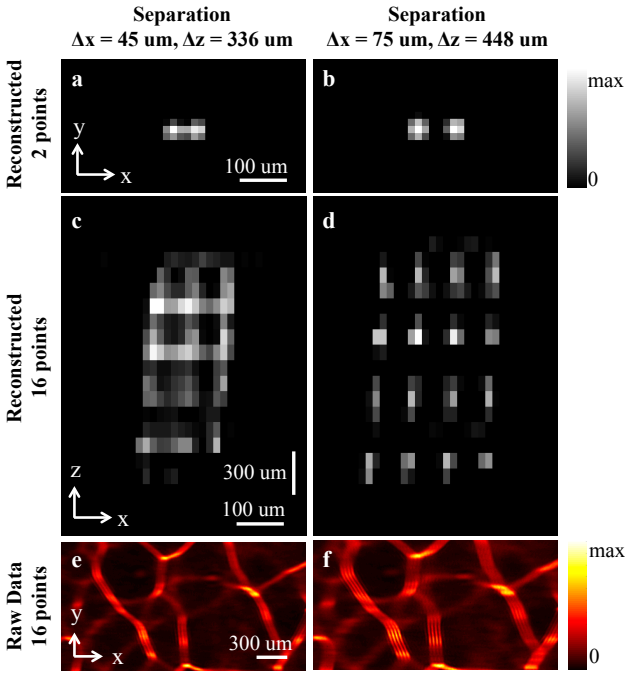


Figure 4.4: Our computational camera has object-dependent performance, such that the resolution depends on the number of points. (a) To illustrate, we show here a situation with two points successfully resolved at the two-point resolution limit $(\Delta x, \Delta z) = (45\mu m, 336\mu m)$ at a depth of approximately 20 mm. (c) However, when the object consists of more points (16 points in a 4×4 grid in the $x - z$ plane) at the same spacing, the reconstruction fails. (b,d) Increasing the separation to $(\Delta x, \Delta z) = (75\mu m, 448\mu m)$ gives successful reconstructions. (e,f) A close-up of the raw data shows noticeable splitting of the caustic lines for the 16 point case, making the points distinguishable. Heuristically, the 16 point resolution cutoff is a good indicator of resolution for real-world objects.

4.3 System Analysis

Unlike traditional cameras, the performance of computational cameras depends on properties of the scene being imaged (*e.g.* the number of sources). As a consequence, standard two-point resolution metrics may be misleading, as they do not predict resolving power for complex objects. To address this, we propose a new local condition number metric that better predicts performance. We analyze resolution, FoV and the validity of the convolution model, then combine these analyses to determine the appropriate sampling grid for our experiments.

Field-of-View

At every depth in the volume, the angular half-FoV is determined by the most extreme lateral position that contributes to the measurement. There are two possible limiting factors. The first is the geometric angular cutoff, α, set by the aperture size, w, the sensor size, l, and the distance from the diffuser to the sensor, d (see Fig. 4.3a). Since the diffuser bends light, we also take into account the diffuser's maximum deflection angle, β. This gives a geometric angular half-FoV at every depth of $l + w = 2d\tan(\alpha - \beta)$. The second limiting factor is

the angular response of the sensor pixels. Real-world sensor pixels may not accept light at the high angles of incidence that our lensless camera accepts, so the sensor angular response (shown in Fig. 4.3b) may limit the FoV. Defining the angular cutoff of the sensor, α_c, as the angle at which the camera response falls to 20% of its on-axis value, we can write the overall FoV equation as:

$$\text{FoV} = \beta + \min[\alpha_c, \tan^{-1}(\tfrac{l+w}{2d})]. \tag{4.6}$$

Since we image in 3D, we must also consider the axial FoV. In practice, the axial FoV is limited by the range of calibrated depths. However, the system geometry creates bounds on possible calibration locations. Point sources arbitrarily close to the sensor would produce caustic patterns that exceed the sensor size. To avoid this complication, we impose a minimum object distance at which an on-axis point source creates caustics that fill the sensor. Point sources arbitrarily far from the sensor theoretically can be captured, but axial resolution degrades with depth. The hyperfocal plane represents the axial distance beyond which no depth discrimination is available, establishing an upper bound. Objects beyond the hyperfocal focal plane can still be reconstructed to create 2D images for photographic applications [72], without any hardware modifications.

In our prototype, the axial FoV ranges from the minimum calibration distance (7.3 mm) to the hyperfocal plane (2.3 m). The angular FoV is limited by the pixel angular acceptance ($\alpha_c = 41.5^\circ$ in x, $\alpha_c = 30^\circ$ in y). Combined with our diffuser's maximum deflection angle ($\beta = 0.5^\circ$) this yields an angular FoV of $\pm 42^\circ$ in x and $\pm 30.5^\circ$ in y. We validate the lateral FoV experimentally by capturing a scene at optical infinity and measuring the angular extent of the result (see Supplementary Fig. S3).

Resolution

Investigating optical resolution is critical for both quantifying system performance and choosing our reconstruction grid. Although the raw data is collected on a fixed sensor grid, we can choose the non-uniform 3D reconstruction grid arbitrarily. This choice of reconstruction grid is important. When the grid is chosen with voxels that are too large, resolution is lost, and when they are too small, extra computation is performed without resolution gain. In this section we explain how to choose the grid of voxels for our reconstructions, with the aim of Nyquist sampling the two-point optical resolution limit.

Two-point resolution

A common metric for resolution analysis in traditional cameras is two-point distinguishablity. We measure our system's two-point resolution by imaging scenes containing two point sources at different separation distances, built by summing together images of a single point source (1μm pinhole, wavelength $532nm$) at two different locations. We reconstruct the scene using our algorithm, with $\tau = 0$ to remove the influence of the regularizer. To ensure best-case resolution, we use the full 5 MP sensor data (no binning). The point sources are considered distinguishable if the reconstruction has a dip of at least 20% between the sources, as in

the Rayleigh criterion. Figure 4.3c shows reconstructions with point sources separated both laterally and axially.

Our system has highly non-isotropic resolution (Fig. 4.3d), but we can use our model to predict the two-point distinguishability over the entire volume from localized experiments. Due to the shift invariance assumption, the lateral resolution is constant within a single depth plane and the paraxial magnification causes the lateral resolution to vary linearly with depth. For axial resolution, the main difference between two point sources is the size of their PSF supports. We find pairs of depths such that the difference in their support widths is constant:

$$c = \frac{1}{z_1} - \frac{1}{z_2}. \tag{4.7}$$

Here, z_1 and z_2 are neighboring depths and c is a constant determined experimentally.

Based on this model, we set the voxel spacing in our grid to Nyquist sample the 3D two-point resolution. Figure 4.3d shows a to-scale map of the resulting voxel grid. Axial resolution degrades with distance until it reaches the hyperfocal plane ($\sim$2.3 m from the camera), beyond which no depth information is recoverable. Due to the non-telecentric nature of the system, the voxel sizes are a function of depth, with the densest sampling occurring close to the camera. Objects within 5 cm of the camera can be reconstructed with somewhat isotropic resolution; this is where we place objects in practice.

Multi-point resolution

In a traditional camera, resolution is a function of the system and is independent of the scene. In contrast, computational cameras that use nonlinear reconstruction algorithms may incur degradation of the effective resolution as the scene complexity increases. To demonstrate this in our system, we consider a more complex scene consisting of 16 point sources. Figure 4.4 shows experiments using 16 point sources arranged in a 4×4 grid in the (x, z) plane at two different spacings. The first spacing is set to match the measured two-point resolution limit (Δx=45μm, Δz=336μm). Despite being able to separate two points at this spacing, we cannot resolve all 16 sources. However, if we increase the source separation to (Δx=75μm, Δz=448μm), all 16 points are distinguishable (Fig. 4.4d). In this example, the usable lateral resolution of the system degrades by approximately 1.7× due to the increased scene complexity. As we show in Section 4.34.3, the resolution loss does not become arbitrarily worse as the scene complexity increases.

This experiment demonstrates that existing resolution metrics cannot be blindly used to determine performance of computational cameras like ours. How can we then analyze resolution if it depends on object properties? In the next section, we introduce a general theoretical framework for assessing resolution in computational cameras like ours.

Local condition number theory

Our goal is to provide new theory that describes how the effective reconstruction resolution of computational cameras changes with object complexity. To do so, we introduce a numerical analysis of how well our forward model can be inverted.

First, note that recovering the image $\mathbf{v}$ from the measurement $\mathbf{b} = \mathbf{H}\mathbf{v}$ entails simultaneous estimation of the locations of all nonzeros within our image reconstruction, $\mathbf{v}$, as well as the values at each nonzero location. To simplify the problem, suppose an oracle tells us the exact locations of every source within the 3D scene. This corresponds to knowing *a priori* the support of $\mathbf{v}$, so we then need only determine the *values* of the nonzero elements in $\mathbf{v}$. This can be done by solving a least squares problem using a sub-matrix consisting of only the columns of $\mathbf{H}$ that correspond to the indices of the nonzero voxels. If this problem fails, then the more difficult problem of simultaneously determining the nonzero locations *and* their values will certainly fail.

In practice, the measurement is corrupted by noise. The maximal effect this noise can have on the least-squares estimate of the nonzero values is determined by the condition number of the sub-matrix described above. We therefore say that the reconstruction problem is ill-posed if any sub-matrices of $\mathbf{H}$ are very ill-conditioned. In practice, ill-conditioned matrices result in increased noise sensitivity and longer reconstruction times, as more iterations are needed to converge to a solution.

In general, finding the worst-case sub-matrix is a hard problem. However, because our system measurements vary smoothly for inputs within a small neighborhood, the worst-case scenario is when multiple sources are in a contiguous block (*i.e.* nearby measurements are most similar, either by shift or scaling). Therefore, we compute the condition number of sub-matrices of $\mathbf{H}$ corresponding to a group of point sources with separation varying by integer numbers of voxels. We repeat this calculation for different numbers of sources. The results are shown in Fig. 4.5. As expected, the conditioning is worse when sources are closer together. In this case, increased noise sensitivity means that even small amounts of noise could prevent us from resolving the sources. This trend matches what we saw experimentally in Figs. 4.3 and 4.4.

Figure 4.5 also shows that the local condition number increases with the number of sources in the scene, as expected. This means that resolution will degrade as more and more sources are added. We see in Fig. 4.5, however, that as the number of sources is increased, the conditioning approaches a limiting case. Hence, the resolution does not become arbitrarily worse with increased number of sources. Therefore we can estimate the system resolution for complex objects from distiguishability measurements with a limited number of point sources. This is experimentally validated in Sec. 4.4, where we find that the experimental 16-point resolution is a good predictor of the resolution for a USAF target.

Unlike the traditional two-point resolution metric, our new local condition number theory explains the resolution loss we observe experimentally. Since many optical systems are locally shift invariant, we believe that it is sufficiently general to be applicable to other computational cameras that use nonlinear algorithms, which likely exhibit similar performance loss.

Validity of the Convolution Model

In Sec. 4.24.2, we modeled the caustic pattern as shift invariant at every depth, leading to simple calibration and efficient computation. Since our convolution model is an approxima-

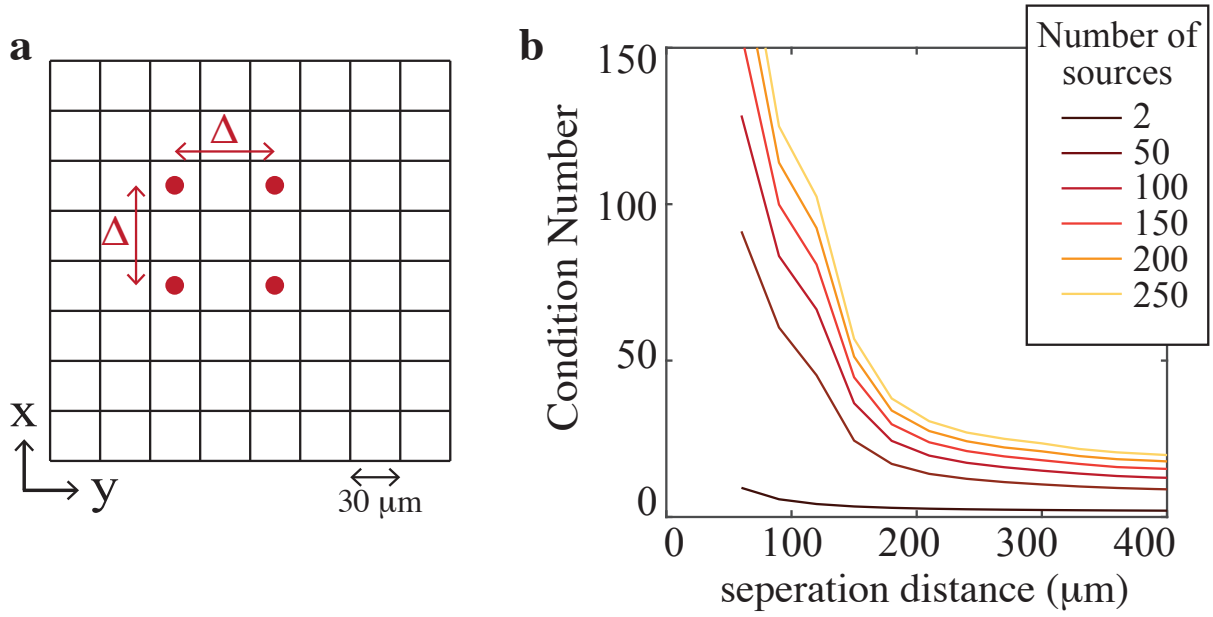


Figure 4.5: Our local condition number theory shows how resolution varies with object complexity. (a) Virtual point sources are simulated on a fixed grid and moved by integer numbers of voxels to change the separation distance. (b) Local condition numbers are plotted for sub-matrices corresponding to grids of neighboring point sources with varying separation (at depth 20 mm from the sensor). As the number of sources increases, the condition number approaches a limit, indicating that resolution for complex objects can be approximated by a limited number (but more than two) sources.

tion, we should quantify its validity. Figure 4.6a-c shows registered close-ups of experimentally measured PSFs from plane waves incident at 0°, 15° and 30°. The convolution model assumes that these are all exactly the same, though, in reality, they have subtle differences. To quantify the similarity across the FoV, we plot the inner product between each off-axis PSF and the on-axis PSF (see Fig. 4.6d). The inner product is greater than 75% across the entire FoV and particularly good within ±15° of the optical axis, indicating that the convolution model holds relatively well.

To investigate how the spatial variance of the PSF impacts system performance, we use the peak width of the cross-correlation between the on-axis and off-axis PSFs to approximate the spot size off-axis. Figure 4.6e (solid) shows that we retain the on-axis resolution up to ±15°. Beyond that, the resolution gradually degrades. To avoid model mismatch, one could replace the convolution model with exhaustive calibration over all positions in the FoV. This procedure would yield higher resolution at the edges of the FoV, as shown by the dashed line in Fig. 4.6e. The gap between these lines is what we sacrifice in resolution by using the convolution model. However, in return, we gain simplified calibration and efficient computation, which makes the large-scale problem feasible.

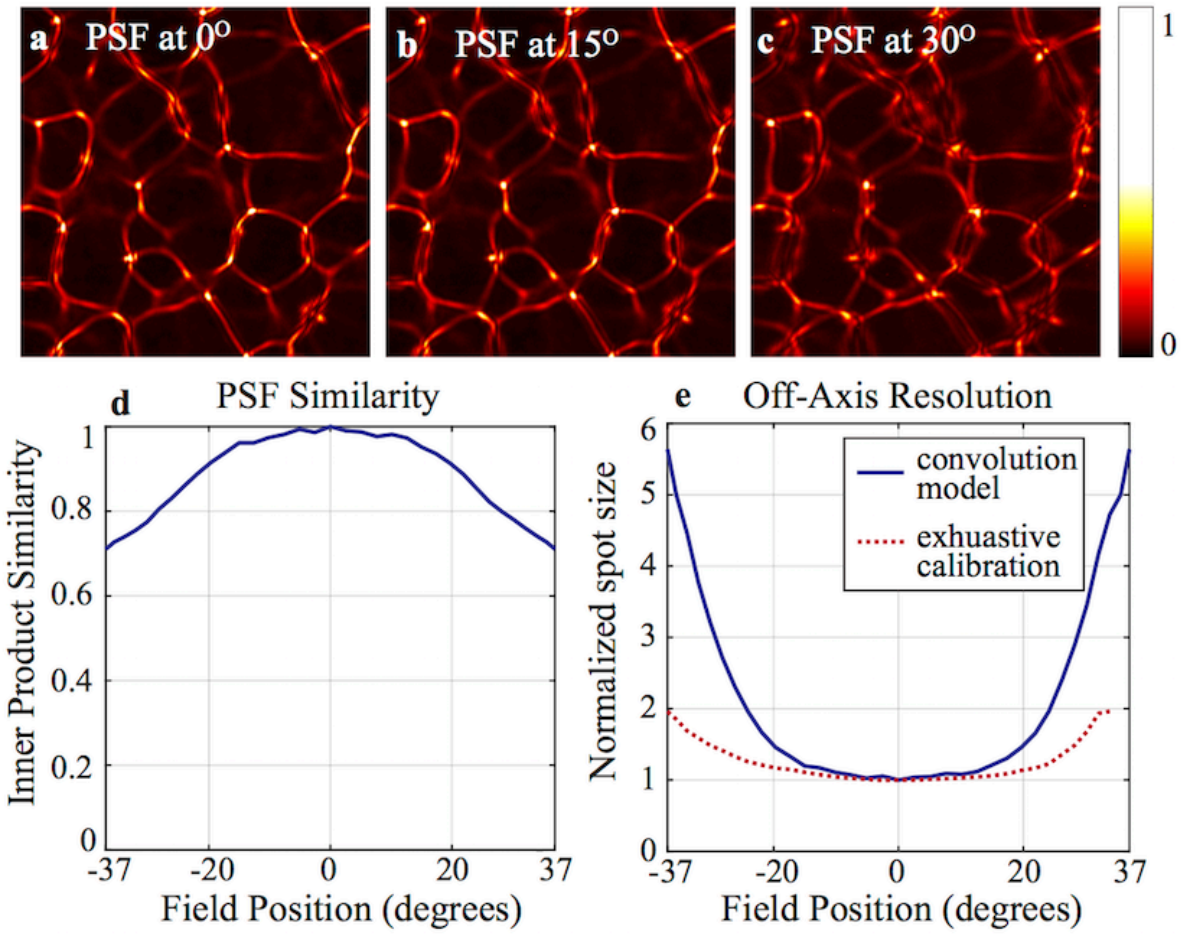


Figure 4.6: Experimental validation of the convolution model. (a)-(c) Close-ups of registered experimental PSFs for sources at 0°, 15° and 30°. The PSF at 15° is visually similar to that on-axis, while the PSF at 30° has subtle differences. (d) Inner product between the on-axis PSF and registered off-axis PSFs as a function of source position. (e) Resulting spot size (normalized by on-axis spot). The convolution model holds well up to ±15°, beyond which resolution degrades (solid). Exhaustive calibration would improve the resolution (dashed), at the expense of complexity in computation and calibration.

4.4 Experimental Results

Images of two objects are presented in Fig. 4.7. Both were illuminated using broadband white light and reconstructed with a 3D TV regularizer. We choose a reconstruction grid that approximately Nyquist samples the two-point resolution (by 2×2 binning the sensor pixels to yield a 1.3 megapixel measurement). Calibration images are taken at 128 different z-planes, ranging from z=10.86mm to z=36.26mm (from the diffuser), with spacing set according to conditions outlined in Sec. 4.34.3. The 3D images are reconstructed on a 2048×2048×128 grid, but the angular FoV restricts the usable portion of this grid to the center 100 million voxels. Note that the resolvable feature size on this reconstruction grid can still vary based on object complexity.

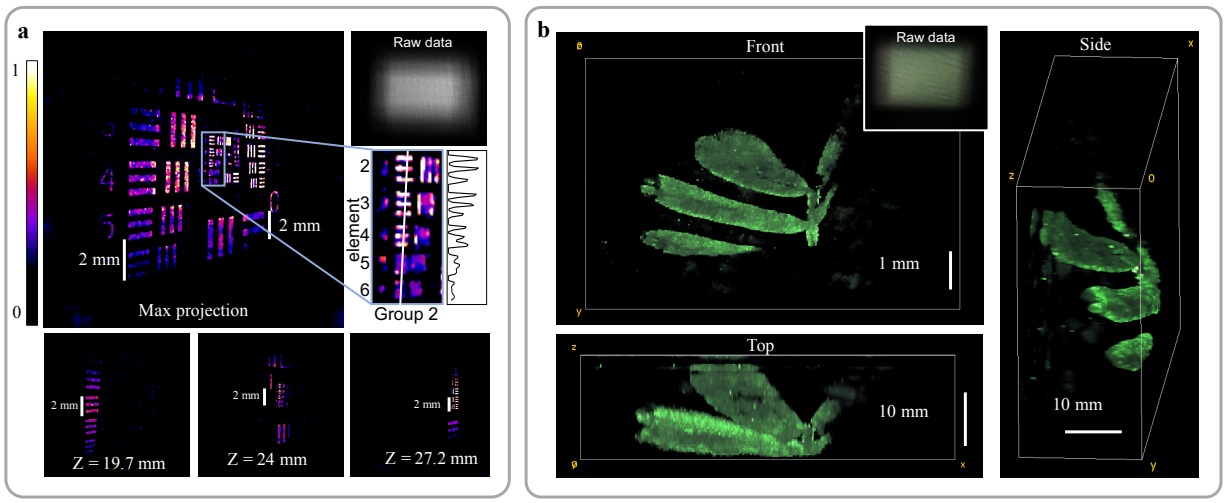


Figure 4.7: Experimental 3D reconstructions. (a) Tilted resolution target, which was reconstructed on a 4.2 MP lateral grid with 128 z-planes and cropped to 640×640×50 voxels. The large panel shows the max projection over z. Note that the spatial scale is not isotropic. Inset is a magnification of group 2 with an intensity cutline, showing that we resolve element 5 at a distance of 24 mm, which corresponds to a feature size of 79 μm (approximately twice the lateral voxel size of 35μm at this depth). The degraded resolution matches our 16-point distinguishability (75 μm at 20 mm depth). Lower panels show depth slices from the recovered volume. (b) Reconstruction of a small plant, cropped to 480×320×128 voxels, rendered from multiple angles.

The first object is a negative USAF 1951 fluorescence test target, tilted 45° about the y-axis (Fig. 4.7a). Slices of the reconstructed volume at different z planes are shown in order to highlight the system's depth sectioning capabilities. As described in Sec. 4.34.3, the spatial scale changes with depth. Analyzing the resolution in the vertical direction (Fig. 4.7a inset), we can easily resolve group 2 element 4 and barely resolve group 2 element 5 at z=24mm. This corresponds to resolving features 79μm apart on the resolution target. This resolution is significantly worse than the two-point resolution at this depth (50μm), but similar to the 16-point resolution (75μm). Hence, we reinforce our claim that two-point resolution is a misleading metric for computational cameras, but multi-point distinguishability can be extended to more complex objects.

Finally, we demonstrate the ability of DiffuserCam to image natural objects by reconstructing a small plant (Fig. 4.7b). Multiple perspectives of the 3D reconstruction are rendered to demonstrate the ability to capture the 3D structure of the leaves.

4.5 Conclusion

We demonstrated a simple optical system, with only a diffuser in front of a sensor, that is capable of single-shot 3D imaging. The diffuser encodes the 3D location of point sources in caustic patterns, which allow us to apply compressed sensing to reconstruct more voxels than we have measurements. By using a convolution model that assumes that the caustic pattern is shift invariant at every depth, we developed an efficient ADMM algorithm for

image recovery and simple calibration scheme. We characterized the FoV and two-point resolution of our system, and showed how resolution varies with object complexity. This motivated the introduction of a new condition number analysis, which we used to analyze how inverse problem conditioning changes with object complexity.

4.6 Supplemental comments

4.7 System Properties

Diffuser Properties

To quantify the properties of our diffuser, we used an LED array microscope to capture a quantitative Differential Phase Contrast (DPC) [130] image of the diffuser phase. After using the index of refraction of the diffuser material (polycarbonate, $n = 1.58$) to convert phase into surface shape, we show in Fig. 4.8 the measured relative height profile of a small patch on our 0.5° diffuser. The surface slope of the diffuser is Gaussian distributed with average magnitude of 0.7°. The deflection angle at the diffuser surface has a HWHM angle of 0.25°, which matches the manufacturer specifications. The maximum deflection angle is $\beta = 0.5°$, as shown in the histograms in Fig. 4.8.

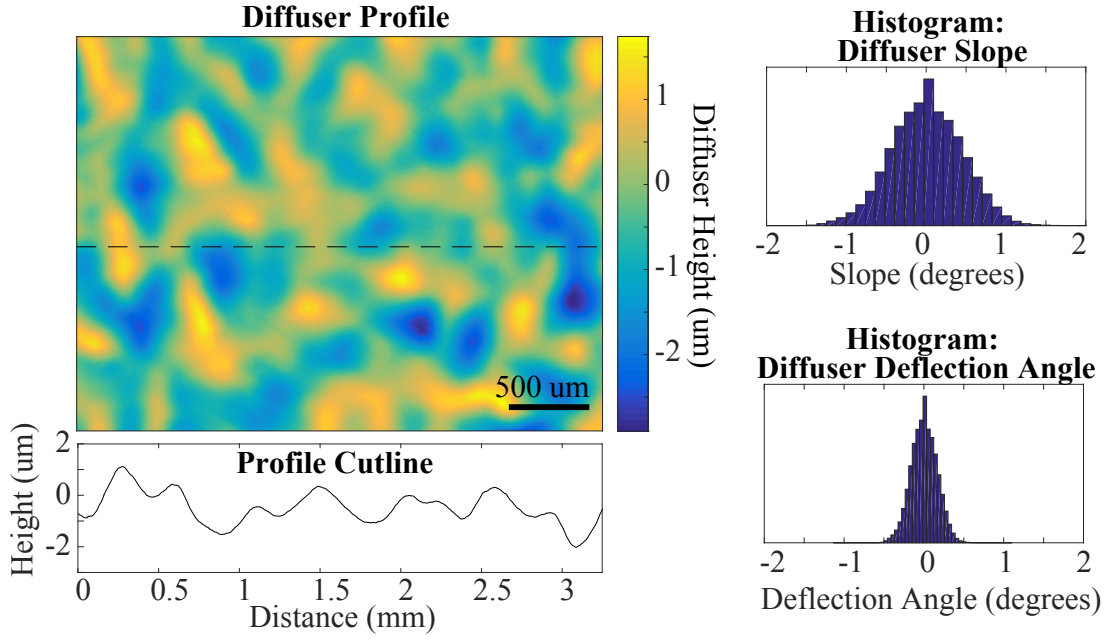


Figure 4.8: Left: The thickness profile of a small patch of our diffuser, as measured by quantitative Differential Phase Contrast (DPC) microscopy. Below is a cut-line plot along the dashed line. Right: Histograms of the diffuser slope (top) and the deflection angle of a ray normally incident on the diffuser (bottom). The maximum deflection angle is about 0.5°.

To illustrate the overall size and spread of the caustic PSF patterns in our system, we show in Fig. 4.9 the full PSF patterns captured for the closest and farthest axial distances

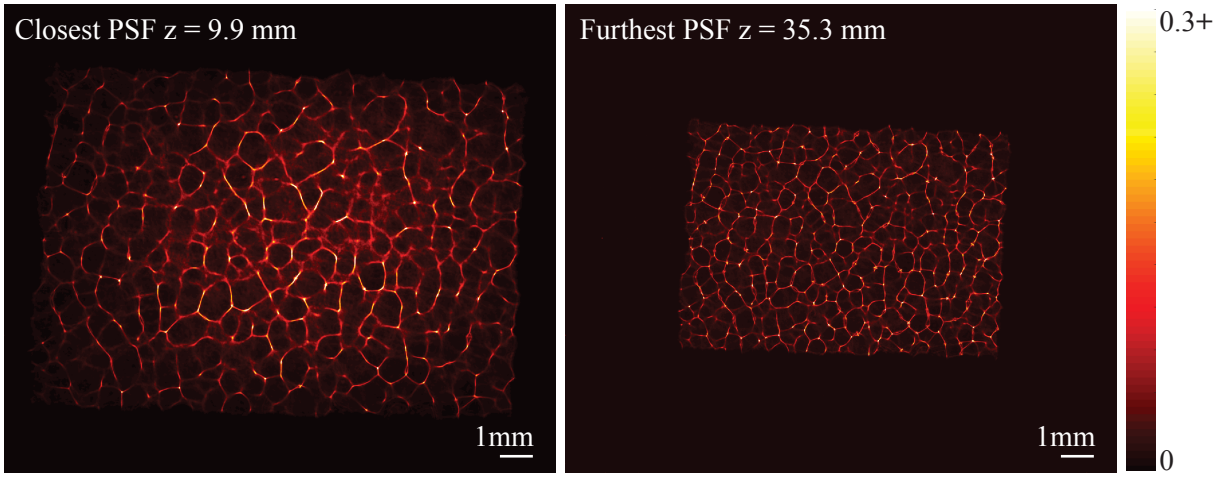


Figure 4.9: Un-cropped, false color sensor measurements of PSFs for the closest and farthest planes used in our reconstructions. These were measured by placing a point source on-axis at the front and back of the volume. The closest PSF has a caustic pattern that fills the sensor. Both PSFs have been contrast stretched from 0 to 30% of the max value for visibility.

used. Note that the closest axial distance is the one at which the caustic pattern just fills the sensor, and therefore depends on the aperture size. The caustics contain high-frequency information in all orientation directions, as evidenced by the sharp lines randomly spread in all directions. This facilitates good resolution at all depths and a highly structured PSF for deconvolution. Our calibration point source is a $30\mu m$ pinhole illuminated by a planar RGB LED array (λ = 630 nm, 515 nm, and 460 nm, $\Delta\lambda$ = 20 nm, 35 nm, and 25 nm, respectively) placed behind a 80° diffuser. As shown in [8], the caustics from narrowband and broadband sources are indistinguishable, and we do not find problems with using narrowband calibration.

Field-of-View Validation

In the main text in Sec. 3A, we derive the field-of-view (FoV) of our system to be

$$\text{FoV} = \beta + \min[\alpha_c, \tan^{-1}(\tfrac{l+w}{2d})], \tag{4.8}$$

where the FoV can be limited by either the geometry of the system (l, w, d) or by the angular acceptance of the pixels (α_c). Here l is the sensor size, w is the aperture size, and d is the distance between the diffuser and the sensor. In our system, $d = 8.9$ mm. In the x-direction, $l_x = 16.6$ and $w_x = 7.5$ mm; the y-direction values are $l_y = 14$ mm and $w_y = 5.5$ mm.

The angular response of the sensor, shown in Figure 3 of the main text, was measured by placing a white LED at optical infinity and rotating the sensor both vertically and horizontally. The average intensity measured at each angle was normalized by the on-axis measurement. We define the angular cutoff, α_c, as the angle at which the response falls to 20% of its on-axis value. For our camera, the x and y cutoffs are $\alpha_{cx} = 41.5°$ and $\alpha_{cy} = 30°$,

respectively. Finally, from our diffuser measurements in Fig. 4.8, we find that the maximum deflection angle of the diffuser, β, is 0.5°.

Plugging these values into the FoV equation yields a FoV of 42° in x and 30.5° in y, where the limiting factor is the angular acceptance. Figure 4.10 shows the recovery of a large, evenly illuminated scene at optical infinity. The angular extent visible in the reconstruction matches our predicted FoV.

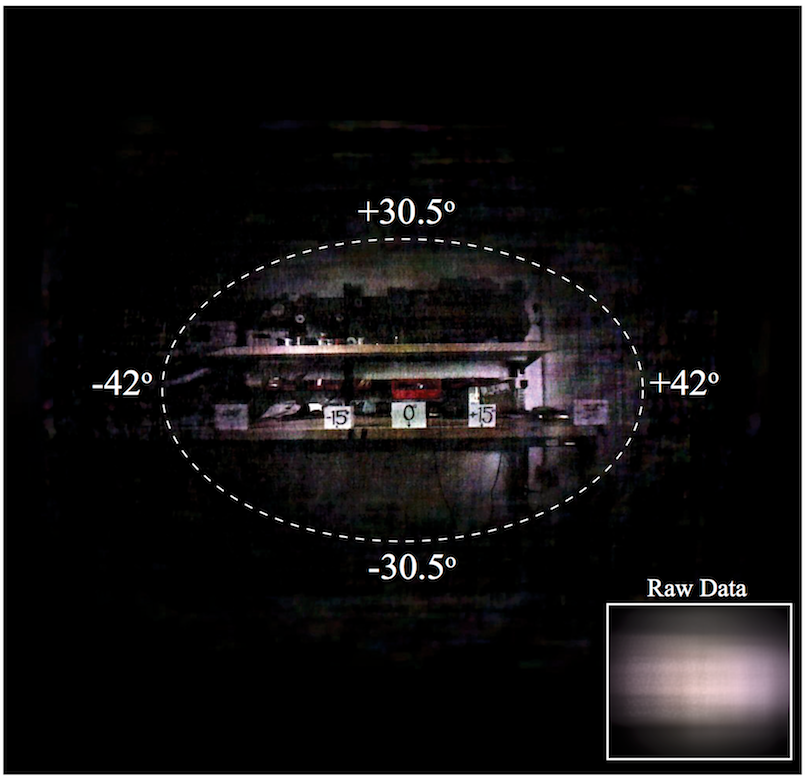


Figure 4.10: Validation of FoV calculations: based on the measured angular pixel response, α_c, and maximum diffuser deflection angle, β, we calculate our theoretical FoV to be 42° in x and 30.5° in y. This matches our recovered FoV in a scene at optical infinity. The inset shows the raw data.

PSF similarity

We quantify the similarity of the PSF versus shift and scale across the volume to validate our claim that the resulting underdetermined matrix has good properties for sparse recovery techniques. Figure 4.11 shows the autocorrelation of the PSFs acquired at the minimum and maximum object distances, as well as the cross-correlation between the two. Notice that the PSF autocorrelation maintains a sharp central peak and relatively low sidelobes for all depths within our calibration volume. This means that a shifted version of the PSF is roughly 50% similar to the un-shifted version. Importantly, the cross-correlation has no values greater than 50%, meaning that the scaled caustics are dissimilar to any shift of the unscaled caustics. To quantify this further, we plot the inner product between the central

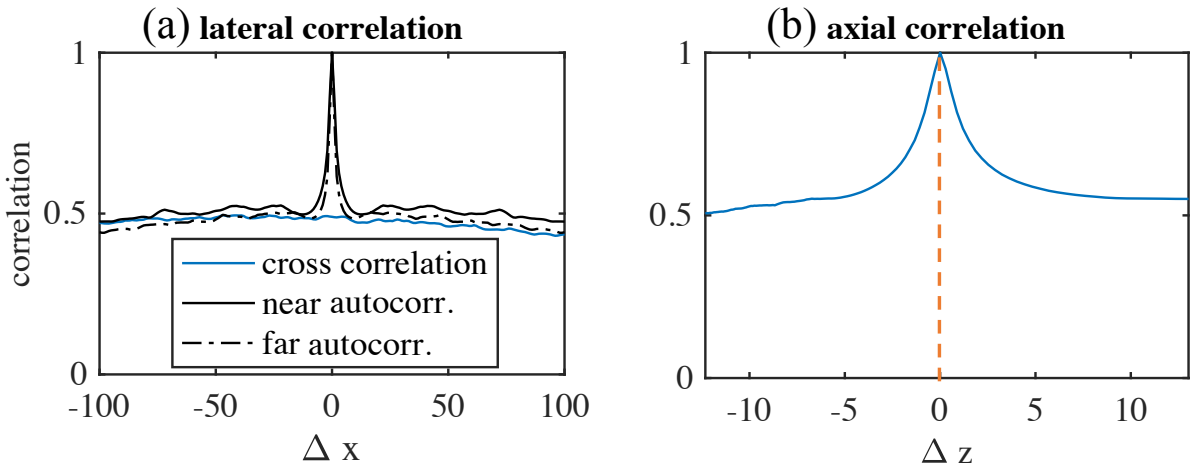


Figure 4.11: Correlation of various caustics patterns. (a) The caustics at a given depth are unique over shifting, and caustics from two different depths are not similar to each other, even under translation. The solid black curve is a slice of the autocorrelation of a PSF for a point source near the front of the volume, and the dotted black line is the autocorrelation for a far away point source's PSF. The solid blue line is the cross-correlation between the two. (b) The inner product of the PSF from the middle of the volume (corresponding to the orange dotted line) with all other PSFs at varying depths. In both (a) and (b), shifting or scaling the caustics leads to an inner product of approximately 0.5 compared to a peak value of 1.

image in the calibration stack, corresponding to the orange dotted line in Fig. 4.11b, with all other images in the stack. We again observe a relatively sharp peak and side lobes on the order of 50% in the axial direction. This validates our claim that the caustics produced by any point in the volume are unique.

4.8 Algorithm Details

Cropping in Forward Model

In Eq. (4) of the main text, we show that our forward model is a sum of convolutions followed by a crop operation. We would like to emphasize that the crop operation is due directly to the physical cropping caused by the finite sensor size. Consider an off-axis point source, as shown in Fig. 4.12a. In the experimental measurement from the source (Fig. 4.12b), half of the on-axis PSF is cut off by the finite size of the sensor. If we do not take this into account in our forward model, our estimate of the measurement would look like Fig. 4.12c, which is not physical due to the circular boundary conditions. Including the crop operation in our forward model fixes the problem, creating estimates of the measurement that look like the experimental data (Fig. 4.12d).

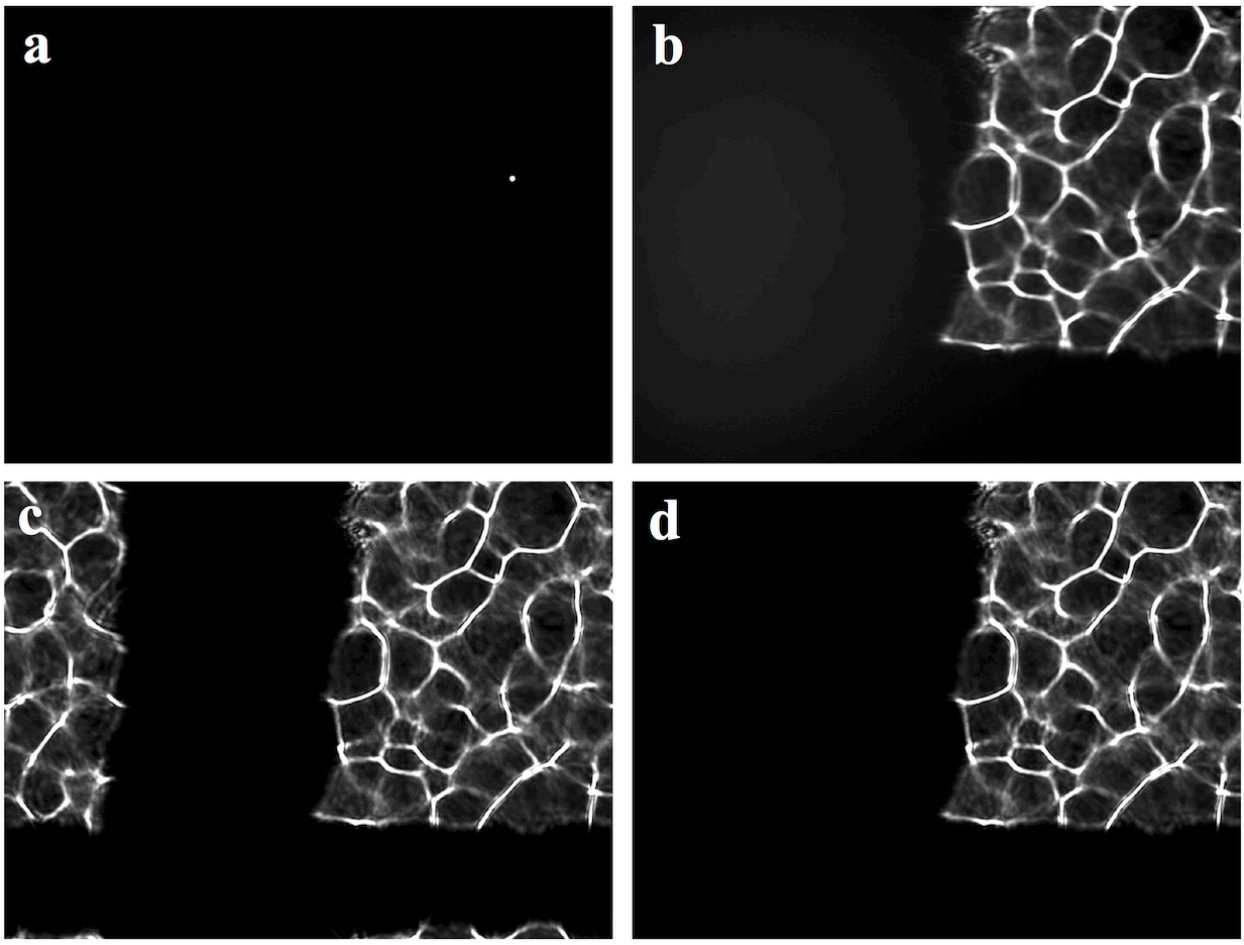


Figure 4.12: The crop operation in the forward model accounts for the finite sensor size. (a) Off-axis point source, size exaggerated for visibility. (b) Experimental measurement from the source. (c) Simulated measurement without crop operation. Since the convolution has circular boundary conditions, the PSF wraps around to the opposite side of the sensor. (d) Simulated measurement with crop operation matches the experimental measurement.

Derivation of ADMM Inverse Algorithm Formulation

As stated in the main paper in Section 1B, the problem we seek to solve is:

$$\hat{\mathbf{v}} = \underset{\mathbf{v}\geq 0}{\operatorname{argmin}} \tfrac{1}{2}\|\mathbf{b} - \mathbf{H}\mathbf{v}\|_2^2 + \tau\|\Psi\mathbf{v}\|_1. \tag{4.9}$$

We transform this into the equivalent problem:

$$\begin{aligned} \hat{\mathbf{v}} = \underset{w,u,v}{\operatorname{argmin}} & \tfrac{1}{2}\|\mathbf{b} - \mathbf{D}v\|_2^2 + \tau\|u\|_1 + \mathbb{1}_+(w) \\ \text{s.t. } & v = \mathbf{M}\mathbf{v} \\ & u = \Psi\mathbf{v} \\ & w = \mathbf{v}, \end{aligned} \tag{4.10}$$

where $\mathbb{1}_+(\cdot)$ is the nonnegativity barrier function, which returns 0 when the argument is nonnegative, and ∞ when the argument is negative.

In order to compute the ADMM updates efficiently, we will see that it is useful for both $\mathbf{M}$ and Ψ to represent 3D convolutions. Clearly, when Ψ is the identity matrix, this holds. Additionally, when Ψ is the 3D finite difference operator, it can be expressed as a concatenation of 3D convolutions with the finite difference kernel, oriented in each of the 3 directions. In order to express $\mathbf{M}$ as a 3D convolution, we must choose the diagonal operator, $\mathbf{D}$, such that Eq. (4) can be written as $\mathbf{D}\left(m \overset{(x,y,z)}{*} \mathbf{v}\right)$, where m is a 3D kernel, and $\overset{(x,y,z)}{*}$ represents convolution over the variables, x, y, and z. To accomplish this, we use the fact that a sum of 2D convolutions between an object, $\mathbf{v}(x,y,z)$, and a stack of 2D kernels, $h(x,y;z)$, can be expressed as the first 2D (x,y)-slice in the 3D convolution between the object and a z-flipped version of the kernel stack:

$$\sum_z h(x,y;z) \overset{(x,y)}{*} \mathbf{v}(x,y,z) = \left[h(x,y;-z) \overset{(x,y,z)}{*} \mathbf{v}(x,y,z)\right]\Big|_{z=0}. \tag{4.11}$$

For proof, we can take the right hand side of (4.11) and apply the definition of discrete 3D convolution directly:

$$\begin{aligned}
&\left[h(x,y;-z) \overset{(x,y,z)}{*} \mathbf{v}(x,y,z)\right]\Big|_{z=0} \\
&= \sum_{z'=0}^{N_z-1} \sum_{y'=0}^{N_y-1} \sum_{x'=0}^{N_x-1} \mathbf{v}(x',y',z') h(x-x', y-y'; z'-z)|_{z=0} \\
&= \sum_{z'=0}^{N_z-1} \mathbf{v}(x,y,z') \overset{(x,y)}{*} h(x,y;z').
\end{aligned}$$

Using this identity, we can write the forward operator in Eq. (4) as:

$$\begin{aligned}
&\mathbf{C} \sum_z \left[\mathbf{v}\left(\frac{-x'}{m}, \frac{-y'}{m}, z\right) \overset{(x,y)}{*} h\left(x', y'; z\right)\right] \\
=&\mathbf{C}\left[\mathbf{v}\left(\frac{-x'}{m}, \frac{-y'}{m}, z\right) \overset{(x',y',z)}{*} h\left(x', y'; -z\right)\Big|_{z=0}\right] \\
=&\mathbf{D}\left[\mathbf{v}\left(\frac{-x'}{m}, \frac{-y'}{m}; z\right) \overset{(x',y',z)}{*} h\left(x', y'; -z\right)\right],
\end{aligned}$$

where $\mathbf{D}$ is a diagonal operator that simultaneously performs the 2D crop, $\mathbf{C}$, as well as selecting the $z = 0$ slice. Effectively, $\mathbf{D}$ comprises taking the center crop of the first layer of the 3D array resulting from the circular 3D convolution of $h(x', y'; -z)$ with $\mathbf{v}$. Note that our definition of z is as a parameter indexing each slice in the 3D array h, not the

physical distance to each slice. We assume circular boundary conditions for h, such that $h(\cdot,\cdot;-z) = h(\cdot,\cdot;N_z - z)$ is a z-stack that is flipped in the z-direction.

Using (4.11), we present an efficient method for solving (4.10). We begin by transforming (4.10) into an unconstrained augmented Lagrangian form, and consider the saddle-point problem:

$$\begin{aligned}\max_{\xi,\eta,\rho}\Bigg[\min_{u,v,w,\mathbf{v}} &\tfrac{1}{2}\left\|\mathbf{b}-\mathbf{D}v\right\|_2^2 + \tfrac{\mu_1}{2}\left\|\mathbf{M}\mathbf{v} - v + \tfrac{\xi}{\mu_1}\right\|_2^2 \\ &+ \tau\|u\|_1 + \tfrac{\mu_2}{2}\left\|\Psi\mathbf{v} - u + \tfrac{\eta}{\mu_2}\right\|_2^2 \\ &+ \mathbb{1}_+(w) + \tfrac{\mu_3}{2}\left\|\mathbf{v} - w + \tfrac{\rho}{\mu_3}\right\|_2^2\Bigg].\end{aligned}$$

To solve the above equation using ADMM, we first derive the optimality conditions for each primal variable, assuming the others are fixed:

$$\begin{aligned}
u^{k+1} &\leftarrow \operatorname*{argmin}_u && \tau\|u\|_1 + \tfrac{\mu_2}{2}\left\|\Psi\mathbf{v}^k - u + \tfrac{\eta^k}{\mu_2}\right\|_2^2 \\
v^{k+1} &\leftarrow \operatorname*{argmin}_v && \tfrac{1}{2}\left\|\mathbf{b}^k - \mathbf{D}v\right\|_2^2 + \tfrac{\mu_1}{2}\left\|\mathbf{M}\mathbf{v}^k - v + \tfrac{\xi^k}{\mu_1}\right\|_2^2 \\
w^{k+1} &\leftarrow \operatorname*{argmin}_w && \mathbb{1}_+(w) + \tfrac{\mu_3}{2}\left\|\mathbf{v}^k - w + \tfrac{\rho^k}{\mu_3}\right\|_2^2 \\
\mathbf{v}^{k+1} &\leftarrow \operatorname*{argmin}_{\mathbf{v}} && \tfrac{\mu_1}{2}\left\|\mathbf{M}\mathbf{v} - v^{k+1} + \tfrac{\xi^k}{\mu_1}\right\|_2^2 \\
& && + \tfrac{\mu_2}{2}\left\|\Psi\mathbf{v} - u^{k+1} + \tfrac{\eta^k}{\mu_2}\right\|_2^2 \\
& && + \tfrac{\mu_3}{2}\left\|\mathbf{v} - w^{k+1} + \tfrac{\rho^k}{\mu_3}\right\|_2^2.
\end{aligned}$$

And update each dual variable as

$$\begin{aligned}
\xi^{k+1} &\leftarrow \xi^k + \mu_1(\mathbf{M}\mathbf{v}^{k+1} - v^{k+1}) \\
\eta^{k+1} &\leftarrow \eta^k + \mu_2(\Psi\mathbf{v}^{k+1} - u^{k+1}) \\
\rho^{k+1} &\leftarrow \rho^k + \mu_3(\mathbf{v}^{k+1} - w^{k+1}).
\end{aligned}$$

The final result is the algorithm outlined in Sec. 2C of the main text.

Implementation details

Evaluation of the cropped discrete convolution at a single depth,

$$\mathbf{C}\left[h(x',y';z) \overset{(x',y')}{*} \mathbf{v}(-x'/m, -y'/m, z)\right](x',y'),$$

a FISTA, ℓ_1, 2048x2048x64 (4 hours) **b** ADMM, 3DTV, 2048x2048x128 (26 mins)

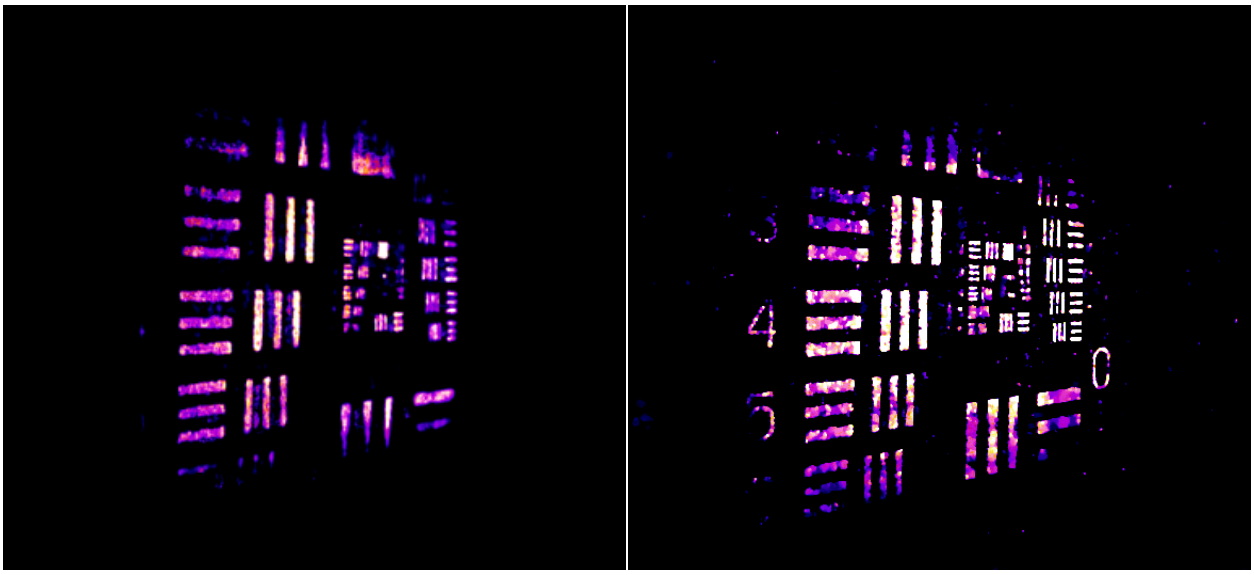


Figure 4.13: ℓ_1 vs 3DTV regularization with different algorithm implementations. (a) Max z-projection of FISTA reconstruction using ℓ_1 (soft thresholding on the volume after each iteration). This took 4 hours to run on a Titan X GPU using MATLAB. The soft thresholding has erased some key features. (b) Max z-projection of reconstruction using ADMM with a 3DTV prior. Clearly the result is better, largely due to a better sample-prior match. This reconstruction also required 10x less time to obtain.

is done by zero padding $h(x', y'; z)$ to twice its original size in each dimension, then using FFT-based convolution. This ensures that any aliasing artifacts introduced by the circular boundary conditions of the FFT will fall outside the sensor area, causing such artifacts to be removed by the cropping, $\mathbf{C}(\cdot)$. Note that this requires our variable, $\mathbf{v}$, to be approximately twice as many samples in each dimension as our sensor measurement. Interestingly, it is possible for useful information to lie anywhere within this extended FoV. In our prototype, the angular falloff of the sensor means that measurements in the extended region are attenuated too much to be useful. However, a future system using different geometry could leverage this effect to gain even more useful samples in the final reconstruction. In operator notation, the convolution can be evaluated as

$$\text{crop}\,(F)^{-1}\left\{\,[FPh(x,y;z)]\cdot[F\mathbf{v}(x,y,z)]\,\right\} \tag{4.12}$$

where F is the 2D FFT, $\cdot$ is point-wise multiplication, and P is the zero-padding operator.

ℓ_1 vs 3D Total Variation

To improve the quality of reconstruction, we use the 3D Total Variation (TV) penalty parameter. This is inefficient to compute as part of a projected gradient technique, because the proximal operator for the TV norm must be computed iteratively. On volumes of the size used here, this requires minutes per outer-loop iteration. Of the priors considered in

this work, only native sparsity and nonnegativity are feasible when using projected gradient methods. To demonstarate the benefit of using ADMM, we show in Fig. 4.13 the result from ℓ_1 regularized FISTA after running for 4 hours on a GPU using MATLAB compared to our algorithm with 3DTV regularization for 20 minutes. Not only does our algorithm run much faster, but it produces an image with more detail. In particular, note that the ℓ_1 regularization has erased the numbers and eroded the bars, whereas 3DTV runs an order or magnitude faster *and* uses a more sophisticated prior, resulting in categorically better performance.

Chapter 5

Designing diffusers

Phase Mask Design

In this section, we present theory for designing and optimizing a phase mask for the task of snapshot 3D imaging. The goal is a mask that achieves a target resolution uniformly across a specified 3D volume. This theory will hold for any mask that is shift-invariant over a reasonably sized patch of the object. The example used here assumes that the phase mask will be placed in the aperture stop of the objective with the sensor at a fixed distance, with the goal of miniaturized 3D integrated fluorescence microscopy; this architecture reduces the size and weight of the device, makes the system close to shift-invariant and enables multiplexing, which is necessary for compressed sensing. Note this model is similar to what could be used in a lensless camera design, so this approach is not restricted to the miniaturized integrated architecture. We aim for all PSFs produced by the mask to have high spatial-frequency content and be mutually incoherent (i.e. all as dissimilar as possible). Toward this goal, we propose a multifocal array of nonuniformly-spaced microlenses as our phase mask.

The first step is to determine the free parameters that describe the phase mask. We choose to use a phase mask made of microlenses because it provides good light throughput, while balancing the trade-offs between SNR and compressive sensing capabilities. Specifically, we represent the microlens phase mask by parameterizing the i^{th} microlens by its lateral vertex location, $(\rho^i_{xc}, \rho^i_{yc}) := \boldsymbol{\rho}^i_{\boldsymbol{c}}$ and radius of curvature, R_i. The spherical sag of the microlens is:

$$s_i = d_i + R_i \sqrt{1 - \left(\frac{\boldsymbol{\rho} - \boldsymbol{\rho}^i_c}{R_i} \right)^2}, \tag{5.1}$$

where d_i is an offset constant added to each microlens to control its clear aperture. We parameterize aspheric terms in the microlenses by using Zernike polynomials. The j^{th} Zernike coefficient for microlens i is denoted α_{ij}, so the total aspheric component at that microlens is $\sum_j \alpha_{ij} Z_j(\boldsymbol{\rho} - \boldsymbol{\rho}^i_c)$ with Z_j being the j^{th} Zernike polynomial. As long as the microlenses are all convex ($R_i > 0$), a phase mask with full fill-factor can be constructed by taking the

point-wise maximum thickness (see Fig. 5.1). The phase mask surface is thus:

$$T(\rho_x, \rho_y; \boldsymbol{\theta}) = \max_i \left[s_i + \sum_j \alpha_{ij} Z_j(\boldsymbol{\rho} - \boldsymbol{\rho}_c^i) \right], \tag{5.2}$$

where $\boldsymbol{\theta}$ denotes the collection of parameters that define the phase mask: vertex locations $\{\boldsymbol{\rho}_c^i\}$, radii $\{R_i\}$, offsets $\{d_i\}$, and Zernike coefficients $\{\alpha_{ij}\}$. The resulting surface is guaranteed to be continuous and will have a well-defined local focal length given by $f_i = \frac{n-1}{R_i}$ within the region belonging to the i^{th} microlens, provided the power Zernike $j = 4$ is excluded. In practice, we optimize the Zernike coefficients for tilt ($j = 1, 2$) and astigmatism ($j = 3, 5$).

With the microlens array defined, the on-axis PSF at a given sample depth z can be modeled by Fresnel propagation of the pupil wavefront from a point source at depth z, denoted $W(\rho_x, \rho_y; z)$, multiplied by the phase of the designed mask, $\phi(\rho_x, \rho_y; \boldsymbol{\theta}) = \frac{2\pi(n-1)}{\lambda} T(\rho_x, \rho_y; \boldsymbol{\theta})$:

$$\mathbf{h}(u, v; z, \boldsymbol{\theta}) = \left| F_t \left\{ P(\rho_x, \rho_y) \exp\left[i\phi(\rho_x, \rho_y; \boldsymbol{\theta}) \right] W(\rho_x, \rho_y; z) \right\} \right|^2, \tag{5.3}$$

where $P(\rho_x, \rho_y)$ is the GRIN pupil amplitude, n is the microlens substrate index of refraction, and F_t denotes Fresnel propagation to the sensor a distance t away. Importantly, the on-axis PSFs are differentiable with respect to the microlens parameters, $\boldsymbol{\theta}$, enabling us to optimize the design using gradient methods, as discussed in the next section.

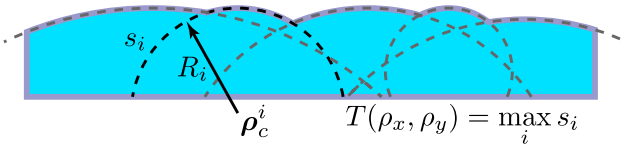


Figure 5.1: Phase mask parameterized by point-wise maximum of convex spheres. Each sphere is outlined by a dashed line, and the final optic is shaded blue (not to scale).

Our previous work employed off-the-shelf diffusers with a pseudorandom Gaussian surface profile [11]. These generate a caustic PSF that has poor SNR due to the spreading of the light by the concave bumps of the diffuser surface. In contrast, microlenses concentrate the light into a small number of sharp spots, giving better performance in low-light applications like fluorescence microscopy (see supplement 8.6). By parameterizing our design as a set of microlenses, we can also derive simple design rules from first-principles (sections *Lateral Resolution & Multifocal Design*), then use those to formulate an optimization problem that locally optimizes the placement and aberrations of each microlens.

We space our microlenses nonuniformly to ensure that the PSFs from all field points are dissimilar. Regularly-spaced arrays will produce highly similar PSFs when shifted by one microlens period, causing certain spatial frequencies to be poorly measured. Previous work avoided this ambiguity by introducing a field stop [88, 114, 51] that prevents the PSFs from overlapping, but this restricts the FoV significantly. Our design yields a larger FoV by using nonuniform spacing and computationally disambiguating the overlapping PSFs. In

Fig. 5.2 we compare PSFs and reconstructions from regularly-spaced and nonuniform phase mask designs. Looking at Fig. 5.2(c), the PSF of the regular array causes unwanted peaks at low frequencies in its radially-averaged *inverse power spectral density* (IPSD), a metric related to deconvolution performance [30] (lower is better). This manifests as artifacts in the simulated reconstruction, which are significantly reduced in reconstructions from both of the nonuniform designs.

Using multiple microlens focal lengths extends the depth range across which we obtain good resolution, as described in the section on *Multifocal Design*. Multifocal designs have sharp focal spots across a wider desired depth range than can be achieved with unifocal designs, trading SNR in-focus for better performance off-focus. Figure 5.2(c,d) compares the PSFs and reconstruction quality of our approach versus unifocal designs in-focus and 200 μm away from the native focus of the unifocal arrays. The blurry features in the out-of-focus PSFs for both unifocal designs cause poor performance, as shown in the reconstructions and high inverse power spectra. To capture the performance across depth, Fig.5.2(b) shows the integrated IPSD (up to the cutoff frequency) of each design versus depth. As expected, our multifocal design is slightly worse than a unifocal design in focus, but achieves far better (lower) values across the full depth range.

In the compact system architecture we propose, it is clear that our nonuniform multifocal microlenses are a good choice of phase mask. This motivates the next sections which provide guidance on optimizing the nonuniform spacing, as well as the focal lengths and aberrations of the microlenses for achieving a target resolution and depth range. For our prototype, we aim for 3.5 μm lateral resolution, and show that this can be achieved over a depth range up to 360 μm, which agrees with our experimental characterization.

Lateral Resolution

Lateral resolution will be primarily determined by the diffraction-limited aperture size of the microlenses, which also determines the number of microlenses that fit across the objective's full aperture, and thus, the depth range we can target. We design for lateral resolution that does not require the full pupil, so that we can fit multiple microlenses in the aperture for better depth coding. The example in Fig. 5.2 targets 3.5 μm resolution (cutoff frequency of 0.35 cycles/μm) using 36 microlenses with average NA=0.09. Because each design has the same number of microlenses, each has a similar resolution limit.

To quantify, we perform a diffraction analysis to find the clear aperture a single microlens needs to support a δx lateral resolution at the sample. Note that this assumes we will recover resolution no better than the band-limit of the measurement, neglecting any resolution gained from the non-linear solver. We start by calculating the magnification for our system:

$$M \approx \frac{-t}{f_G}, \tag{5.4}$$

where f_G is the GRIN focal length and t is the mask-to-sensor distance (derivation in 8.6). Note that M is approximately independent of the microlens focal length. For our system,

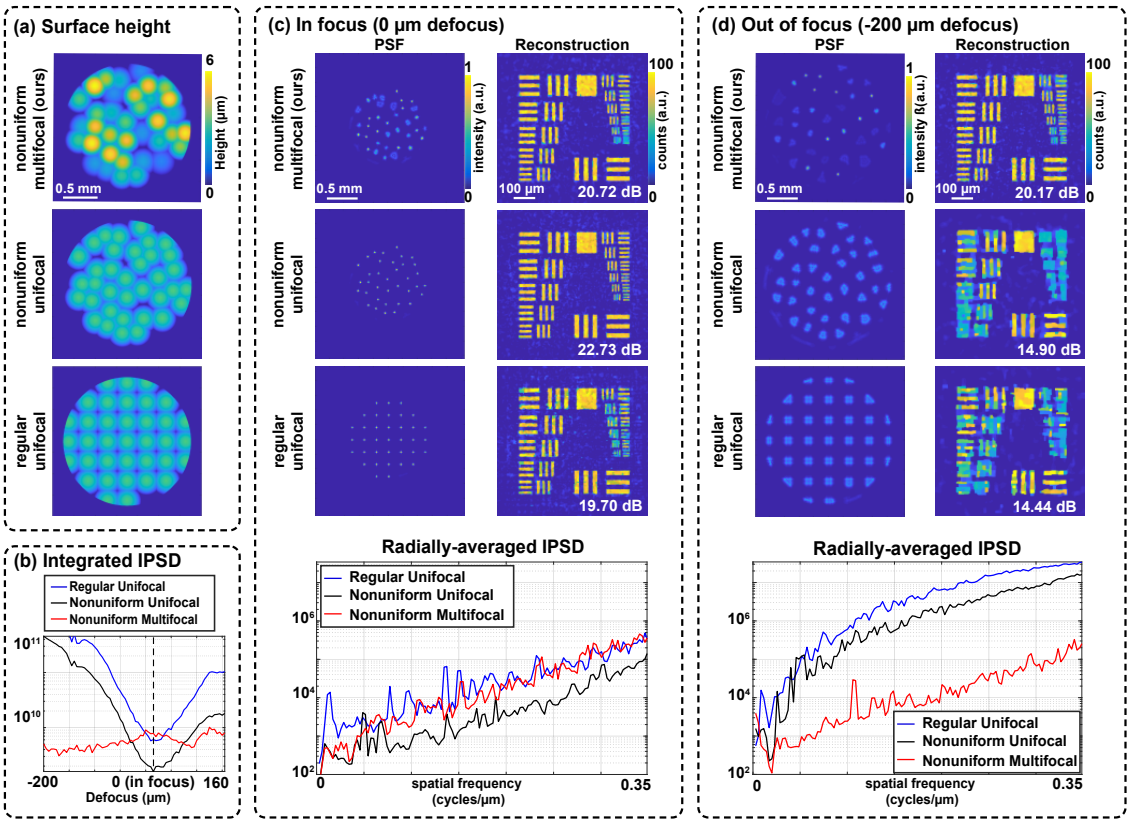


Figure 5.2: Simulations to motivate our phase mask design, comparing our proposed nonuniform multifocal design with regular unifocal and nonuniform unifocal designs. (a) Surface height profiles. (b) Sum of each design's PSF inverse power spectral density (IPSD) versus object depth (up to the designed cutoff frequency, lower is better). (c) PSFs and simulated reconstructions in-focus (at the unifocal arrays' native focus), with the reconstruction peak signal-to-noise ratio (PSNR) listed. The measurement is corrupted with 100 e^{-1} (peak) Poisson noise. In focus, the nonuniform unifocal design has slightly better PSNR and resolution than our design, and regular unifocal performs worse. The radially-averaged IPSD (lower is better) matches this trend. (d) Imaging 200μm off-focus, both unifocal designs produce blurry PSFs which result in significantly worse PSNR and resolution in the reconstruction, as compared to our design. This is also seen in the much higher inverse power spectra curves for unifocal designs.

$f_G = 1.67\ mm$ and $t = 8.7\ mm$, so $M \approx -5.2$. Using Eq. 8.9 and the Rayleigh criterion, the microlens clear aperture, Δ_M, needed for a target object resolution δx at wavelength λ is:

$$\Delta_{ML} = \frac{1.22\lambda t}{|M|\delta x} \approx \frac{1.22\lambda f_G}{\delta x}. \tag{5.5}$$

This expression is also independent of the microlens focal length because we have assumed the microlens is focused. Equation 8.10 allows us to select the appropriate average microlens spacing for a desired resolution. Our system is designed for 3.5 μm lateral resolution (though experimentally we achieve 2.76 μm, due to the non-linear solver), which gives an average microlens diameter of 300 μm. Given that the GRIN clear aperture has diameter 1.8 mm, this results in 36 microlenses that can fit in the phase mask. Note that since the GRIN

is aberration limited, the 2D Miniscope does not achieve the diffraction-limited resolution predicted by its full aperture size. Hence, our experimentally-measured resolution is not much worse than the 2D Miniscope (lateral resolution of 2 μm), despite dividing the GRIN pupil into 36 regions to add depth sensing capabilities.

Multifocal Design for Extended Depth Range

Focal length diversity in the microlens array results in an extended depth range, a key advantage of our architecture over conventional LFM. To maintain a uniform lateral resolution across all depths in the volume of interest, the PSF should have sharp, high-frequency focal spots for each axial position. This requires at least one microlens to be in focus for each object axial plane, with planes spaced by the microlens depth-of-field (DoF). The DoF of a single microlens, d_{ML}, is inversely proportional to the microlens clear aperture, Δ_{ML}, giving $d_{ML} = \pm 20\ \mu m$ in our system (see supplement 8.6).

Our design aims to have a minimum of 4 microlenses in focus within each DoF. Given that our lateral resolution criterion allows 36 microlenses, this means we should have 9 different focal lengths and a depth range of 360 μm, nearly 10$\times$ what a single focal length achieves. Note that there is a trade-off between the imaging depth range and lateral resolution. We can increase the depth range by including more microlenses in the mask; however, that decreases their clear aperture (Eq. 8.10) and thus the lateral resolution. Conversely, for imaging thin samples where only a narrow range of focal lengths is required, better lateral resolution is possible.

To determine the focal length distribution, we find the focal length needed to focus at the beginning of the depth range ($f_{min} = 7mm$) and at the end of the depth range ($f_{max} = 25$ mm). Then, we dioptrically space the focal lengths across the target range because this leads to microlenses that come into focus at linearly-spaced depth planes in the sample space.

Phase mask Optimization Using Matrix Coherence

The previous sections outlined first-order design principles, considering only a single microlens. In the next section, we will optimize the ensemble of microlenses (their positions and added aberrations) with metrics based on compressed sensing theory. Given the first-principles guidance in the above sections, we set the number of microlenses, their characteristic aperture size and their focal length distribution; next, we aim to optimize the microlens positions and aberrations to maximize performance. In order to make the optimization computationally feasible, we ignore the field-varying changes in the PSF and assume that the system is shift invariant for the purposes of design.

To optimize the microlens parameters, $\boldsymbol{\theta}$, in terms of the on-axis PSFs at each depth, we set up a loss function to be optimized that consists of two terms. The first term, a cross-coherence loss, promotes good axial resolution by ensuring that the PSFs at different depths are as dissimilar as possible. Cross-coherence between any two depths is defined as $\|\mathbf{h}(u,v;z_n) \star \mathbf{h}(u,v;z_m)\|_\infty := \max\left[\mathbf{h}(u,v;z_n) \star \mathbf{h}(u,v;z_m)\right]$, where $\star$ represents 2D correlation and $\max\cdot$ is the element-wise maximum. Intuitively, we want the cross-coherence to

be small, since it represents the worst-case ambiguity that would arise by placing two point sources adversarially at depths spaced according to the separation of their PSF's cross-correlation peaks. By computing this quantity for all pairs of z-depths, we can produce a differentiable figure-of-merit that optimizes the matrix coherence [26] between depths. In practice, rather than optimizing the cross-coherence, we smoothly approximate the max [31] using $\|x\|_\infty \approx \sigma \ln \sum \exp\left(x^2/\sigma\right)$. Here, $\sigma > 0$ is a tuning parameter that trades accuracy of the approximation against smoothness. For our purposes, this has the advantage of penalizing all large cross correlation values, not just the single largest. We will denote this $\|\cdot\|_{\overline{\infty}}$.

The total cross-coherence loss is then:

$$q(\boldsymbol{\theta}) = \sum_n \sum_{m>n} \|\mathbf{h}(u, v; \boldsymbol{\theta}, z_n) \star \mathbf{h}(u, v; \boldsymbol{\theta}, z_m)\|_{\overline{\infty}}. \tag{5.6}$$

The second term in the optimization ensures that lateral resolution is maintained. To do so, we optimize the autocorrelation of the PSF at each depth using the frequency domain least-squares method. The analysis in the *Lateral Resolution* section above only applies to a single microlens; building a phase mask of multiple lenses generally degrades resolution by introducing dips in the spectrum that reduce contrast at certain spatial frequencies. Hence, we treat the single-lens case as an upper limit that defines the bandlimit of the multi-lens PSF. To reduce spectral ripple, we penalize the ℓ_2 distance between the MTFs of the PSF and a diffraction-limited single microlens, $|H|$. We include a weighting term, denoted D, that ignores spatial frequencies beyond the bandlimit, as well as low spatial frequencies which are less critical and difficult to optimize due to out-of-focus microlenses. The autocorrelation design term is then

$$p(\boldsymbol{\theta}) = \sum_n \left\| D\left[\mathbb{F}\left\{\mathbf{h}(u, v; \boldsymbol{\theta}, z_n) \star \mathbf{h}(u, v; \boldsymbol{\theta}, z_n)\right\} - |H|^2\right] \right\|_2^2, \tag{5.7}$$

where $\mathbb{F}\{\cdot\}$ is the 2D discrete Fourier transform.

The total loss is the weighted sum of the two terms:

$$f(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) + \tau_0 q(\boldsymbol{\theta}), \tag{5.8}$$

where τ_0 is a tuning parameter to control their relative importance. To initialize, we randomly generate 5,000 heuristically-designed candidate phase masks, each with 36 microlenses spaced according to Poisson disc sampling across the GRIN aperture stop. The focal lengths are distributed dioptrically between the minimum and maximum values computed in the *Multifocal Design* section. The best candidate from these 5,000 is then optimized using gradient descent applied to $f(\boldsymbol{\theta})$. This is implemented in Tensorflow Eager to enable GPU-accelerated automatic differentiation.

The results of our optimized design are shown in Fig. 5.3, where we compare our optimized mask to the random multifocal design that scored worst during initialization, and a regular unifocal array. The optimized design has the best axial cross-coherence (Fig. 5.3(b)), with the random array having worse off-diagonal terms. Hence, in the 3D reconstructions

(Fig. 5.3(c)) the optimized design performs slightly better than the random design. The regular microlenses produce large off-diagonal peaks in the cross-coherence which manifests as poor 3D reconstruction performance off-focus.

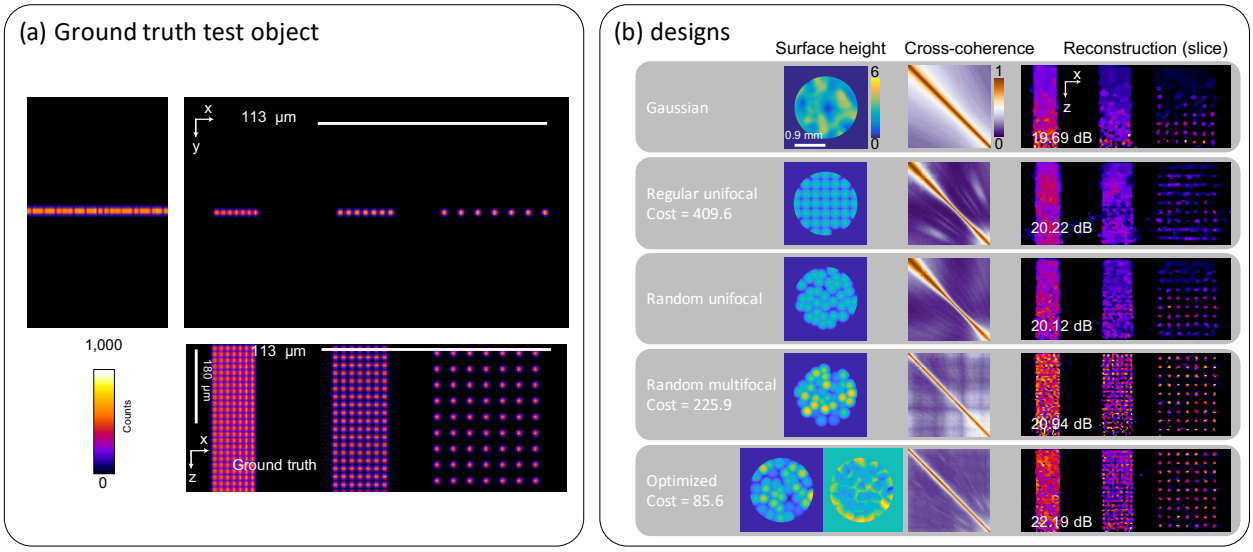


Figure 5.3: Comparison of our optimized phase mask with random multifocal and regular microlens arrays: (a) ground truth test object consisting of differently-spaced point sources (x-spacings of 3.5 μm and 7 μm, z-spacings of 19.4 μm and 38 μm). (b) comparison of different phase mask designs. The first column shows surface heights for varoius masks, the second column shows the cross coherence matrix for each over the target volume, and the rightmost column shows x-z slice of the reconstruction using that design. The Gaussian diffuser performs the worst, and has a poor cross-coherence matrix. The regular unifocal microlenses is only slightly better, but has poor performance in and out of focus. The random unifocal design improves the in focus performance, but due to defocus in the microlenses, it only works over a short depth range. Using a multifocal design improves the out of focus performance, Finally, the optimized design qualitatively is similar to the random multifocal, but has lower error as seen in the high PSNR score.

Chapter 6

Optimized masks for miniaturized single-shot 3D fluorescence microscopy

This is work done jointly with Kyrollos Yanny, William Liberti, Sam Dehaeck, Kristina Monakhova, Fanglin Linda Liu, Konlin Shen, Ren Ng, and Laura Waller. It is based on [142].

Abstract

Miniature fluorescence microscopes are a standard tool in systems biology. However, wide-field miniature microscopes only capture 2D information, and modifications that enable 3D capabilities increase size and weight, and have poor resolution outside a narrow depth range. Here, we achieve 3D capability by replacing the tube lens of a conventional 2D Miniscope with an optimized multifocal phase mask at the objective's aperture stop. Placing the phase mask at the aperture stop significantly reduces the size of the device and varying the focal lengths enables uniform resolution across a wide depth range. The phase mask encodes 3D fluorescence intensity into a single 2D measurement and the 3D volume is recovered by solving a sparsity-constrained inverse problem. We provide methods for designing and fabricating the phase mask and an efficient forward model that accounts for the field-varying aberrations in miniature objectives. We demonstrate a prototype that is 17 mm tall and weighs 2.5 grams, achieving 2.76 μm lateral and 15 μm axial resolution across most of the $900 \times 700 \times 390$ μm^3 volume at 40 volumes per second. The performance is validated experimentally on resolution targets, dynamic biological samples, and mouse brain tissue. Compared to existing miniature single-shot volume-capture implementations, our system is smaller, lighter, and achieves more than 2× better lateral and axial resolution throughout a 10× larger usable depth range. Our microscope design provides single-shot 3D imaging for applications where a compact platform matters, such as volumetric neural imaging in freely-moving animals and 3D motion studies of dynamic samples in incubators and lab-on-a-chip devices.

Introduction

Miniature widefield fluorescence microscopes enable important applications in systems biology - for example, optical recording of neural activity in freely-moving animals [45, 80, 61, 49], and long-term *in situ* imaging within incubators and lab-on-a-chip devices. These miniature microscopes, commonly called 'Miniscopes', are developed by a vibrant open-source community [131] and made of 3D printed parts and off-the-shelf components. While the Miniscope is designed for 2D fluorescence imaging only, many applications can benefit from imaging 3D structure.

Here, we present a new single-shot 3D miniature fluorescence microscope, termed *Miniscope3D*, that is not only smaller and lighter weight than miniaturized plenoptic microscopes like the MiniLFM, but also achieves better resolution over a larger volume. It is designed as a simple hardware modification to the widely-used UCLA Miniscope [131], replacing the tube lens with an optimized phase mask (see Chapter 5) placed directly at the aperture stop (Fourier plane) of the objective lens (Fig. 6.1). The phase mask consists of a set of multi-focal nonuniformly-spaced microlenses, optimized such that each point within a 3D sample generates a unique high-frequency pattern on the sensor, encoding volumetric information in a single 2D measurement. The 3D volume is recovered by solving a sparsity-constrained compressed sensing inverse problem, enabling us to recover 24.5 million voxels from a 0.3 megapixel measurement. Our algorithm assumes the sample to be sparse in some domain, which is valid for a general class of fluorescent samples. We demonstrate the capabilities of our microscope by imaging fluorescent resolution targets, freely swimming biological samples, scattering mouse brain tissue, and optically cleared mouse brain tissue. We also validate the accuracy of our reconstructions against two-photon microscopy and discuss the limitations of our method.

To achieve high-quality imaging in a small, low-weight device, a number of technical innovations were developed. Placing the phase mask in Fourier space (instead of image space) significantly improves compactness, and also reduces computational burden [88, 114, 51]. Varying the focal lengths of the microlenses enhances the uniformity of resolution across depth, as compared to implementations like MiniLFM. Because we use an optimized forward model and calibration scheme to account for the field-varying aberrations inherent to miniature objectives, we are able to add 3D capabilities to the 2D Miniscope, at a cost of only a small loss of lateral resolution, and lower signal-to-noise ratio (SNR). Our algorithm unites optical theory with compressed sensing in a general way that can allow others to design and fabricate optimized phase masks for their applications. The main contributions of this work are:

- A new miniature 3D microscope architecture that improves upon MiniLFM, achieving significantly better resolution across a 10× larger depth range, while reducing overall device size.
- A prototype, based on easily available parts, 3D printing, and open-source designs, that weighs 2.5 grams and achieves 2.76 μm lateral and 15 μm axial resolution across most of the $900 \times 700 \times 390$ μm^3 volume at 40 volumes per second.

- Design principles for optimizing phase masks for 3D imaging and a high-quality fabrication method using two-photon polymerization in a Nanoscribe 3D printer.
- An efficient calibration scheme and reconstruction algorithm that accounts for the field-varying aberrations inherent in miniaturized objective lenses.

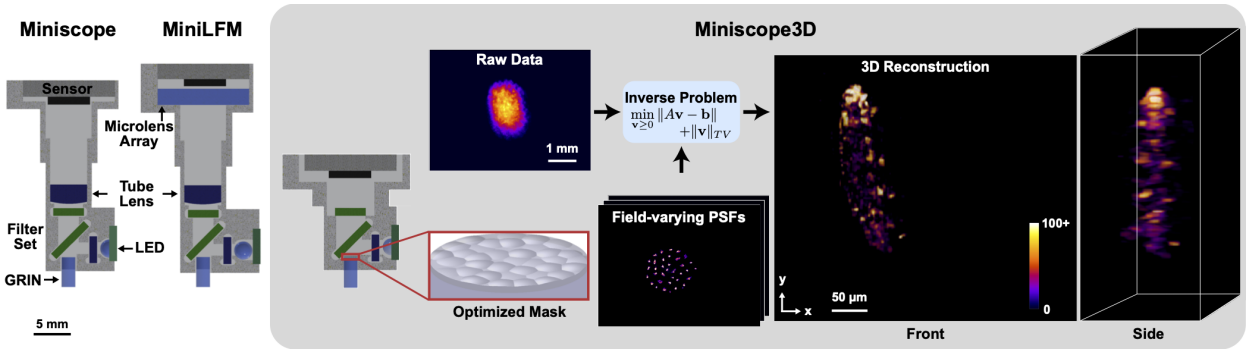


Figure 6.1: Miniscope3D system overview. As compared to previous Miniscope and MiniLFM designs, our Miniscope3D is lighter weight and more compact. We remove the Miniscope's tube lens and place a 55 μm thick optimized phase mask at the aperture stop (Fourier plane) of the GRIN objective lens. A sparse set (64 per depth) of calibration point spread functions (PSFs) is captured by scanning a 2.5 μm green fluorescent bead throughout the volume. We use this dataset to pre-compute an efficient forward model that accurately captures field-varying aberrations. The forward model is then used to iteratively solve an inverse problem to reconstruct 3D volumes from single-shot 2D measurements. The 3D reconstruction here is of a freely-swimming fluorescently-tagged tardigrade.

Materials and Methods

System Theory

Miniscope3D encodes volumetric information via a thin phase mask placed at the aperture stop of the gradient index (GRIN) objective lens (see Fig. 6.1). The goal of our design is to optimize the microscope optics for compressed sensing, enabling capture of a large number of voxels from a small number of sensor pixels. To achieve this, the phase mask comprises an engineered pattern of multifocal microlenses, designed such that each fluorescent point source in the scene produces a unique high-frequency pattern of focal spots at the sensor plane, thus encoding its 3D position. The structure and spatial frequencies present in this pattern, termed the *point spread function* (PSF), determine our reconstruction resolution at that position; theory for these limits is presented in the *Lateral Resolution* section below.

Figure 6.2 shows how our PSF changes with the lateral and axial position of a point source in the object space. As the point source moves laterally, the PSF translates (Fig. 6.2(b)). In an idealized microscope with the phase mask in Fourier space, the system would be shift-invariant [88, 114]; however, because of the inherent aberrations in the GRIN lens, the pattern also slightly changes structure as it shifts. As the point source moves axially, the

overall PSF changes size and different spots come into focus (Fig. 6.2(a)), because we use a diversity of microlens focal lengths in our phase mask. As discussed in the section on *Multifocal Design*, this ensures that the PSFs at a wide range of depths all contain sharp focal spots, unlike unifocal microlenses. To maximize the performance of our system, we optimize the spacing and focal lengths of the microlenses, as described in the *Phase Mask Optimization* section.

Our distributed, unique PSFs satisfy the multiplexing requirement of compressed sensing. Hence, we utilize sparsity-constrained inverse methods to recover the voxelized sparse 3D fluorescence emission, $\mathbf{v}$, from a single 2D sensor measurement, $\mathbf{b}$. To do this, we model $\mathbf{b}$ as a linear function of $\mathbf{v}$, denoting the measurement process as $\mathbf{b} = A\mathbf{v}$. Here, A is the measurement matrix, a linear operator that captures how each voxel maps to the sensor. Provided the sample is sparse in some domain, we reconstruct the volume by solving the sparsity-constrained inverse problem:

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}\geq 0} \|A\mathbf{v} - \mathbf{b}\|_2^2 + \tau\|\Psi\mathbf{v}\|_1, \tag{6.1}$$

with Ψ being a sparsifying transform (e.g. 3D gradient, corresponding to TV regularization) and τ being a tuning parameter.

Equation 6.1 can be solved using a variety of iterative methods; we use Fast Iterative Shrinkage Thresholding (FISTA) [15]. This requires repeatedly applying A and its adjoint. To make this computationally feasible for high megavoxel systems like ours, we need an efficient representation for A. A shift-invariant forward model is extremely computationally efficient because A becomes a convolution matrix [11, 12, 68]. It also requires only a single PSF calibration image, from which the PSFs at all other positions can be inferred. Unfortunately, miniature integrated systems like ours are not shift invariant, due to the off-axis aberrations inherent to compact objectives. To account for this, in the following sections we develop a field-varying forward model and a practical calibration scheme that account for aberrations with minimal added computational cost.

Field-varying Forward Model

Because aberrations in the GRIN lens of the Miniscope render the shift-invariant model invalid, we need to both measure and model how the PSF changes across the FoV. Explicitly measuring the PSF at each position within the volume is infeasible, both in terms of amount of calibration data and computational burden of reconstruction. It is also unnecessary since the PSF structure changes slowly across the FoV. Instead, our calibration scheme samples the PSF sparsely across the field and uses a weighted convolution model to estimate the PSF at other positions [42]. We capture 64 PSF measurements at each depth, then use them to predict the full set of over 300,000 PSFs. Our forward model thus only requires computing a limited number of convolutions (typically 10-20) and achieves 2.2× better resolution and better quality than the shift-invariant model (see Fig. 6.2(c)).

Our field-varying forward model approximates A using a weighted sum of shift-invariant (convolution) kernels. We treat the volumetric intensity as a 3D grid of voxels, denoted

$\mathbf{v}[x, y, z]$. A voxel at location $[x, y, z]$ produces a PSF on the sensor, $\mathbf{h}[u, v; x, y, z]$, where $[u, v]$ indexes sensor rows and columns. For ease of notation we will assume the system has magnification $M = 1$ and apply appropriate scaling to the solution after 3D image recovery. We also assume $\mathbf{v}$ has finite axial and lateral support. By treating the voxels as mutually incoherent, the measurement will be a linear combination of PSFs:

$$\begin{aligned} \mathbf{b}[u, v] &= \sum_{z} \sum_{x,y} \mathbf{v}[x, y; z]\mathbf{h}[u, v; x, y, z] \\ &= A\mathbf{v}, \end{aligned} \tag{6.2}$$

where the bounds of the sums implicitly contain the sample. To capture field-varying behavior, we seek to model the PSF from each voxel as a weighted sum of K shift-invariant kernels [42]. The kernels, $\mathbf{g}_r[u, v; z]$, and weights, $\mathbf{w}_r[x, y, z]$, which will be described below, should be chosen to represent all PSFs accurately with the smallest possible K. Mathematically, the forward model can be written as:

$$\mathbf{h}[u, v; x, y, z] = \Lambda[u, v] \sum_{r=1}^{K} \mathbf{w}_r[x, y, z]\mathbf{g}_r[u - x, v - y; z], \tag{6.3}$$

where $\Lambda[u, v]$ is an indicator function that selects only the values that fall within the sensor pixel grid. In other words, the PSF from position $[x, y, z]$ is modeled by shifting the kernels, $\{\mathbf{g}_r[u, v; z]\}$ $r = 1 \dots K$, associated with depth z, to be centered at the PSF location on the sensor, $[u, v] = [x, y]$. Then, each kernel is assigned a field-dependent weight, $\mathbf{w}_r[x, y, z]$, and the weighted kernels are summed over r. Note that this motivates the placement of the phase mask in the aperture stop. By ensuring that all field points fully illuminate the mask, the system will be close to shift-invariant, which will keep the necessary number of kernels low.

To find the kernels and weights that best represent all of the PSFs, first consider each PSF in a coordinate space relative to the chief ray. We do this by centering each measured PSF on-axis:

$$\mathbf{h}[u + x, v + y; x, y, z] = \sum_{r=1}^{K} \mathbf{w}_r[x, y, z]\mathbf{g}_r[u, v], \tag{6.4}$$

where $[x, y]$ is the chief ray spatial coordinate at the sensor. We assume that the calibration procedure will capture N PSFs across the field, $\{\mathbf{h}[u, v; x_i, y_i, z]\}$ $i = 1 \dots N$, for each depth z. We estimate the chief ray coordinate $[x, y]$ of off-axis PSFs by cross-correlating each with the on-axis PSF. The off-axis measurements are then shifted on-axis, vectorized, and combined into a registered PSF matrix, denoted H. For smoothly varying systems, H will be low rank and can be well approximated by solving

$$\hat{G}, \hat{W} = \underset{G,W}{\operatorname{argmin}} \|GW - H\|_2^2, \tag{6.5}$$

where $G \in \mathbb{R}^{M_p \times K}$ and $W \in \mathbb{R}^{K \times N}$ for a sensor with M_p pixels. The optimal rank-K solution can be found by computing the the K largest values of the singular value decomposition

(SVD) of H. The r-th column of the left singular vector matrix, $\hat{G}$, contains the kernel $\mathbf{g}_r[x, y; z]$ in vectorized form. Similarly, combining the singular values with the right singular vector matrix produces $\hat{W}$, of which the r-th row contains the optimal weights $\mathbf{w}_r[x_i, y_i, z]$ for voxel $[x_i, y_i, z]$. Empirically, we find that the weights vary smoothly across the field, so we use natural neighbor interpolation to estimate the weights between sampled points. After testing the number of sample points per depth (N) empirically, we find 64 to be sufficient for our system.

The computational-efficiency of this model can be analyzed by substituting Eq. 6.3 into Eq. 6.2, yielding:

$$\begin{aligned}\mathbf{b}[u, v] &= \sum_z \sum_{x,y} \mathbf{v}[x, y, z]\Lambda[u, v] \sum_{r=1}^{K} \mathbf{w}_r[x, y, z]\mathbf{g}_r[u - x, v - y; z] \\ &= \Lambda[u, v] \sum_z \sum_{r=1}^{K} \left\{ (\mathbf{v}[x, y, z]\mathbf{w}_r[x, y, z]) \overset{[x,y]}{*} \mathbf{g}_r[x, y; z] \right\} [u, v] \,,\end{aligned} \tag{6.6}$$

where $\overset{[x,y]}{*}$ denotes discrete linear convolution over the lateral variables. In practice, each convolution can be implemented using a combination of padding and FFT-convolution, while $\Lambda[u, v]$ represents a crop [11]. Note that the summation over z assumes no voxel is partially occluded. Because this model comprises K point-wise multiplications and K 2D convolutions per depth, it is approximately $K-$times slower than a shift-invariant model. Hence minimizing K via choice of weights and kernels, or by reducing aberrations in the hardware, improves computational efficiency.

Calibration

Experimentally, our calibration procedure captures PSF images of a 2.5 μm green fluorescent bead at 64 equally-spaced points across the FoV, for each depth. Empirically, we find that the singular values decay quickly and a model with rank between $K = 10$ and $K = 20$ is sufficient for our system. Note that we can trade-off the speed and accuracy of our model by varying K, but the decomposition need only be performed once. This method allows characterization of an extremely large matrix by only capturing a relatively small number of images. For example, our typical calibration requires 80 depths. Densely sampling every PSF using a 0.3 megapixel sensor would require 24 million calibration images (300,000 per depth) and terabytes of storage. In contrast, our method enables calibrating this entire volume using only 80 depths $\times$ 64 images/depth $= 5,120$ images, which takes 2 hours to capture using automated stages and requires a few gigabytes to store.

Reconstruction Algorithm

In solving Eq. 6.1 we use sparsifying transform $\Psi = [\nabla_x \nabla_y \nabla_z]^\intercal$, which corresponds to 3D anisotropic TV regularization, promoting sparse 3D gradients in the reconstruction. The regularization parameter, τ, controls the balance between the data fidelity and the sparse 3D gradients prior. In practice, we hand-tune τ on a range of test data, then leave it fixed

for subsequent captures (see supplement Fig. 8.9). We solve Eq. 6.1 using FISTA [15], with the fast, subiteration-free parallel proximal method [65]. Computationally, our method has similarities to light-field deconvolution [24], but because our PSF is not periodic and our focal lengths are not all the same, we are able to remove the need for aperture matching and achieve higher resolution across a larger volume. In order to solve Eq. 6.1, we compute the linear forward and adjoint matrix-vector multiplies using FFT-convolution. A typical reconstruction takes 1-3k iterations, and runs in 8-24 minutes on a GPU RTX 2080-Ti using MATLAB.

Phase Mask Fabrication

Since our phase mask designs can be tailored to specific applications with different resolution requirements and volumes-of-interest, the ability to rapidly generate phase mask prototypes is very useful. Recently, the Nanoscribe two-photon polymerization 3D printer has been shown to print free-form microscale optics on-demand [128]. However, in its current implementation, Nanoscribe uses planar galvanometric scanning to polymerize the resist, resulting in a limited FoV (diameter of approximately 350 μm with the 25$\times$ Nanoscribe objective). If larger objects need to be printed, several blocks need to be stitched together by moving the substrate with a mechanical stage. Stitching artifacts from this process can seriously impact the produced object [32], usually by causing rectangular or hexagonal blocking artifacts. As can be seen in Fig. 6.3(a), rectangular seams going trough the center of the microlenses can be very detrimental to our design.

One solution to this is an adaptive stitching algorithm that has been demonstrated for slender objects and a non-overlapping microlens array [32]. Here, we propose a new height-based segmentation algorithm capable of placing the stitching seams in the overlapping region between the overlapping microlenses (Fig. 6.3(a)). This is based on the local height functions for each microlens, described in the *Phase Mask Parameterization* section. Each of these functions has a region where they result in the largest values and this region is precisely the printing block that will be printed from that microlens center location (see supplement 8.6). Once the adaptive stitching mask is obtained, the writing instructions per block can be generated using TipSlicer [111]. Figure 6.3(b) compares the designed and experimental PSFs at three depth planes, showing a good match with some degradation at the end of the volume.

Device Assembly

Our prototype Miniscope3D system consists of a custom phase mask, a CMOS sensor (Ximea MU9PM-MH), fluorescent filter set (Chroma ET525/50m, T495lpxr, ET470/40x), GRIN lens (Edmund Optics 64-520), and half-ball lens (Edmund 47-269), with a 3D-printed optomechanical housing. The 55 μm thick phase mask is glued to the back surface of the GRIN lens using optical epoxy. Note that our experimental PSF calibration accounts for slight misalignment in the phase mask. The final device is 17 mm tall and weighs 2.5 grams.

Results

We characterize the performance of our computational microscope with samples of increasing complexity, capturing dynamic 3D recordings at frame rates of up to 40 volumes per second.

Resolution Characterization: Lateral resolution is measured at different depths by imaging a fluorescent resolution target every 10 μm axially and determining the smallest resolved group by eye. Figure 6.4(a) demonstrates 2.76 μm uniform lateral resolution over 270 μm in depth. The resolution degrades to 3.9 μm over the next 120 μm in depth, for a total usable depth range of 390 μm. This relatively uniform resolution through a wide depth range is due to our multifocal design. Axial resolution is determined by imaging a thin layer of 4.8 μm fluorescent beads at different depths and using Rayleigh criterion (at least a 20% dip between the peaks of the two reconstructed points) to determine resolution. Raw data from multiple depths are added to synthesize a measurement of two layers of beads with varying separations (see supplement 8.6). We achieve 15 μm axial resolution across the entire 390 μm depth range, which matches the axial full-width-half-maximum (FWHM) in the reconstructions of the 3D fluorescent beads sample in Fig. 6.4(b).

Two-Photon Verification: To validate the accuracy of our results, we compare against two-photon microscopy, which is considered ground truth. Figure 6.4(b) shows results for a 160 μm thick sample of 4.8 μm green fluorescent beads. Miniscope3D accurately recovers all the beads in the volume, after visually adjusting for tip/tilt misalignment in post-processing.

Mouse Brain Tissue: Next, we show feasibility for neuro-biological samples by imaging post-fixed mouse brain slices where GFP is expressed in a sparse population of neurons throughout the sample. Figure 6.5(a) shows reconstructions from two 100 μm thick scattering samples from different parts of the hippocampus, and Fig. 6.5(b) shows results from a 300 μm thick optically cleared section. In the 300 μm slice, dendrites can be seen running across the reconstruction axially (~1 μm features), and individual cell bodies appear at distinct depths (see **Video** 1).

Dynamic Biological Samples: Finally, we image dynamic samples of freely-swimming SYBR-green stained tardigrades at a maximum of 40 frames per second. Figure 6.5(c) shows maximum intensity projections of the reconstructed videos at different time points from two different recordings. The reconstructions show that Miniscope3D can track freely-moving biological samples at high spatial and temporal resolution (see **Videos** 2, 3, 4, 5, & 6).

Discussion

Our device is designed with compressed 3D imaging and miniaturization in mind. For some 2D imaging applications where the loss of SNR (see supplement Fig. 8.10) and lateral resolution (2.76 μm vs 2 μm) are acceptable, our device may have advantages over 2D Miniscope, due to its smaller size (17 mm vs. 23.5 mm tall) and weight (2.5 grams vs. 3 grams), or the ability to digitally refocus via 3D reconstruction. However, we expect that most applications of Miniscope3D will be for true 3D microscopy, so we mainly compare our

specifications to MiniLFM, which is considered the gold-standard for single-shot miniature 3D fluorescence imaging.

Miniscope3D offers multiple improvements over MiniLFM. First, using multifocal microlenses (as opposed to unifocal in LFM) allows us to achieve better lateral resolution ($2.76 - 3.9\ \mu m$) across a larger depth range ($390\ \mu m^3$). In contrast, MiniLFM [121] demonstrated *best-case* lateral resolution of 6 μm at a particular depth and, while their resolution at other depths was not reported, we predict that their unifocal microlens design will result in lateral resolution that degrades significantly beyond 40 μm depth, based on previous analysis [24] and that in the *Multifocal Design* section below. We estimate that our Miniscope3D provides approximately 10× increase in the usable measurement volume over MiniLFM, with 2.2× better peak lateral resolution. Taken together, our Miniscope3D reconstructs approximately 50× more usable voxels than MiniLFM, significantly improving the utility of the device. This improved performance comes in a hardware package that is smaller than MiniLFM (17 mm tall vs. 26 mm tall) and lighter weight (2.5 grams vs. 4.7 grams), because we replace the heavy doublet tube lens and the microlens array assembly with a thin phase mask. This will be particularly valuable in head-mounted experiments with freely-moving animals.

Both our method and MiniLFM make sparsity assumptions on the sample in order to solve the inverse problem to recover a 3D volume from a 2D image. We require the sample to be sparse in some domain, meaning that there is some representation of the sample that has many zeros in its coefficients [26, 11]. Fluorescence imaging is a good candidate for such priors, since most biological samples are sparsely labelled. Because we optimize the microscope optics explicitly for single-shot 3D imaging, typical sparsity priors such as native sparsity, sparse 3D gradients (*Total Variation (TV)*, as used in this paper), or sparse wavelets work well in our system. The MiniLFM is designed specifically for neural activity tracking and so makes further structural and temporal sparsity assumptions, which improves their axial resolution from 30 μm (single-shot performance) to $15\mu m$ (temporal video processing performance). In contrast, our Miniscope3D achieves 15 μm single-shot axial resolution, across a large depth range, and could presumably improve upon that by incorporating temporal application-specific priors. In this paper, however, we aim to record highly dynamic samples (see supplementary videos) and so only impose sample sparsity. We demonstrate the generality of our approach experimentally with samples that exhibit different levels of sparsity (Fig. 6.4,6.5), achieving resolution sufficient for single-neuron imaging. As sparsity decreases, image quality and resolution degrade smoothly (see supplement Fig. 8.11), roughly following previous theoretical analyses [26, 25, 11].

Scattering is a limitation for all single-photon microscopes, including ours. For applications such as neural imaging and studying the 3D motion of freely-swimming samples like C. elegans or tardigrades, the small amount of scattering should not hinder resolution. However, as the imaging depth within the scattering medium increases, we expect the resolution to degrade in a way similar to other single-photon microscopes. We show experimental reconstructions with and without scattering for the 100 μm thick scattering mouse brain tissue, and the 300 μm thick cleared brain tissue. Both reconstructions achieve single-neuron resolution.

Another limitation of our model is that it assumes no partial occlusions. This is a common limitation of 3D recovery methods in fluorescence microscopy (e.g. double helix [104], light field deconvolution microscopy [24], 3D localization microscopy) and generally works well in non-absorbing fluorescent samples. Modeling occlusions would be valuable in many practical situations, but remains a challenging problem.

Accessibility was a key consideration in our Miniscope3D design. By building on the popular open-source Miniscope platform, our method can be easily adopted into existing experimental pipelines. Any of the 450 labs currently using the 2D Miniscope can upgrade to our 3D prototype with minimal effort. Also, our method for 3D printing custom phase masks can enable others to fabricate their own mask designs tailored to particular applications. Because experimental results are in good agreement with our theoretical design and analysis, we are confident that our design theory can provide a useful framework for future customization of single-shot 3D systems.

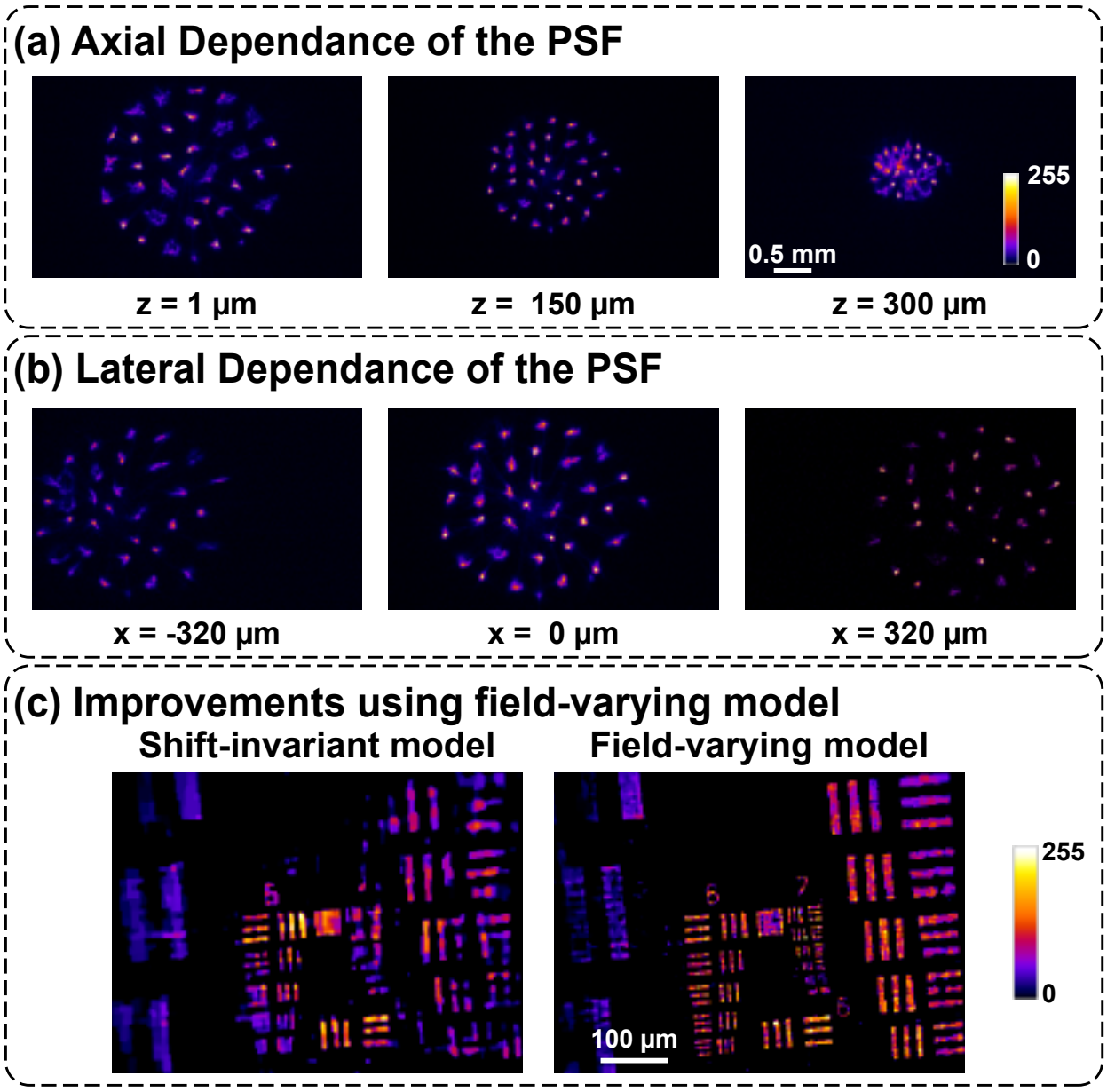


Figure 6.2: Each 3D voxel maps to a different PSF: (a) As a point source translates axially, the PSF scales and different spots come into focus. (b) As a point source translates laterally, the PSF shifts and incurs field-varying aberrations which destroy shift invariance. (c) When a shift-invariant approximation is made, reconstructions of a fluorescent resolution target (at $z = 250\ \mu m$) display worse resolution (6.2 μm resolution) and more artifacts than when our field-varying model is used (2.76 μm resolution).

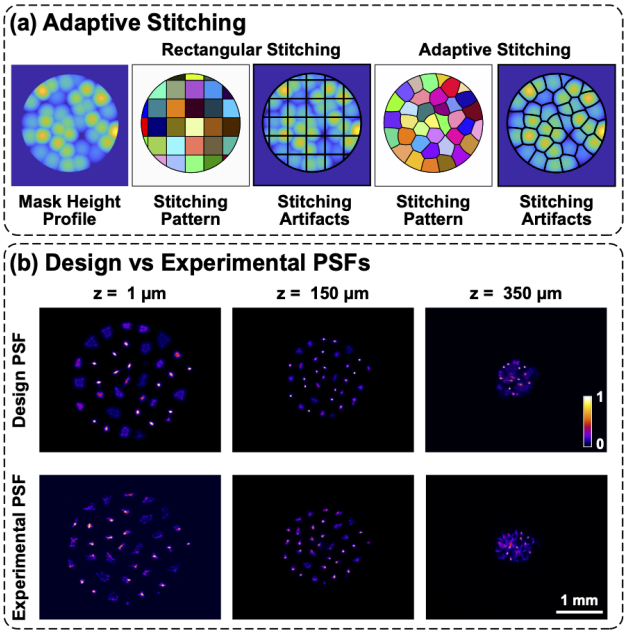


Figure 6.3: Phase mask fabrication with Nanoscribe: (a) Rectangular stitching leads to seams (black lines) going trough the many microlenses, while adaptive stitching puts the seams at the boundaries of the microlenses to mitigate artifacts. (b) Comparison between designed and experimental PSFs at a few sample depths, showing good agreement, with slight degradation at the edge of the volume.

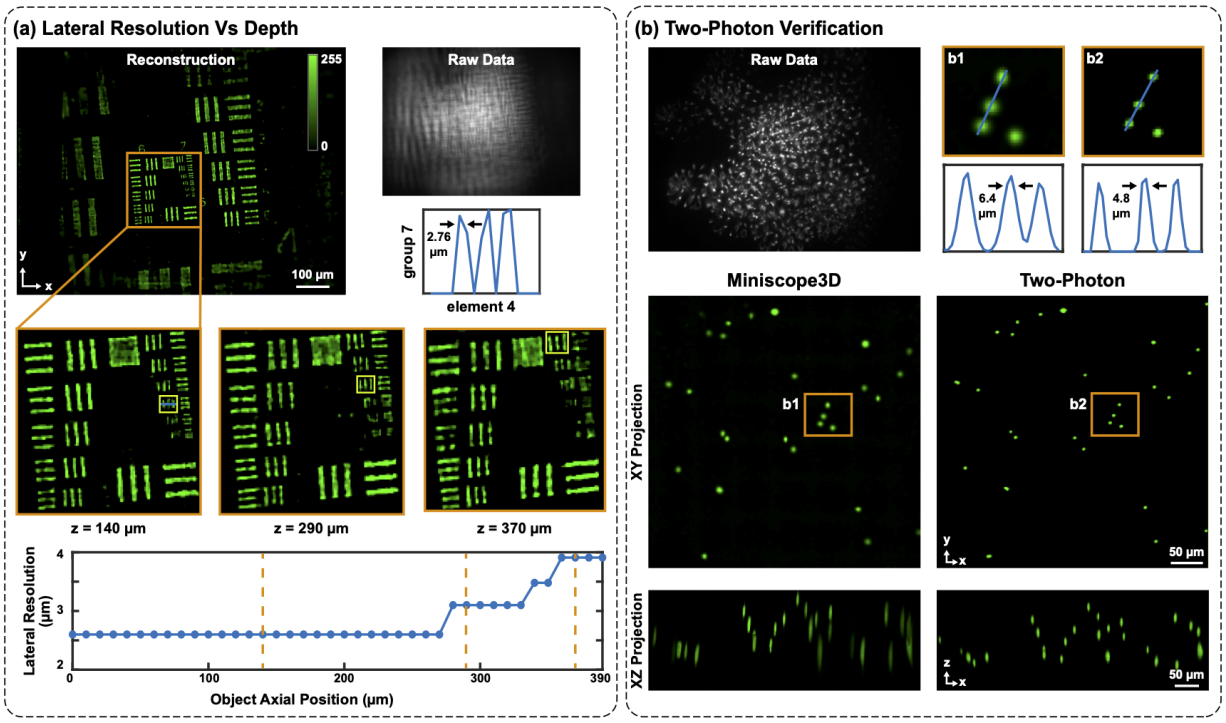


Figure 6.4: Experimental characterization: (a) Reconstructions of a fluorescent USAF target at different axial positions to determine depth-dependent lateral resolution. We recover 2.76 μm resolution across most of the 390 μm range of depths, with a worst case of 3.9 μm (dashed orange lines mark inset locations and yellow boxes on insets indicate smallest resolved groups). Note that the resolution target has discrete levels of resolution that result in jumps in the data and resolution refers to the gap between bars, not the line-pair width. (b) Reconstruction of a 160 μm thick sample of 4.8 μm fluorescent beads, as compared to a two-photon 3D scanning image (maximum intensity projections in yx and zx are shown). Our system detects the same features, with a slightly larger lateral spot size.

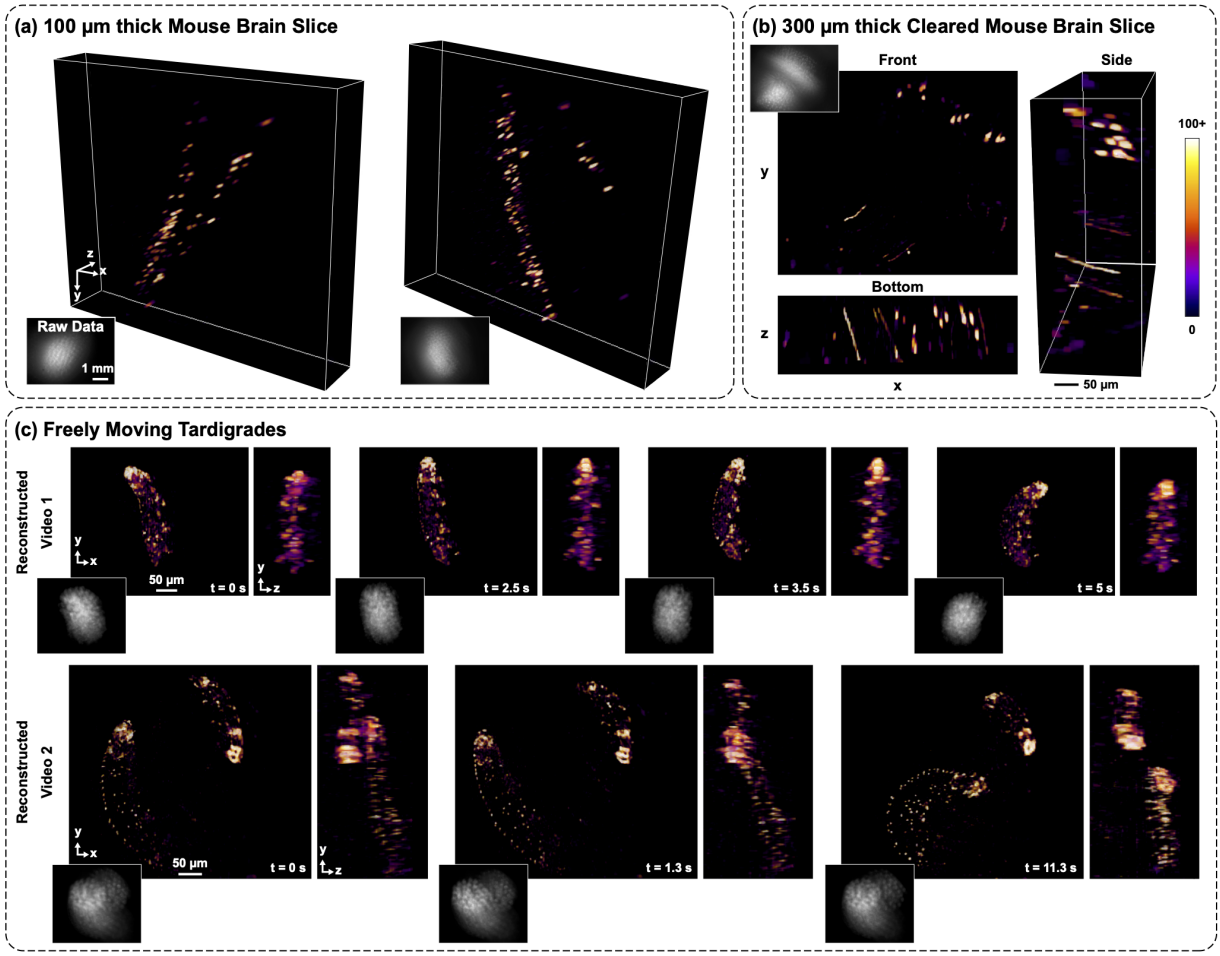


Figure 6.5: Experimental 3D reconstructions of (a) GFP-tagged neurons in two different samples of 100 μm thick fixed mouse brain tissue, and (b) 300 μm thick optically cleared mouse brain slice. We clearly resolve dendrites running across the volume axially (see **Video** 1). All mouse brain volume reconstructions are $790 \times 617 \times 210$ μm^3. (c) Maximum intensity projections from several frames of the reconstructed 3D videos of two different samples of freely moving tardigrades captured at a maximum of 40 frames per second (see **Video** 2 & 3).

Chapter 7

Focal plane diffuser encoding of light fields

This is work done jointly with Sylvia Necula, Ren Ng, and Laura Waller and is published as [8].

7.1 Introduction

Recording and processing of 4D light fields offers new capabilities over traditional 2D imaging. These include the ability to compute a different focus and depth of field after the fact, change the viewpoint slightly, compute depth and computationally correct for optical aberrations in the camera's lens. Various approaches have been studied to project the 4D light field onto a 2D sensor such that the light field can be inferred from a single-shot. The two most common approaches are based on microlens arrays and attenuation masks placed at a small distance from the image sensor. Microlens arrays use regular grid of lenslets to refractively encode the light field onto the sensor. Attenuation masks encode the light field into shadow patterns on the sensor, providing enhanced resolution, but at the cost of absorbing a portion of the light [133] [94] [64].

In this paper we generalize both of these approaches to use a transparent phase plate (e.g. a diffuser). Compared with microlens arrays, we allow arbitrary height maps. Compared with attenuation masks, we allow a similar coding effect, but with higher light throughput. Such diffusers provide an inexpensive and flexible means for single-shot light field recording using an off-the-shelf diffuser. A challenge in utilizing such diffusers is that they are generally diffractive, producing speckles that exhibit significant wave-optical effects due to interference. We show here a theoretical analysis of when it is appropriate to use wave-optics versus ray optics models for interpreting light fields encoded by a phase plate. Leveraging this, our camera is designed such that the image synthesis and reconstruction are correctly described by a ray optics model, independent of the object or illumination coherence. We further present a wave-optics calibration routine, based on the Transport of Intensity Equation (TIE) [127], that recovers the phase surface height map from images captured through focus.

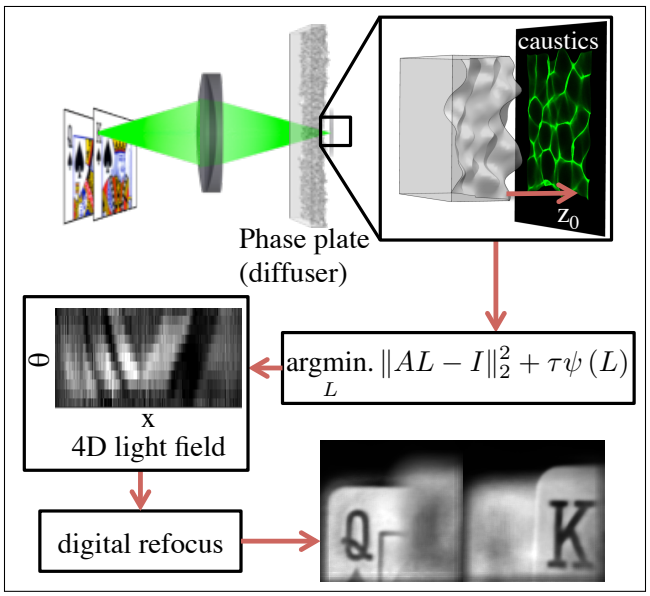


Figure 7.1: Pipeline for recording and reconstructing light fields with phase plates (a diffuser). The object light passes through an imaging lens and the phase plate, then propagates to the sensor, where caustics encode spatial and angular information. A linear inverse problem is solved to reconstruct the light field, which contains 3D information, enabling digital refocus, among other benefits.

We use ray tracing to build a linear model for the system, based on the phase measurements, which we then invert to recover the 4D light field. Lastly, we demonstrate the efficacy of these methods by showing experimental results.

7.2 Theory

Consider the 4D light field, $L(x, y, \theta, \phi)$, inside a camera body after passing through a primary imaging lens having numerical aperture less than 1. Each ray is described by two lateral coordinates, (x, y), and two angular coordinates (θ, ϕ), corresponding to x and y, respectively. The light at the imaging plane then passes through a non-absorbing phase plate (e.g. a diffuser), and propagates a distance z_0 to the 2D sensor plane, where it is recorded as an intensity image. The system architecture is illustrated in Figure 7.1.

As an example application of this approach, we use an inexpensive off-the-shelf Light Shaping Diffuser [90]. These diffusers are thin pieces of polymer with refractive index $n \approx 1.5$ that are planar on the input side and have an output surface that can be modeled as a smooth random Gaussian surface, described by a height field, $D(x, y)$:

$$D(x, y) = s\left[R(x, y) * K(\sigma)\right], \tag{7.1}$$

where s is a unitless scaling factor, $K(\sigma)$ is a zero-mean Gaussian smoothing kernel having full-width half-maximum (FWHM) value of σ and $R(x, y)$ a set of random height values chosen from the normal distribution at each discrete sample location (x, y). It is assumed

that σ is greater than the wavelength of light, λ, thereby avoiding sub-wavelength scattering effects. We show in Section 7.2 that diffusers create high-contrast intensity patterns (caustics) at certain distances. These patterns are unique to particular regions on the diffuser surface, thus they encode multiplexed spatial and angular information in an invertible way.

Wave Optics Model

The phase plate can be thought of as a thin transparency which imparts a spatially-varying phase delay, $\phi_D(x, y)$, onto any wave passing through it. Consider a coherent incident wave having amplitude $A(x, y)$ and phase $\phi_i(x, y)$. The wavefront exiting the phase plate (at $z = 0$) will be the product of the incident wave's complex-field and the complex transmittance of the phase plate [47],

$$E(x, y, z = 0) = A(x, y)e^{i\left[\phi_i(x,y)+\phi_D(x,y)\right]}. \tag{7.2}$$

Assuming the phase plate has homogeneous index of refraction, n, the phase delay is directly proportional to the height map of the phase plate

$$\phi_D(x, y) = \frac{2\pi\Delta n}{\lambda} D(x, y), \tag{7.3}$$

where Δn is the refractive index difference between the phase plate and surrounding medium. For plane wave illumination, $A(x, y)$ and $\phi_i(x, y)$ are constant, so Equation (7.2) simplifies to

$$E(x, y, z = 0) = \exp\left[i\phi_D(x, y)\right]. \tag{7.4}$$

The resulting complex-field at the sensor, $E(x_0, y_0, z_0)$, is predicted using Fresnel diffraction theory [47]. Finally, the intensity at the sensor is proportional to absolute value squared of the complex-field, $I(\xi, \eta; z_0) \propto |E(\xi, \eta; z_0)|^2$.

In this wave-optical model, the gradient of the incident beam's phase describes the local angle of propagation, according to the Poynting vector description of energy flow. For partially coherent (or incoherent) light, however, the optical field cannot be described by a single complex field; rather, it is the incoherent superposition of many [20]. The Wigner function provides a wave-optical analog to the light field [144]. Still, it is significantly more complicated than a ray optics model, which is preferred where accurate.

Ray Optics Model

Ray tracing approaches are generally thought to be only valid for incoherent imaging, whereas diffractive effects require wave optics. However, the diffractive nature of the phase masks used here does not always imply that a full wave-optical model is necessary. We will show here that for sufficiently small diffuser-to-sensor distances, ray optics is a suitable approximation, irrespective of whether the object is coherent or incoherent.

To model ray transport through a dielectric interface described by (7.1), a full 3D ray tracing approach is suitable. We assume that the diffuser is flat enough that we can neglect self-shadowing, total internal reflection and multiple refractions. However, for weak diffusers

(on the order of 1°) and apertures below $F/2.8$, the maximum angle of incidence at the phase surface will be on the order of 11°, which is small enough to adopt the paraxial (small angle) approximation to Snell's law. While the paraxial model is not necessary for our methods to work, it provides valuable insight into the behavior of phase diffusers.

For simplicity, we describe our model with a 2D paraxial light field, $L(x, \theta)$, traveling along the optical axis of the system ($+z$ direction)[1]. In the paraxial regime, refraction at each interface becomes a linear form of Snell's law, $ni = n'i'$, where i is the incident angle and i' is the output angle, both measured relative to the interface normal; n and n' are the input and output refractive index, respectively. The diffuser surface gradient, $D_x(x) = -1/\hat{n}$, with $\hat{n}$ being the surface normal. As shown in Figure 7.2, we can write i in terms of θ and the surface gradient: $i = \theta + D_x(x)$. Similarly, the exit angle is given by $i' = \theta' + D_x(x)$. Plugging this into Snell's Law, the output ray angle is

$$\theta' = \frac{n}{n'}\theta + \left(\frac{n}{n'} - 1\right) D_x(x). \tag{7.5}$$

Since refraction changes the ray angle but not its position, the output position does not change ($x' = x$). Substituting (7.5) into the paraxial ray propagation equation, $x_0 = x + \theta' z_0$, results in the ray position, x_0, being

$$x_0 = x + z_0 \left[\frac{n}{n'}\theta + \left(\frac{n}{n'} - 1\right) D_x(x)\right], \tag{7.6}$$

where z_0 is the diffuser-sensor separation distance.

The final irradiance on the sensor is the sum of the radiance along all rays that fall within each pixel from any angle. This is equivalent to projecting the resulting light field (at the sensor) along the angle dimension before sampling.

Caustics

To illustrate how Equation (7.6) behaves, we explore the refraction and propagation of a plane wave through a diffuser described by Equation (7.1) with index n, in air ($n' = 1$). In (x, θ) space, a plane wave is represented by equal radiance at all positions and a single angle, as shown in Figure 7.3a. After applying Equation (7.5), each ray in L is displaced in angle by $(n-1)D_x(x)$. Notice that immediately after the diffuser, the irradiance is unchanged (projecting through angle will not change irradiance). However, propagation of the warped light field shears this curve (by shifting each point in x by an amount $z_0 n\theta'$ in accordance with Equation (7.6)). Because the shear is proportional to θ, the angular ripples created by the diffuser result in structure in the final projected irradiance (Figure 7.3b). Another way to visualize this is via the peaks that arise from the bunching of rays under regions where the local diffuser curvature causes it to act like a positive lens (see Figure 7.3c).

The intensity peaks induced by the combination of refraction and propagation, known as caustics, create an intensity pattern at the sensor that is directly related to the local

[1]Extending to 4D non-paraxial optics is straightforward with the ray tracing approaches used here, based on the surface normals in $\mathbb{R}^3$

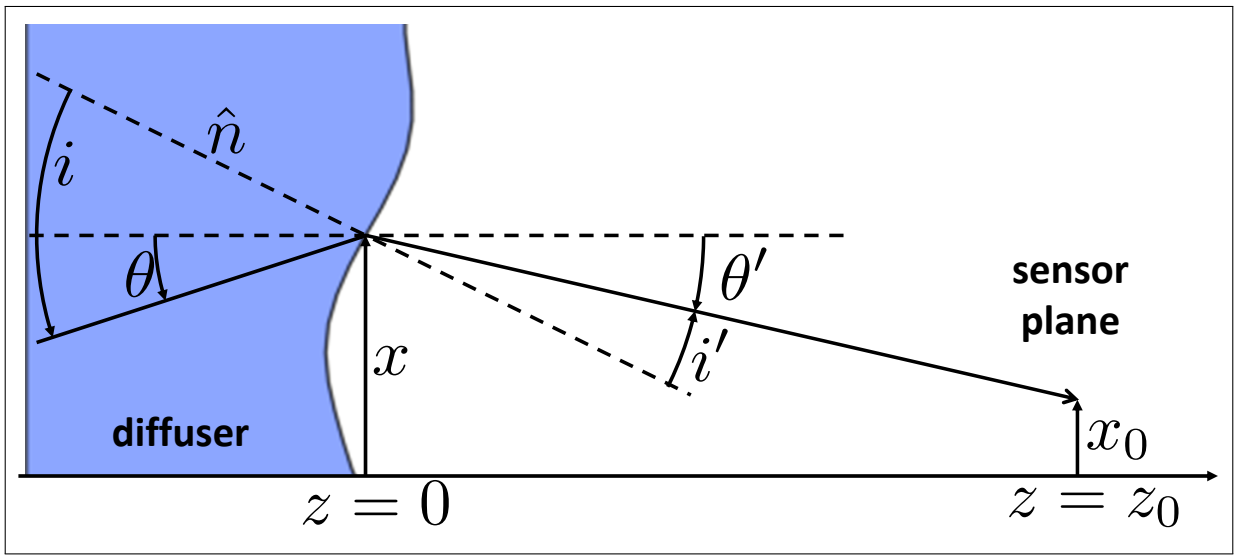


Figure 7.2: Ray geometry for a single ray hitting a diffuser surface and refracting before reaching the sensor plane.

structure of the mask surface. Intuitively, these peaks will be located under the strongly convex regions of the diffuser. Because Equation (7.5) is linear in θ, changing the incident illumination angle leads to a linear shift of the caustic pattern. Thus, the intensity pattern formed by light striking any part of the diffuser is uniquely determined by the incident angle and the local diffuser structure. This is also true for amplitude-coded masks and is the fundamental building block for the invertible linear model in Section 7.2.

For a 4D treatment of the light field, we can treat (x, θ) and (y, ϕ) independently and apply Equation (7.6) to each direction separately. This leads to 2D caustic patterns demonstrated in Figure 7.3d, which was simulated using our ray tracing model (described in Section 7.2).

Coherence

Diffusers generally produce diffractive speckle patterns that depend greatly on the coherence properties of the illumination [48]. Therefore, to design our system to be valid across a broad range of objects and illumination, we must consider the effects of diffraction and coherence.

Diffraction depends on the wavelength and distance propagated, as compared to the size scale of the object. These effects are captured by the Fresnel number, F, which provides a guideline for describing the amount of diffraction (larger F means less diffraction effects) [47]:

$$F = \frac{a^2}{z\lambda}, \tag{7.7}$$

where z is the propagation distance and a is the size of the object under consideration (for diffusers, we use $a = \sigma$). Generally, diffraction becomes important when $F < 1$, for example outside of the small defocus regime [52]. Our ray optics model should thus be accurate as

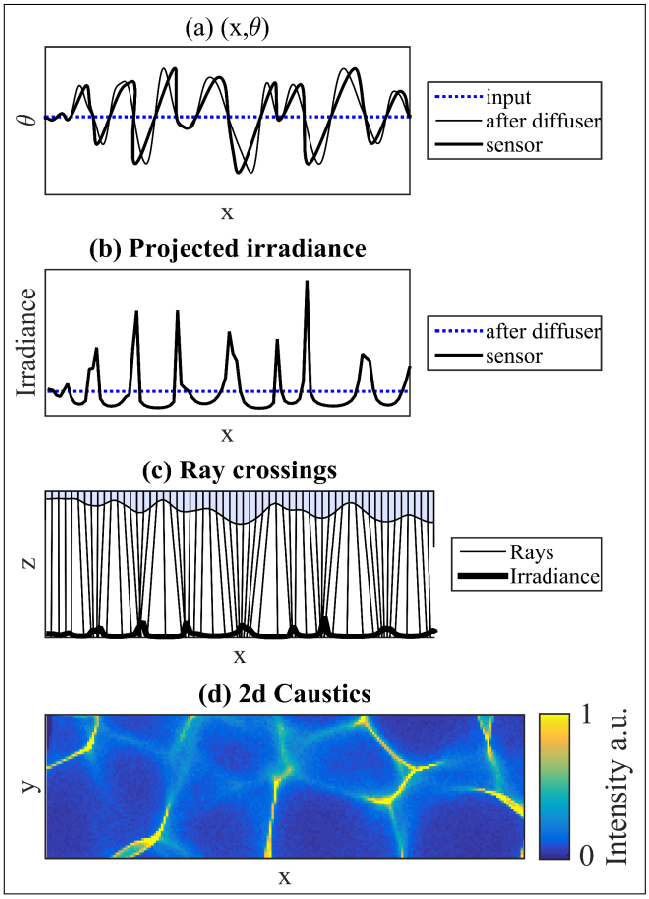


Figure 7.3: Simulation of diffuser caustics from plane wave illumination. (a) Space-angle plots for the input plane wave, post-diffuser, and sensor plane. (b) The resulting irradiance at the sensor, generated by integrating over θ. (c) Axial cross-section of rays passing through the diffuser to form caustic patterns at the sensor plane. (d) 2D caustics predicted by 4D ray tracing.

long as the diffuser-to-sensor propagation distance corresponds to a Fresnel number larger than $F = 1$.

Figure 7.4 compares the intensity evolution of a diffuser illuminated by a coherent plane wave, using both wave and ray optics models. Our wave-optics model applies Fresnel propagation to the electric field in Equation (7.4) with $\lambda = 532nm$, then computes irradiance at propagation distance z as $I_c(x, y, z) = |E(x, y; z)|^2$. Our ray optics model uses the methods discussed in Section 7.2 to compute the output positions of each ray, then calculates irradiance, I_r, by binning the rays onto the same grid used in the wave model. Figure 7.4 shows an x-z slice of the irradiance pattern generated by each method. Since wave optics captures interference and diffraction effects, discrepancies are considered errors due to the ray optics approximation.

Clearly, high-contrast caustic patterns arise before $F = 1$, though the rms error between the wave and ray models is small, indicating that the ray optics model is valid. For larger propagation distances ($F < 1$), the error is dominated by the interference fringes surrounding

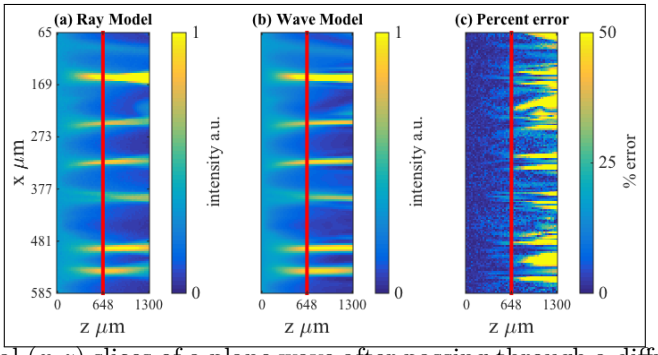


Figure 7.4: Simulated axial (x-z) slices of a plane wave after passing through a diffuser, under both our wave optics and ray optics models. The red line corresponds to a Fresnel number of $F = 1$ (at $z = 648\mu m$ for our system), which demarcates approximately the propagation distance at which the ray and wave models diverge. For smaller propagation distances, the models agree.

each caustic peak. This suggests that a good choice of propagation distance[2] is slightly less than $F = 1$, providing strong caustic patterns (thus good signal to noise), while also making the system independent of illumination coherence properties[3]. This greatly simplifies computation by allowing us to ignore lighting conditions and coherence, while still preserving the phase-coding behavior of the diffuser.

Linear Model

Because we designed the system to operate in a regime where interference is negligible, we effectively treat the object as temporally and spatially incoherent. That is, all light striking a single detector pixel adds linearly in intensity. This enables the optical system to be represented via a linear mapping from the 4D light field (radiance) before the diffuser, $L(x, y, \theta, \phi)$, to the 2D sensor irradiance, $I(x_0, y_0)$.

To derive the linear forward mapping as a matrix, $\mathbf{A}$, we consider sampled versions of both $L(x, y, \theta, \phi)$ and $I(x_0, y_0)$. We discretize $L(x, y, \theta, \phi)$ into 4D boxes, each with spatial extent Δx by Δy and angular extent $\Delta\theta$ by $\Delta\phi$. Figure 7.5 shows a 2D example with N spatial samples and P angular samples at each position.

To construct $\mathbf{A}$, consider that the first column is mapped by multiplying with a column vector of zeros and a 1 in the first element. Physically, this corresponds to a bundle of light rays striking the diffuser across an area Δx centered at x_n from angles spanning $\Delta\theta$ centered at θ_p, then propagating to the sensor. Therefore, column $j = Pn+p$ of $\mathbf{A}$ is the sensor image due to uniform illumination at x between $x_n - \frac{\Delta x}{2}$ and $x_n + \frac{\Delta x}{2}$, and θ between $\theta_p - \frac{\Delta\theta}{2}$ and $\theta_p + \frac{\Delta\theta}{2}$. Each entry of $\mathbf{A}$, $a_{i,j}$, is the fraction of light that strikes pixel i from the light field point indexed by j. This is equivalent to the fractional area of each sensor pixel in (x, θ) space that falls within box j, as shown in Figure 7.5(a).

[2]This refers to the propagation distance from the diffuser to the sensor and is independent of amount of depth present in the original scene.

[3]Note that this metric does not account for phase height, which may lead to larger discrepancies for strong phase objects.

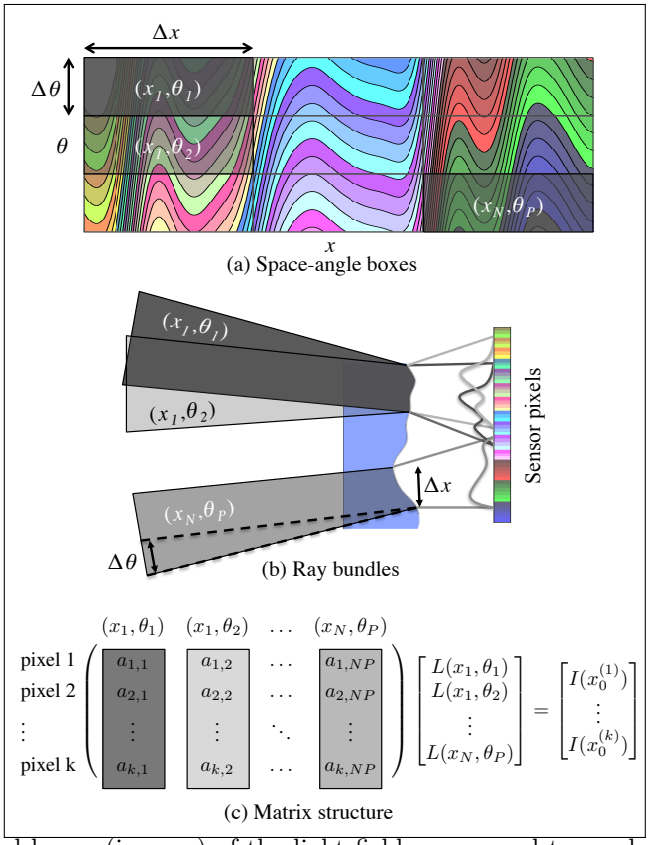


Figure 7.5: (a) Finite-sized boxes (in grey) of the light field correspond to ray bundles hitting the diffuser. The structure of each sensor pixel in (x, θ) takes on the shape of the sheared diffuser gradient. Here, each band of color corresponds to all the (x, θ) pairs that strike a single sensor pixel. A bundle will span multiple pixels in (x, θ) space. (b) Each ray bundle creates a unique caustic pattern on the sensor, which shifts according to the input angle. The set of sensor pixels illuminated matches those within each bundle's box in (a). (c) The corresponding matrix structure for a light field consisting of N spatial samples with P angular samples at each x. $I \in \mathbb{R}^k$ and L is a 1D vector $L \in \mathbb{R}^{NP}$.

Only the 2D phase plate shape and refractive index are needed for computing the entire $\mathbf{A}$ matrix. Extending to 4D implies a convenient method for generating $\mathbf{A}$: for the j^{th} column, we generate many rays randomly distributed across light field bundle j, then compute their output positions using (7.6). Finally, we bin rays into sensor pixels, then column-stack the resulting image as column j of $\mathbf{A}$. Extending this method to multiple colors is done by repeating the above procedure at different wavelengths, accounting for the diffuser material dispersion curve.

Inverse Problem

The ultimate goal of the linear forward model is to recover the 4D light field from a single sensor measurement by solving the inverse problem. Here, we explore the properties of $\mathbf{A}$ that enable stable inversion and how these numerical requirements map to the physical system. Equation (7.6) states that output ray position is linearly related to the input angle

and the diffuser gradient. Bundles sharing common (x, y) coordinates will exhibit the same caustic structure for all incident angles; the only difference will be a lateral shift at the sensor due to their relative input angle differences. This shifting behavior is visualized in Figure 7.5b.

In the matrix, this shift behavior imparts a block circulant structure where columns belonging to a common (x, y) are shifted copies of each other. The critical implication of this is that it is possible for two different light field regions to strike the exact same set of output pixels, while remaining distinguishable. Hence, the diffuser-encoded light field recording process can be thought of as a multiplexing approach, since each pixel measures a linear combination of points in light field space. This is equivalent to two columns of $\mathbf{A}$ having nonzero elements in identical rows. If these rows have proportional *values* at each nonzero pixel, the rows are linearly dependent and the problem is ill-posed.

In order to prevent this issue, the phase mask must have sufficient structure within each bundle so that it creates distinct values in each row. This is the key to why a diffuser works: its random surface has extremely low probability of repeating a pattern. However, this also implies that we are not free to make the light field sampling arbitrarily dense in the lateral direction. In other words, we must choose Δx and Δy to be sufficiently large compared to σ so that distinct caustics are present within each bundle at the sensor. In practice, setting Δx and Δy to be at least σ leads to reasonable inversion behavior. Additionally, the angular sampling is limited by the sensor pixel pitch, since the change in angle between two consecutive bundles must cause a shift at the sensor that is large enough to be sampled correctly.

Once a well-conditioned $\mathbf{A}$ matrix has been constructed, we recover the 4D light field, L_{rec}, from a 2D sensor image by solving the following least squares inverse problem:

$$L_{rec} = \arg\min_{L} \|\mathbf{A}L - I\|_2^2 + \tau\psi(L), \tag{7.8}$$

where $\psi(L)$ is a regularization function and τ is a scalar regularization parameter. As a baseline, we solve the ℓ_2 regularized problem using $\psi(L) = \|L\|_2^2$. In the experimental section, we also explore the use of two nonlinear regularizers for exploiting sparsity: 3D Total Variation (3DTV) regularization from Tian et al. [129] and ℓ_1 regularization in the 2D wavelet domain, similar to Veeraraghavan et al. [133]:

$$\alpha_{rec} = \arg\min_{\alpha} \|\mathbf{A}W^{-1}\alpha - I\|_2^2 + \tau\|\alpha\|_1, \tag{7.9}$$

where α is a vector of the wavelet coefficients for each sub-aperture image, and W^{-1} is the inverse 2D wavelet transform operator. It follows that $L_{rec} = W^{-1}\alpha_{rec}$.

7.3 Implementation

To implement our approach in practice, two key pieces are needed. One is the imaging system itself, which is described in Figure 7.1. The other is a diffuser height map; we must either use a known surface shape, or measure it. We propose here a phase-from-focus method

for measuring the diffuser surface shape *in situ*. The height field for ray tracing can be computed from the phase map using Equation 7.3. From this one-time calibration, we can then computationally reconstruct light fields according to Section 7.2.

Experimental Setup

In our experiments, we use a 1 degree Light Shaping Diffuser (Luminit, LLC). Because the diffuser-to-sensor propagation distance needed is short (approximately $650\mu m$) and our physical sensor's packaging does not allow the diffuser to be placed so close, we add a $4f$ relay system with $1.33\times$ magnification to image the diffuser onto the sensor. This introduces unwanted aberrations in the diffuser wavefront; however, our TIE measurement incorporates these into the result, thereby mitigating their effects on our final images. To record images, we use an imaging lens with f=125mm stopped down to f/16. The total field of view is approximately 25 mm laterally. Our PCO Edge 5.5 sCMOS monochrome camera has 5 Megapixels with pixel pitch 6.5 μm. The image of the diffuser is placed $648\mu m$, corresponding approximately to $F = 1$, in front of the sensor.

Phase imaging for diffuser calibration

Since phase is linearly related to surface profile and can be measured with sub-wavelength accuracy, phase retrieval methods are a practical means for calibrating the diffuser surface *in situ*. In particular, our method uses only a few images taken at different focus positions [62], which is easy to implement in our existing system by translating the camera between images. The TIE describes how intensity evolves axially with respect to phase [127]

$$\frac{\partial I(x,y)}{\partial z} = -\frac{1}{k}\nabla_{\perp}\bullet[I(x,y)\nabla_{\perp}\phi(x,y)], \tag{7.10}$$

where $\nabla_{\perp}$ is the gradient operator in the lateral (x, y) dimensions only and $k = 2\pi/\lambda$ is the wave vector magnitude. Using this equation, a few images taken with small defocus can be used to solve for phase. The algorithm we use is a GP-TIE solver [62] which is offered open-source on Laura Waller's Computational Imaging Lab website.[4]

Experimentally, we calibrate the system using a coherent collimated plane wave from a 532 nm laser diode. The camera is mounted on a micrometer axial translation stage, which we use to take a focus stack of 100 images with z step size of 25.4 μm. In fact, only 5 through-focus images are necessary for a good phase result which correctly predicts the other intensity images; however, we use the full stack to ensure robustness. A few raw images are shown in Figure 7.6, along with the recovered phase map. Notice that the diffuser becomes invisible at focus, where it is a pure phase delay that does not change intensity. The phase profile recovers an average angle of 1.4 deg after magnification, and $\sigma = 18\mu m$, with rms surface height of $1.15\mu m$. We observe good agreement between the measured caustics within the Fresnel range and the caustics predicted using our matrix.

[4] http://www.laurawaller.com/opensource/

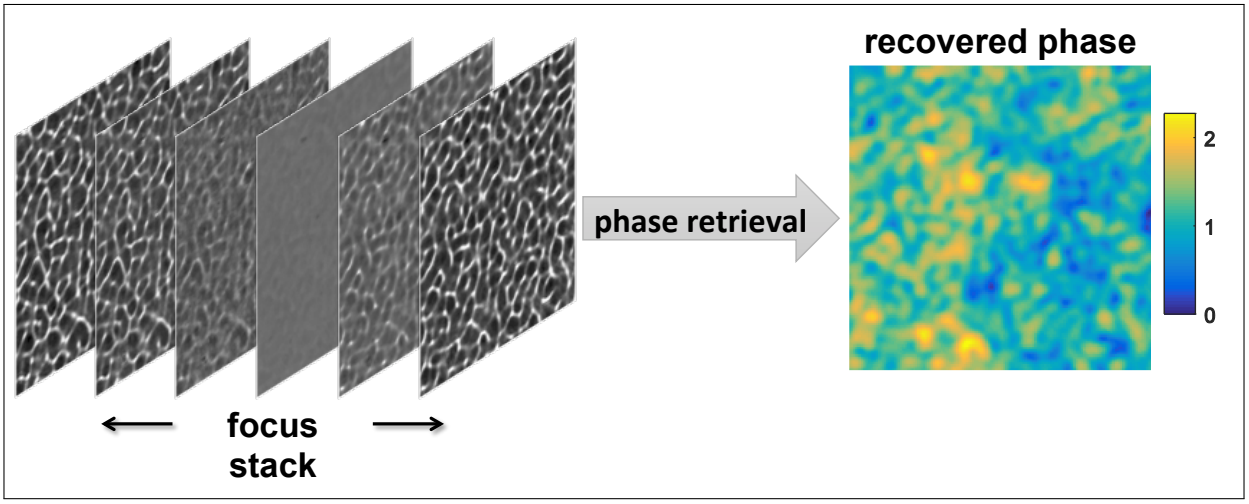


Figure 7.6: A stack of irradiance images collected at different focus positions in our experimental setup are used to recover the phase map of the diffuser surface, which directly relates to height.

Simulations

In order to account for the fact that light field radiance values may change over the span of a single bundle in the inversion matrix, we use a high spatial resolution forward matrix to simulate the sensor data and a lower resolution one for inversion. Our synthetic input light field, $L(x, y, \theta, \phi)$, is rendered using POV-Ray with 5×5 angular samples and 512×512 spatial samples using scene files shared from [133]. We then use the ray tracing approach described in Section 7.2 in conjunction with the experimentally measured diffuser data from Section 7.3 to create a forward matrix, $\mathbf{A}_f$, that projects $L(x, y, \theta, \phi)$ onto a 1024×1024 pixel sensor. We simulate the sensor data as the matrix-vector product $\mathbf{A}_f L$, then add 5% Gaussian noise. To invert the problem, we trace a second lower resolution matrix, $\mathbf{A}$, that projects back to 128×128 spatial samples and 5×5 angular samples in light field space. We then use $\mathbf{A}$ to solve for L.

Inverse Problem

The inverse problem is solved by the gradient descent solver LSMR [43] for the ℓ_2 problem and Two Step Iterative Shrinkage/Thresholding (TwIST) for 3DTV and ℓ_1 [17] to solve equations (7.8) and (7.9). We choose the angular resolution such that each successive caustic pattern shifts by at least 1 pixel compared to the neighboring bundles. The spatial sampling is determined heuristically, but we find it must be at least σ to obtain good results.

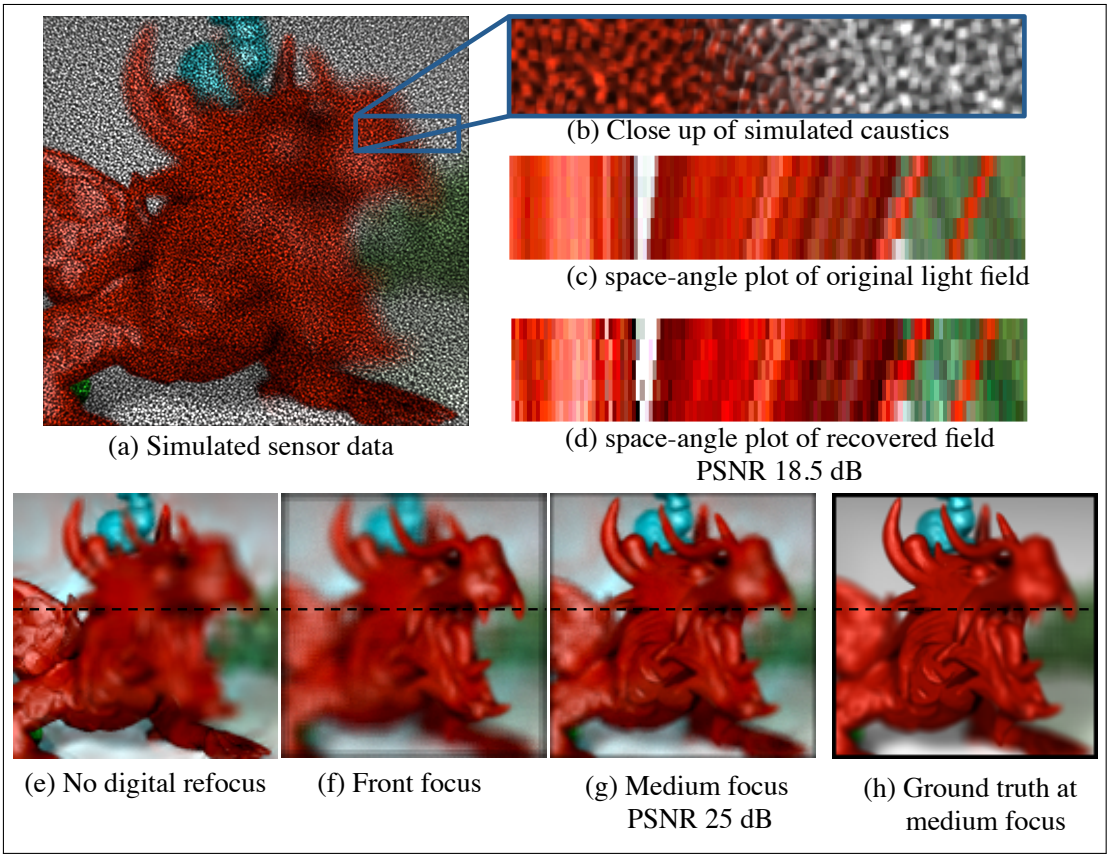


Figure 7.7: (a) Simulated sensor data with a zoom-in to show caustics shown in (b). We achieve good qualitative agreement between our simulated caustics and those shown in figure 7.8(b). (c) An (x, θ) plot from the original light field along the black line in (e)-(h), with 5% Gaussian noise added. (d) Image at same (x, θ) from our recovered light field. We are able to recover full parallax and occlusion effects. (e)-(g) Reconstructed synthetic-focus images generated from recovered light field. (e) No digital refocus, (f) refocused at the front plane, and (g) refocused on the blue bunny in the mid-focus. (h) Ground image of original light field refocused to same plane as (g).

7.4 Results

Simulation Results

The rendered light field is represented in RGB form, so we solve each color independently using the 2D wavelets with ℓ_1 regularizer. Figure 7.7 shows the simulated sensor data as well as the caustic patterns and the original and reconstructed (x, θ) plots along one line in the image. Figure 7.7 (e) shows the irradiance detected at the diffuser, computed by summing the recovered light field over θ and ϕ. We use the shift-and-add technique [98] to digitally refocus the reconstructed light field to several planes and compare with the original refocused light field. We achieve good agreement between the original light field and our reconstruction, with PSNR of 25 dB in the synthetic focus images, and 18.5 dB in the (x, θ) images.

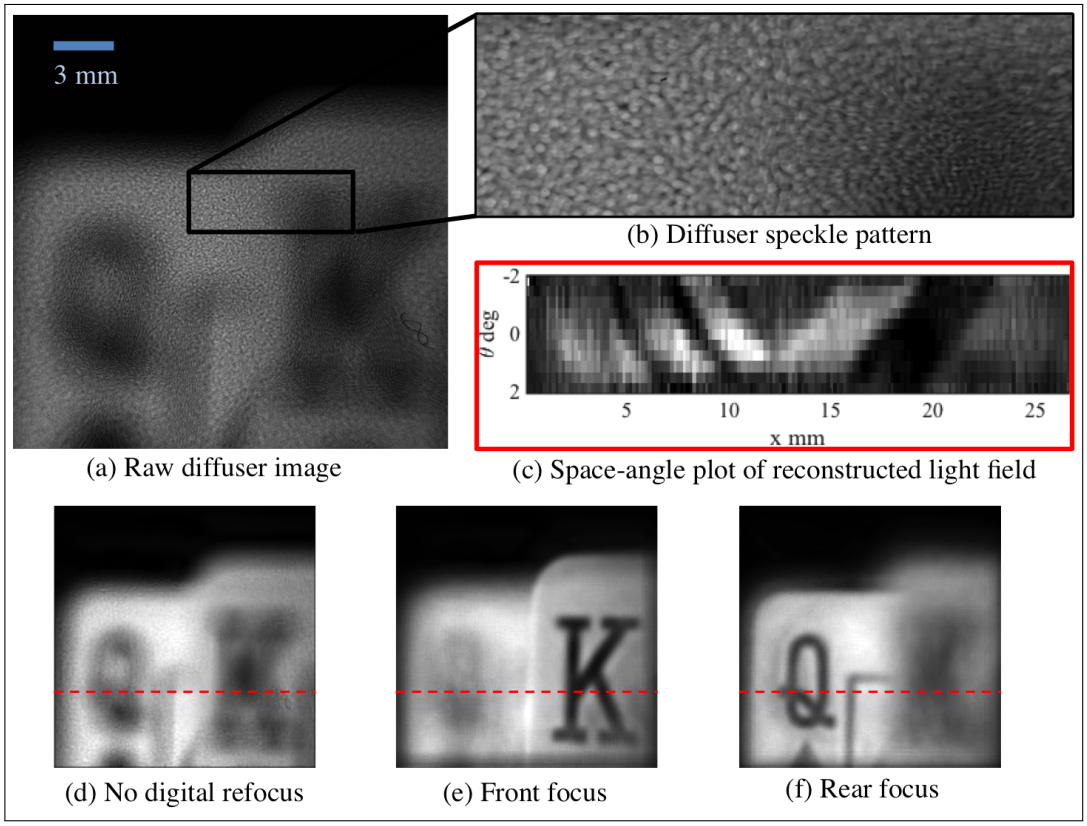


Figure 7.8: Experimental light field reconstruction from two playing cards using wavelet denoising. (a) Raw data, (b) close-up of diffuser caustics. (c) An x-θ plot along the red line in (d)-(f). Notice that the parallax due to the depth differences manifests as strong angular variations, and we also observe occlusion effects in the center. (d) Shows the reconstructed light field projected to $z = 0$ (no refocusing). (e) and (f) are the digitally refocused images at $+40$ mm and -40 mm, respectively.

Experimental Images

With the calibration complete, individual diffuser-blurred images are recorded through the imaging path of the system. We restrict the primary lens aperture to $f/16$ solely for the purpose of controlling aberrations in the $4f$ system. We solve the inverse problem using TwIST with the 3DTV regularizer or with an ℓ_1 regularizer on in the 2D wavelet coefficients of each sub-aperture image. We are able to reconstruct a light field with 11×11 angular samples in each direction and 170×170 lateral samples from a 2048×2048 sensor measurement, and demonstrate a large refocus distance. Figure 7.8 shows experimental results for a pair of playing cards placed $+40$ mm and -40 mm from the native focal plane, using 3DTV regularization. In the (x, θ) plot, strong angular variation is visible, including occlusion effects. Figure 7.9 shows another experimental set of images from a ruler tilted at an angle to the optical axis, using wavelet regularization.

We find that the ℓ_2 regularizer performs poorly on real-world data, destroying angular structure before it brings noise under control. 3DTV is extremely good at suppressing noise, but imparts a distinctive piecewise-constant look to the sub-aperture images that is only

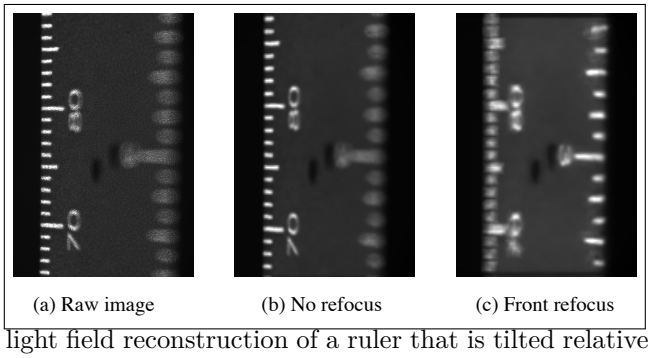


Figure 7.9: Experimental light field reconstruction of a ruler that is tilted relative to optical axis by approximately 30 degrees. (a) Raw data, (b) no refocus, and (c) focused to -20 mm.

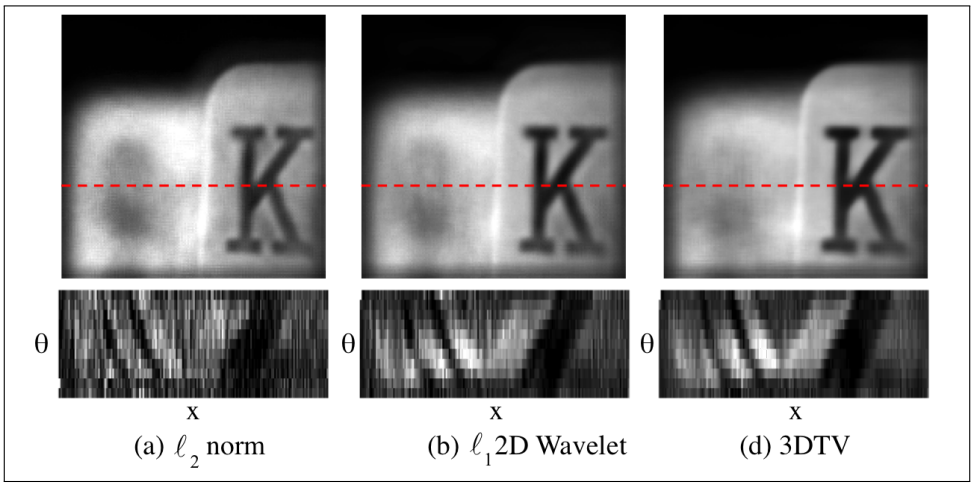


Figure 7.10: The effects of different regularizers on experimental reconstructions. (Top row) The reconstructed light fields refocused at the front plane. (Bottom row) (x, θ) plots along the red line. (a) ℓ_2 regularization suffers from noise artifacts, and increasing τ destroys angular information before adequately reducing noise. (b) ℓ_1 regularized 2D Wavelets is able to reduce noise significantly without destroying angular information. (c) 3DTV qualitatively performs the best in this case, due to the piecewise constant nature of this object.

suitable for piecewise-constant scenes. For natural objects, we find the most robust approach to be ℓ_1 regularized 2D wavelets. Figure 7.10 shows an example of all three regularizers applied to the playing card image from Figure 7.8.

7.5 Limitations

While the thin phase plate approach provides high light throughput, we find that significant noise is present in the sub-aperture images, which we attribute to ray error due to aberrations in the $4f$ relay lenses. Although the TIE phase measurement helps overcome this by measuring the aberrated diffuser phase for monochromatic on-axis illumination, it does not compensate for off-axis or chromatic aberrations induced by the relay optics. This severely limits the F-numbers and wavelengths we are able to use, but is not a fundamental limitation

of our approach. To overcome this, we plan to place the diffuser directly in front of the sensor in the future. This will dramatically improve consistency of our phase measurement for different illumination angles, and will enable us to model polychromatic illumination using the diffuser's material dispersion curve. This will also have the added benefit of making the system more compact.

Because we solve for the radiance within an entire ray bundle of $\mathbf{A}$, we are unable to resolve variations in the light field that happen across spatial scales smaller than each bundle. We observe, empirically, that this leads to artifacts in the recovered sub-aperture images at very strong edges.

Lastly, our approach requires significant computational time as compared to microlens systems. Creating an entire matrix requires roughly 20 billion rays and takes 20-60 minutes to create the calibration matrix. However, once it is computed, solving the inverse problem takes 1-2 minutes.

7.6 Future Work

Because the angular sampling is determined by the diffuser-sensor distance, we believe it may be possible to adjust the focus distance to compensate for changes in the main lens F-number. The matrices for various focus distances could be precomputed, enabling F-number flexibility in a way that lenslets cannot accomplish.

In our prototype system, the spatial and angular sampling has been determined heuristically. The impact of the discretization in the matrix representation is still an open problem the warrants future work.

We also believe it would be possible to build a camera that operates lens-free, provided the diffuser characteristics are appropriately chosen. Finally, the benefits of compressed sensing are not born out in this system, largely due to the compactness of the spatial footprint of each ray bundle on the diffuser. It would be interesting to explore compressive sampling approaches in snapshot light field capture [133, 84], though it is difficult to compete with direct sampling approaches using regular microlenses (after all, a condition number of 1 is pretty good!).

Chapter 8

Appendix

8.1 A not-so-brief comment on vector notation

Throughout this work I will define $\mathbf{v}$ as a vector. That is, it follows the rules of a vector space: for $\mathbf{w}$, $\mathbf{u}$, and $\mathbf{u}$ in vector space V over field F, the following definitions hold:

- element-wise addition $+ : V \times V \rightarrow V$. In English, $+$ takes vectors $\mathbf{u}$ and $\mathbf{v}$ and produces $\mathbf{w}$, where each element of $\mathbf{w}$ is the sum of corresponding elements in $\mathbf{u}$ and $\mathbf{v}$. Note $\mathbf{w}$ must remain in V
- Scalar multiplication $\cdot : F \times V \rightarrow V$. In English, for scalar $a \in F$ and vector $\mathbf{v} \in V$, $a \cdot \mathbf{v} = a\mathbf{v}$, where each entry in $\mathbf{v}$ is multiplied by a. Again, the result, $a\mathbf{v}$ must remain in V.

This can be confusing because multiple arguments will be used to index $\mathbf{v}$, while conventionally, vectors are specified by a single index (e.g. a column vector). Note, however, that these definitions for $+$ and $\cdot$ do not require the vectors to have a particular shape. For instance, the notation $\mathbf{v}[x, y]$ is used in 2D imaging to denote that two spatial variables are needed to describe the physical quantity of interest. It is worth pointing out a few technicalities to avoid confusion within this convention. First, for an $M \times N$ grid, $\mathbf{v}$ would constitute a vector of dimensionality MN, not 2. However, commonly, such an image would be referred to as 2-dimensional (or 2D), despite having dimensionality MN in the sense of vector spaces. For this dissertation, the term *dimension* will be reserved to speaking about the number of degrees of freedom needed to accurately describe either the continuous image $\boldsymbol{v}$ or the discrete image $\mathbf{v}$. Second, while it is convenient to index $\mathbf{v}[x, y]$ with two arguments, this can equivalently be accomplished with a single index that is a function of x and y: $\mathbf{v}[x, y] = \mathbf{v}[n]$ where $n = Mx + y$. Note that all other axioms of vector spaces hold as well **cite**.

Finally, we will assume that the optical system performs a linear transformation of $\mathbf{v}$, mapping from the object space to sensor space. This can be represented as linear map $A : V \rightarrow W$, where V is the vector space of possible objects, and W is the vector space of sensor exposures. When working with discrete inputs and outputs, as will be the convention

from here on, A can be thought of as a matrix, $\mathbf{A}$. However, the convention of matrices being specified by two indices (row and column) becomes again confusing when recording multi-dimensional inputs. To deal with this, we use the same trick as above, realizing that both the input and output spaces can be indexed with multiple variables for convenience, but can always be equivalently represented using a single index in each space. Hence, denoting linear maps between two multi-dimensional spaces using the matrix-vector product, $\mathbf{b} = \mathbf{A}\mathbf{v}$, does not require us to be considering fundamentally 1D signals. As a final note, the matrix $\mathbf{A}$ will frequently be computed by composing linear operators that operate along independent dimensions of the input. For example, a 2D Discrete Fourier Transform (DFT) that applies the 2D DFT to $\mathbf{v}[x, y]$ along the x- and y-directions can equivalently be written as a matrix $\mathbf{F}$. In practice, it is more efficient to operate on multi-dimensional arrays, so this matrix notation will be reserved for algebra and compact notation, with multi-dimensional linear operators being preferred during implementation. In summary, do not be confused by the convention that vectors have one index and matrices have two. It is perfectly fine to index the row and column spaces of a discrete linear system with more than one index each.

8.2 Build-your-own diffusercam

This section written by Camille Biscarrat and Shreyas Parthasarathy under the supervision of Nick Antipa, Grace Kuo, Laura Waller.

8.3 Introduction

This guide is meant as a tutorial for the lensless image reconstruction algorithms used in DiffuserCam. It provides a brief overview of the optics involved and how it was used to develop the most current version. See our other document ("How to build a (Pi) DiffuserCam") for information on how to actually build and calibrate DiffuserCam.

Why Diffusers?

For most 2D imaging applications, lens-based systems have been optimized in design and fabrication to be the best option. However, *lensless* imaging systems have not been investigated nearly as much. DiffuserCam is a lensless system that replaces the lens element with a diffuser (a thin, transparent, lightly scattering material). See Figure 8.1 below.

Possible advantages include:

- *Lensless systems are lightweight.* Most of the weight and size of imaging systems comes from the physical constraints of lens design. Substituting lenses for a thin, flat material can allow for smaller, lighter imaging systems.
- *Diffusers require less precise fabrication.* We demonstrated that DiffuserCams (of varying quality) can be created by household scatterers such as Scotch tape. Since the

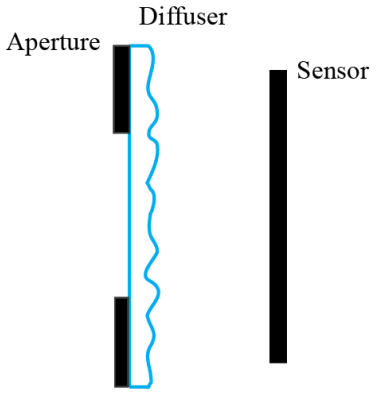


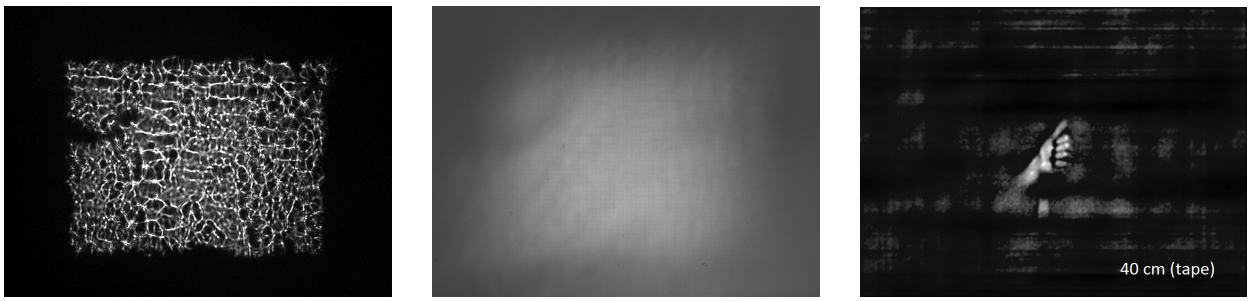
Figure 8.1: Cartoon schematic of DiffuserCam

structure of a diffuser is naturally random, you can create a DiffuserCam yourself without access to precise fabrication tools.

- *Possibility of 3D imaging/microscopy.* We've also shown that lensless cameras can capture 3D images and are robust to missing or dead pixels (see this paper), both of which are promising in the field of microscopy.

DiffuserCam

Every diffuser has a "focal plane". Instead of mapping a faraway point source to a point in this plane (as lenses do), the diffuser maps a point source to a "caustic pattern" (see Fig. 8.2a) over the entire plane. So, replacing the lens in a camera with a diffuser of the same focal length creates a system that maps points in the scene to many points on the sensor (see Fig. 8.2b)



(a) Caustic image of a single point source

(b) Sensor reading of a hand

(c) Reconstructed image of a hand

Figure 8.2: The 3 important steps in DiffuserCam's operation.

The key to DiffuserCam's operation is that, while light information is spread out over the sensor, none of that information is lost. You can see in Fig. 8.2b that the sensor reading

won't look like the object. However, we can recover the object image using a reconstruction algorithm that requires a single calibration measurement of the caustic produced by a point source. This measurement, called a point spread function (PSF), completely characterizes the scattering behavior of the diffuser (under certain assumptions).

Imaging Systems

To derive the algorithm and understand where these assumptions come from, it's helpful to think of the imaging system as a function that maps objects in the real world to images on the sensor. More precisely, it is a function f that maps a 2D array $\mathbf{v}$ of light intensity values (the scene) to a 2D array of pixel values $\mathbf{b}$ on the sensor. Recovering the scene $\mathbf{v}$ from a sensor reading $\mathbf{b}$ is equivalent to inverting this function (though sometimes the function isn't invertible):

$$f(\mathbf{v}) = \mathbf{b} \implies \mathbf{v} = f^{-1}(\mathbf{b})$$

First, we need to describe f mathematically. In computational imaging, characterizing f (usually through a theoretical model of the optics involved) is known as constructing a "forward model," and inverting it efficiently is known as the corresponding "inverse problem." This tutorial covers DiffuserCam algorithms in roughly that order. Note that f is not always invertible, but that is usually because many v's can map to the same b. So, we often introduce *priors*, or assumptions that constrain the possible v's in order to construct an estimate for the scene.

8.4 Problem Specification

Forward Model

Roughly speaking, f is the composition of everything that happens to light as it travels from the object scene to the sensor. Each ray from a point in the scene propagates a certain distance to the diffuser and is locally refracted by the diffuser surface, then propagated again to the sensor plane. Whether or not the ray hits the sensor depends on how it was bent – we will start by ignoring this issue and addressing the finite sensor size after constructing the rest of the model.

We make the following approximations:

- *Shift invariance:* A lateral shift of the point source causes a lateral translation of the sensor reading.
- *Linearity:* Scaling the intensity of a point source corresponds to scaling the intensity of the sensor reading by the same amount. Also, the pattern due to two point sources is the sum of their individual contributions. These two assumptions amount to having *incoherent* light sources and a sensor that responds to light intensity linearly. Both of these conditions are often satisfied.

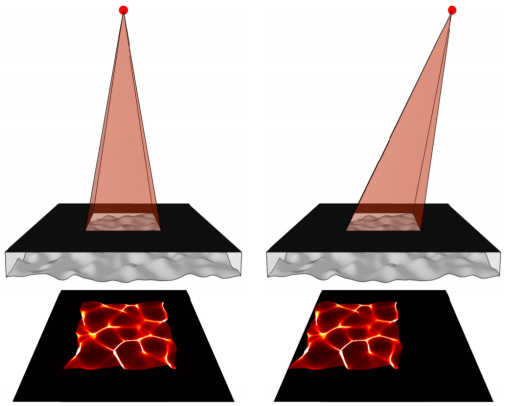
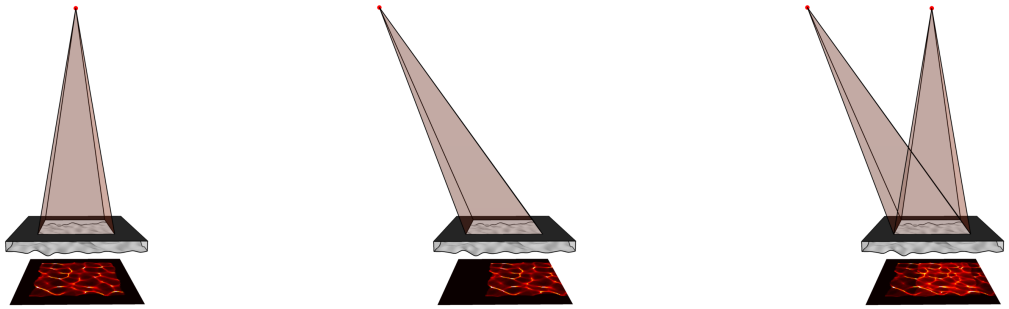


Figure 8.3: As the point source shifts to the right, the image on the sensor shifts to the left



(a) Point source on axis

(b) Point source off axis

(c) Superposition of both point sources

Figure 8.4: Each point source creates a pattern on the sensor. When two point sources are present, the sensor reads the superposition of the patterns created by each individual point source.

In short, the diffuser system is assumed to be linear shift-invariant (LSI). We assume that $\mathbf{v}$ can be represented as the sum of many point sources of varying intensity and position. By the LSI property of the system, the output $f(\mathbf{v})$ corresponding to the input $\mathbf{v}$ can be represented as a 2D convolution with a single PSF $\mathbf{h}$:

$$f(\mathbf{v}) = \mathbf{h} * \mathbf{v}$$

Since f is linear, it is conceptually helpful to think of it as a matrix. However, matrices operate on vectors, not 2D images like $\mathbf{v}$ and $\mathbf{b}$. We can get around this by *vectorizing* the images – creating a vector that contains the same information as the image by stacking all of the columns on top of each other. Thus our mathematical model can consistently treat these images as 1-dimensional vectors. For example, an $m \times n$ sensor reading would now be an mn-length vector. This trick allows us to represent our convolution as a 2D matrix $\mathbf{H}$ where $\mathbf{h} * \mathbf{v} \iff \mathbf{H}\mathbf{v}$. For all the following derivations, we will reserve lowercase letters

for images, and bolded lowercase letters for the corresponding vectorized images. Function notation (with parentheses or braces denoting arguments) will be used to denote linear operators, and bolded uppercase letters will be used to denote the matrix representations of these operators.

Now that we've constructed a model for how the light propagates to the sensor plane, we need to account for the sensor's finite size. While all of the light rays hit the sensor plane, not all of them hit the physical sensor. So while the output of the diffuser system is a convolution, only part of that convolution is recorded on the sensor. In other words, the 2D sensor reading is a cropped convolution: $f(\mathbf{v}) = \text{crop}(\mathbf{h} * \mathbf{v})$. The equivalent vectorized formulation is

$$\begin{aligned} \text{crop}(\mathbf{h} * \mathbf{v}) &\iff \mathbf{CHv} \\ f(\mathbf{v}) &\iff \mathbf{Av} \end{aligned}$$

where $\mathbf{C}$ is a matrix representation of cropping. We use $\mathbf{A}$ as shorthand for $\mathbf{CH}$. This equation serves as our forward model.

Inverse Problem

A first approach to solving for $\mathbf{v}$, which ignores the crop, would be to try Wiener deconvolution. This method is a common way to reverse convolution, but it relies on diagonalizing the measurement matrix, and cannot model the cropping behavior at all (see our ADMM Jupyter notebook for explanation of diagonalization). While Wiener deconvolution would work if $\mathbf{A}$ were convolutional, i.e. $\mathbf{A} = \mathbf{H}$, adding in the crop makes $\mathbf{A}$ too complex to invert analytically.

Instead, we must find an efficient numerical way to "invert" f. In general, f isn't invertible at all: multiple $\mathbf{v}$'s can be mapped to the same $\mathbf{b}$. We can see $\mathbf{A}$ isn't invertible for two reasons:

- Information is lost in the crop operation, so $\mathbf{C}$ is not an invertible matrix.
- Convolving with a fixed function, e.g. $\mathbf{h}$, is not always invertible, so $\mathbf{H}$ is not necessarily invertible.

The typical approach to solving $\mathbf{Av} = \mathbf{b}$ for non-invertible $\mathbf{A}$ is to formulate it as an optimization problem, which has the same form regardless of whether $\mathbf{A}$ is convolutional or not:

$$\mathbf{v}^* = \underset{\mathbf{v}}{\text{argmin}} \frac{1}{2}\|\mathbf{Av} - \mathbf{b}\|_2^2$$

When $\mathbf{v} = \mathbf{v}^*$, $\mathbf{Av}^* = \mathbf{b}$ and the objective function $\mathbf{Av} - \mathbf{b}$ is minimized.

It is worth noting that $\mathbf{A}$ is extremely large, and scales with the area of the sensor. Our sensor has $\sim 10^6$ pixels, so $\mathbf{A}$ would have on the order of $10^6 \times 10^6 = 10^{12}$ entries. While $\mathbf{A}$ is useful mathematically, it's computationally useless to ever load/store it in memory. Whichever algorithm we choose to solve the minimization problem has to avoid ever loading

A in memory. Our general approach to addressing this issue will be to make sure the algorithm can be implemented in terms of the linear operators that make up f: crop and convolution. Both of these operations have fast implementations on 2D images that don't require loading their corresponding matrices.

8.5 Solving for v

Gradient Descent

Gradient descent is an iterative algorithm that finds the minimum of a convex function by following the slope "downhill" until it reaches a minimum. To solve the minimization problem

$$\text{minimize } g(\mathbf{x}),$$

we find the gradient of g wrt $\mathbf{x}$, $\nabla_{\mathbf{x}} g$, and use the property that the gradient always points in the direction of steepest *ascent*. In order to minimize g, we go the other direction:

$$\begin{aligned} \mathbf{x}_0 &= \text{ initial guess} \\ \mathbf{x}_{k+1} &\leftarrow \mathbf{x}_k - \alpha_k \nabla g(\mathbf{x}_k), \end{aligned}$$

where α is a step size that determines how far in the descent direction we go at each iteration.

Applied to our problem:

$$\begin{aligned} g(\mathbf{v}) &= \frac{1}{2}\|\mathbf{A}\mathbf{v} - \mathbf{b}\|_2^2 \\ \nabla_{\mathbf{v}} g(\mathbf{v}) &= \mathbf{A}^H(\mathbf{A}\mathbf{v} - \mathbf{b}), \end{aligned}$$

where $\mathbf{A}^H$ is the adjoint of $\mathbf{A}$. Again, we want to write $\mathbf{A}$ as a composition of linear operators that are easy to implement, so we never have to deal with $\mathbf{A}$ itself. For a product of arbitrary linear matrices $\mathbf{FG}$, the adjoint is $(\mathbf{FG})^H = \mathbf{G}^H\mathbf{F}^H$. In our case:

$$\begin{aligned} \mathbf{A}\mathbf{v} &= \mathbf{C}\mathbf{H}\mathbf{v} \\ \mathbf{A}^H\mathbf{v} &= \mathbf{H}^H\mathbf{C}^H\mathbf{v} \end{aligned}$$

We've reduced the problem of finding the adjoint of $\mathbf{A}$ to finding the adjoints of $\mathbf{H}$ and $\mathbf{C}$.

Finding the adjoint of $\mathbf{H}$*:* The adjoint of $\mathbf{H}$, a convolution, can be found by writing the operation using Fourier transforms. The convolution theorem states:

$$\mathbf{H}\mathbf{v} \iff \mathbf{h} * \mathbf{v} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{h}) \cdot \mathcal{F}(\mathbf{v})),$$

where the $\cdot$ denotes pointwise multiplication, and $\mathcal{F}$ denotes the 2D Fourier transform operator. This theorem is also known as "convolution of two signals in real space is multiplication in Fourier space." Next, we vectorize the previous statement by recognizing that 2D Fourier transforms are linear operators, so we have the equivalence $\mathcal{F}(\mathbf{v}) \iff \mathbf{F}\mathbf{v}$. To fully write

H as a product of matrices, we must also convert the pointwise multiplication to a matrix multiplication:

$$\mathcal{F}(\mathbf{h}) \cdot \mathcal{F}(\mathbf{v}) \iff \text{diag}(\mathbf{Fh})\ \mathbf{Fv}.$$

Also, $\mathbf{F}^H = \mathbf{F}^{-1}$ by "unitarity" of the Fourier transform. Finally, the adjoint of a diagonal matrix is formed by taking the complex conjugate of its entries.

In summary,

$$\begin{aligned}\mathbf{H}^H\mathbf{v} &= \left(\mathbf{F}^{-1}\ \ \text{diag}(\mathbf{Fh})\ \ \mathbf{F}\right)^H \mathbf{v} \\ &= \left(\mathbf{F}^H\ \ \text{diag}(\mathbf{Fh})^H\ \ (\mathbf{F}^{-1})^H\right)\mathbf{v} \\ &= \mathbf{F}^H\ \text{diag}(\mathbf{Fh})^*\ \mathbf{F}(\mathbf{v}),\end{aligned}$$

where * denotes complex conjugation.

Finding the adjoint of $\mathbf{C}$*:* Finally, we note that the adjoint of cropping, $\mathbf{C}^H$, is zero-padding (see section 2.4 the appendix)

Plugging in to the formula for $\mathbf{A}^H$, we find

$$\begin{cases}\mathbf{A} = \mathbf{C}\mathbf{F}^{-1}\ \text{diag}(\mathbf{Fh})\ \mathbf{F} \\ \mathbf{A}^H = \mathbf{F}^{-1}\ \text{diag}(\mathbf{Fh})^*\ \mathbf{F}\mathbf{C}^H\end{cases} \iff \begin{cases}f(\mathbf{v}) = \text{crop}\left[\mathcal{F}^{-1}\left\{\mathcal{F}(\mathbf{h}) \cdot \mathcal{F}(\mathbf{v})\right\}\right] \\ f^H(x) = \mathcal{F}^{-1}\left\{\mathcal{F}(\mathbf{h})^* \cdot \mathcal{F}(\text{pad}\,[x])\right\},\end{cases}$$

where we have written **A** in its matrix formulation (left) and the corresponding way it is implemented in code (right). Note that we converted efficient operations like pointwise multiplication to matrices purely for the derivation. See the GD Jupyter notebook for the actual implementation of these operators.

GD Implementation

The iterative reconstruction of **v** looks like:

$$\begin{aligned}&\mathbf{v}_0 = \text{anything} \\ &\mathbf{v}_{k+1} \leftarrow \mathbf{v}_k - \alpha_k \mathbf{A}^H(\mathbf{A}\mathbf{v}_k - \mathbf{b}) \\ &\qquad \text{Repeat forever}\end{aligned}$$

$\mathcal{F}(\mathbf{h})$ can be precomputed (because **h** is measured beforehand), and the action of $\text{diag}(\mathbf{Fh})^H$ can be implemented as pointwise multiplication with the conjugate $\mathcal{F}(h)^*$. Since all the other operations involve only Fourier transforms, every operation in the gradient calculation can be efficiently calculated. For implementation details, see the GD Jupyter notebook.

In our problem, we need to keep in mind the physical interpretation of **v**. Since it represents an image, it must be non-negative. We can add this constraint into the algorithm by "projecting" **v** onto the space of non-negative images. In short, we zero all negative pixel values in the current image estimate at every iteration.

One thing to keep in mind is the step size, α_k. We want it to be large at first – "coarse" jumps to get closer to the minimum quickly. As we get closer, large steps will cause the

estimate to “bounce around" the minimum, overshooting it each time. Ideally we would want to decrease the step size with each iteration at a rate that would ensure continual progress. While varying step size might yield a faster convergence, it requires hand tuning and can be time consuming. A constant but sufficiently small step size is guaranteed to converge, with no parameter tuning necessary. In our case, it is possible to calculate the largest constant step size that guarantees convergence in terms of $\mathbf{A}$: $0 < \alpha < \frac{2}{\|\mathbf{A}^H\mathbf{A}\|_2}$, where $\|\mathbf{A}^H\mathbf{A}\|_2$ is the maximum singular value of $\mathbf{A}^H\mathbf{A}$ (see this page for why). The GD Jupyter notebook shows how we actually approximate this singular value (using $\mathbf{H}$ instead).

Lastly, all convergence guarantees are for an infinite number of iterations: “repeat forever". In practice, after a certain number of iterations (which varies by application) the updates are too small to change the estimate significantly. In our case, after incorporating the speedup techniques below, most of the progress is seen in the first 150-200 iterations. Sharper, more detailed images may require a few hundred more.

We also need to supply an initial “guess" of our image. It doesn't actually matter what we use for this. Currently, we are using a uniform image of half intensity, but you could initialize with all 0's or a random image.

Incorporating all of these details, we have:

$$
\begin{aligned}
&\mathbf{v}_0 = I/2 \\
&\textbf{for } k = 0 \text{ to num\_iters:} \\
&\quad \mathbf{v}'_{k+1} \leftarrow \mathbf{v}_k - \frac{1.8}{\|\mathbf{A}^H\mathbf{A}\|}\mathbf{A}^H(\mathbf{A}\mathbf{v}_k - \mathbf{b}) \\
&\quad \mathbf{v}_{k+1} \leftarrow \underset{\mathbf{v}\geq 0}{\text{proj}}(\mathbf{v}'_{k+1})
\end{aligned}
$$

Gradient Descent Speedup

Gradient descent as written above works, but in practice, people always add a “momentum term" that incorporates the old descent direction into the calculation of the new descent direction. This guards against changing the descent direction too much and too often, which can be counterproductive. We implement momentum by introducing μ, a factor that determines how much the new descent direction is determined by the old descent direction. Typically $\mu = 0.9$ is a good place to start. Another common practice is to use “Nesterov" momentum, which involves an intermediate update $\mathbf{p}$. We call this method, along with the projection step, “accelerated projected gradient descent".

$$
\begin{aligned}
&\mathbf{v}_0 = I/2, \quad \mu = 0.9, \quad \mathbf{p}_0 = 0 \\
&\textbf{for } k = 0 \text{ to num\_iters:} \\
&\quad \mathbf{p}_{k+1} \leftarrow \mu\mathbf{p}_k - \alpha_k \texttt{grad}(\mathbf{v}_k) \\
&\quad \mathbf{v}'_{k+1} \leftarrow \mathbf{v}_k - \mu\mathbf{p}_k + (1+\mu)\mathbf{p}_{k+1} \\
&\quad \mathbf{v}_{k+1} \leftarrow \underset{\mathbf{v}\geq 0}{\text{proj}}(\mathbf{v}'_{k+1})
\end{aligned}
$$

See this page for more details on parameter updates using momentum terms.

FISTA

Another way to speed up gradient descent is the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA). This also computes the accelerated projected gradient descent, but is more flexible about what the projection step (or more generally the "proximal" step p_L) does. For example, one can show that doing accelerated descent with ℓ_1-regularization only requires exchanging the projection step with a soft-thresholding step. Enforcing sparsity in other domains (for instance, on the gradient of the image rather than the image itself) can be achieved via soft-thresholding transformations of the image. This algorithm is very useful for solving linear inverse problems in image processing.

Each iteration is as follows (see this paper for a derivation and explanation of each term):

$$
\begin{aligned}
&\mathbf{v}_0 = I/2, \quad t_1 = 1, \quad x_0 = \mathbf{v}_0 \\
&\textbf{for } k = 0 \text{ to num\_iters:} \\
&\qquad x_k \leftarrow p_L(\mathbf{v}_k) \\
&\qquad t_{k+1} \leftarrow \frac{1 + \sqrt{1 + 4t_k^2}}{2} \\
&\qquad \mathbf{v}_{k+1} \leftarrow x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})
\end{aligned}
$$

ADMM

Although gradient descent is a reliable algorithm that is guaranteed to converge, it is still slow. If we want to process larger sets of data (e.g. 3D imaging), have a live feed of DiffuserCam, or just want to process images more quickly, we need to tailor the algorithm more closely to the optical system involved. While this introduces more tuning parameters ("knobs" to turn), speed of reconstruction can be drastically improved. Here we present (without proof) the result of using *alternating direction method of multipiers (ADMM)* to reconstruct the image.

We will only briefly motivate the use of ADMM and then provide the derivation of the update steps specific to our problem. For background on ADMM, please refer to sections 2 and 3 of: Prof. Boyd's ADMM tutorial. To understand this document, background knowledge from Chapters 5 (Duality) and 9 (Unconstrained minimization) from his textbook on optimization may be necessary.

Recall the original minimization problem:

$$
\hat{\mathbf{v}} = \underset{\mathbf{v} \geq 0}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{b} - \mathbf{A}\mathbf{v}\|_2^2, \tag{8.1}
$$

where 2D images are interpreted as vectors. We seek to *split* the single minimization over

the vector $\mathbf{v}$ into separable minimizations – for example:

$$\begin{aligned} \hat{\mathbf{v}} = \underset{w \geq 0, x}{\operatorname{argmin}} \ & \frac{1}{2}\|\mathbf{b} - \mathbf{C}v\|_2^2 \\ \text{s.t. } & x = \mathbf{H}\mathbf{v}, w = \mathbf{v}, \end{aligned} \tag{8.2}$$

where we have decomposed the action of DiffuserCam $\mathbf{C} = \mathbf{C}\mathbf{H}$ into the convolution $\mathbf{H}$ followed by a crop $\mathbf{C}$. The primary reason is to make the expression more amenable to the ADMM algorithm, which adds a set of "update steps" for each additional constraint. If we don't find a nice decomposition, some of these updates will be inefficent to calculate.

In addition, because of these parallel update steps, we can add constraints (prior information) easily. A common useful prior we add is to encourage the gradient of the image to be sparse – most natural images can be approximated by piecwise constant intensities. Typically, gradient sparsity is enforced through "total variation" regularization, where we include the ℓ_1-norm of the gradient in our objective function:

$$\begin{aligned} \hat{\mathbf{v}} = \underset{w \geq 0, u, x}{\operatorname{argmin}} & \frac{1}{2}\|\mathbf{b} - \mathbf{C}x\|_2^2 + \tau\|u\|_1 \\ \text{s.t. } & x = \mathbf{H}\mathbf{v}, u = \Psi\mathbf{v}, w = \mathbf{v}, \end{aligned} \tag{8.3}$$

where Ψ is a derivative (difference) operator.

The next step is to form the *augmented Lagrangian* (see section 2.3 in the ADMM reference), which can be directly read off from the constraints and objective function:

$$\begin{aligned} \mathcal{L}(\{u, x, w, \mathbf{v}\}, \{\xi, \eta, \rho\}) = & \frac{1}{2}\|\mathbf{b} - \mathbf{C}x\|_2^2 + \tau\|u\|_1 \\ & + \frac{\mu_1}{2}\|\mathbf{H}\mathbf{v} - x\|_2^2 + \xi^\intercal(\mathbf{H}\mathbf{v} - x) \\ & + \frac{\mu_2}{2}\|\Psi\mathbf{v} - u\|_2^2 + \eta^\intercal(\Psi\mathbf{v} - u) \\ & + \frac{\mu_3}{2}\|\mathbf{v} - w\|_2^2 + \rho^\intercal(\mathbf{v} - w) \\ & + \mathbb{1}_+(w), \end{aligned} \tag{8.4}$$

where the $\mathbb{1}_+(w)$ term arises from the implicit constraint $w \geq 0$:

$$\mathbb{1}_+(w) = \begin{cases} \infty & w < 0 \\ 0 & w \geq 0 \end{cases}$$

The Lagrangian dual approach to minimizing the objective function is to solve the following optimization problem:

$$\underset{\xi, \eta, \rho}{\text{maximize}} \ \min_{u, x, w, \mathbf{v}} \mathcal{L}(\{u, x, w, \mathbf{v}\}, \{\xi, \eta, \rho\}) \tag{8.5}$$

The min above indicates that, ideally, we would want to jointly minimize over all the *primal* variables $(u, x, w, \mathbf{v})$ first, before performing the outer maximization over the *dual*

variables (ξ, η, ρ). The ADMM algorithm is a specific way of iteratively finding this optimal point. In reality, we only have estimates of each of the variables, so the algorithm updates our estimates for the minimum primal variables during every iteration that solves for the maximum dual variables.

Based on this paradigm, we can write down all the intermediate updates that take place in one "global" update step:

$$\text{Primal Updates:} \begin{cases} u_{k+1} & \leftarrow \text{argmin}_u \ \mathcal{L}(\{u, x_k, w_k, \mathbf{v}_k\}, \{\xi_k, \eta_k, \rho_k\}) \\ x_{k+1} & \leftarrow \text{argmin}_x \ \mathcal{L}(\{u_{k+1}, x, w_k, \mathbf{v}_k\}, \{\xi_k, \eta_k, \rho_k\}) \\ w_{k+1} & \leftarrow \text{argmin}_w \ \mathcal{L}(\{u_{k+1}, x_{k+1}, w, \mathbf{v}_k\}, \{\xi_k, \eta_k, \rho_k\}) \\ \mathbf{v}_{k+1} & \leftarrow \text{argmin}_{\mathbf{v}} \ \mathcal{L}(\{u_{k+1}, x_{k+1}, w_{k+1}, \mathbf{v}\}, \{\xi_k, \eta_k, \rho_k\}) \end{cases}$$

$$\text{Dual Updates:} \begin{cases} \xi_{k+1} & \leftarrow \xi_k + \mu_1(\mathbf{H}\mathbf{v}_k - x_{k+1}) \\ \eta_{k+1} & \leftarrow \eta_k + \mu_2(\Psi\mathbf{v}_{k+1} - u_{k+1}) \\ \rho_{k+1} & \leftarrow \rho_k + \mu_2(\mathbf{v}_{k+1} - w_{k+1}) \end{cases}$$

Notice that each dual update step tries to solve the maximization problem via gradient *ascent*. In each global iteration, we make one step in the ascent direction.

Next, for each primal variable, the individual optimization problem only depends on the terms in the Lagrangian corresponding to that variable. For example, in the u-update, we only need to include the terms $\tau\|u\|_1$, $\frac{\mu_2}{2}\|u - \Psi\mathbf{v}\|_2^2$, and $\eta^\intercal(u - \Psi\mathbf{v})$; all the other terms are constant with respect to u. So, we have:

$$\begin{cases} u_{k+1} & \leftarrow \text{argmin}_u \ \tau\|u\|_1 + + \frac{\mu_2}{2}\|\Psi\mathbf{v}_k - u\|_2^2 + \eta_k^\intercal(\Psi\mathbf{v}_k - u) \\ x_{k+1} & \leftarrow \text{argmin}_x \ \frac{1}{2}\|\mathbf{b} - \mathbf{C}x\|_2^2 + \frac{\mu_1}{2}\|\mathbf{H}\mathbf{v}_k - x\|_2^2 + \xi_k^\intercal(\mathbf{H}\mathbf{v}_k - x) \\ w_{k+1} & \leftarrow \text{argmin}_w \ \frac{\mu_3}{2}\|\mathbf{v}_k - w\|_2^2 + \rho_k^\intercal(\mathbf{v}_k - w) + \mathbb{1}_+(w) \\ \mathbf{v}_{k+1} & \leftarrow \text{argmin}_{\mathbf{v}} \ \frac{\mu_1}{2}\|\mathbf{H}\mathbf{v} - x_{k+1}\|_2^2 + \frac{\mu_2}{2}\|\Psi\mathbf{v} - u_{k+1}\|_2^2 + \frac{\mu_3}{2}\|\mathbf{v}_{k+1} - w_{k+1}\|_2^2 \end{cases}$$

$$\begin{cases} \xi_{k+1} & \leftarrow \xi_k + \mu_1(\mathbf{H}\mathbf{v}_k - x_{k+1}) \\ \eta_{k+1} & \leftarrow \eta_k + \mu_2(\Psi\mathbf{v}_{k+1} - u_{k+1}) \\ \rho_{k+1} & \leftarrow \rho_k + \mu_2(\mathbf{v}_{k+1} - w_{k+1}) \end{cases}$$

The primal minimization updates can be solved using standard convex optimization techniques, which are worked out in the DiffuserCam Derivations Supplement. The results are:

$$\begin{aligned} u_{k+1} &\leftarrow \mathcal{T}_{\frac{\tau}{\mu_2}}\left(\Psi\mathbf{v}_k + \eta_k/\mu_2\right) \\ x_{k+1} &\leftarrow \left(\mathbf{C}^\intercal\mathbf{C} + \mu_1 I\right)^{-1}\left(\xi_k + \mu_1\mathbf{M}\mathbf{v}_k + \mathbf{C}^\intercal\mathbf{b}\right) \\ w_{k+1} &\leftarrow \max(\rho_k/\mu_3 + \mathbf{v}_k, 0) \\ \mathbf{v}_{k+1} &\leftarrow (\mu_1\mathbf{M}^\intercal\mathbf{M} + \mu_2\Psi^\intercal\Psi + \mu_3 I)^{-1} r_k, \\ \xi_{k+1} &\leftarrow \xi_k + \mu_1(\mathbf{H}\mathbf{v}_{k+1} - x_{k+1}) \\ \eta_{k+1} &\leftarrow \eta_k + \mu_2(\Psi\mathbf{v}_{k+1} - u_{k+1}) \\ \rho_{k+1} &\leftarrow \rho_k + \mu_2(\mathbf{v}_{k+1} - w_{k+1}) \end{aligned}$$

where

$$r_k = (\mu_3 w_{k+1} - \rho_k) + \Psi^\intercal(\mu_2 u_{k+1} - \eta_k) + \mathbf{M}^\intercal(\mu_1 x_{k+1} - \xi_k)$$

8.6 Miniscope3D supplemental details

Microlenses vs Gaussian Diffuser

For our phase mask, we choose a microlens array instead of the Gaussian diffuser used in our previous work[11]. This is because the microlenses can achieve point spread functions (PSFs) with higher SNR and frequency content than the diffuser (see Fig. 8.5), due to their better concentration of light in focus. Microlenses focus light into small focus spots, with dark areas between them, as opposed to the diffuser, which has some light spread between the caustics, generating unwanted low frequencies in the PSFs. Sharper focus spots in the microlens PSF mean that the SNR of the measurements is better and the inverse problem better posed. While using fewer focal spots would improve 2D measurement SNR and resolution, using a small number of microlenses does not provide enough multiplexing to gain 3D capability over a large depth range.

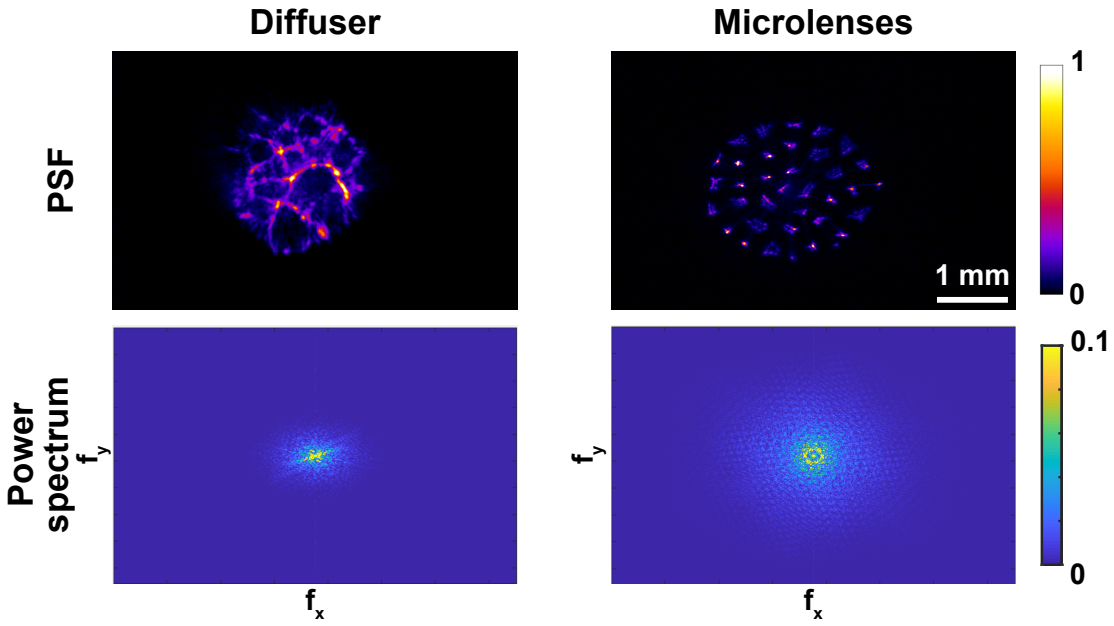


Figure 8.5: Comparison of experimental PSFs resulting from a Gaussian diffuser and our microlens phase mask. The microlenses generate PSFs with more high-frequency content, as seen in the power spectrum. The microlenses also have better light concentration; to achieve the same brightness as the microlenses PSF, the diffuser requires 4× the exposure time.

Axial Resolution

We determined the axial resolution by imaging a thin layer of 4.8 μm fluorescent beads. Because it is difficult to controllably place two beads at specific axial separation distances, raw data from a single bead at different depths are digitally added in order to synthesize a measurement of two layers of beads with varying separations. Figure 8.6 shows that we achieve a uniform 15μm axial resolution across our depth range of 360 μm. This closely matches with the axial full-width-half-maximum (FWHM) we observe in the 3D fluorescent beads sample in the main-paper *Results* section.

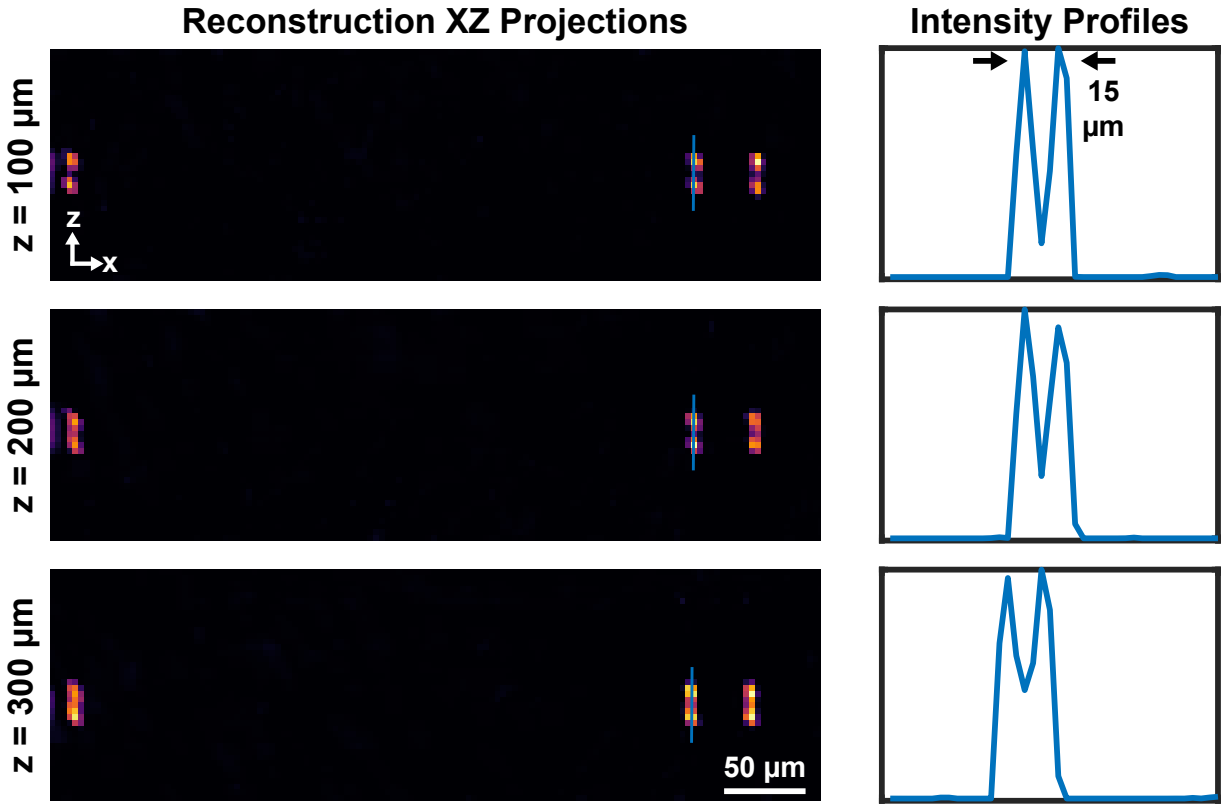


Figure 8.6: Reconstructions results demonstrating 15μm axial resolution across our depth range. On left are x-z projections of the 3D reconstruction for the case of two layers of 3 beads each, separated by 15 μm axially. At right we show cross-cuts of the projections demonstrating clear resolving of the beads. The rows show results for placing the pairs of beads at different axial distances from the native focus plane.

Lateral Resolution

Examining a single microlens, the Rayleigh criterion defines the minimum resolvable separation of two diffraction-limited spots on the sensor, $\delta x'$, in terms of the wavelength, λ, the

microlens clear aperture, Δ_{ML}, and the distance from the mask to the sensor, t:

$$\delta x' = \frac{1.22\lambda t}{\Delta_{ML}} = M\delta x. \tag{8.6}$$

Here we have used the fact that two points in object space separated by δ_x will appear as a separation of $M\delta_x$ on the sensor. Thus, we need to calculate the magnification of our system.

We use ray transfer matrices (with a paraxial approximation) to evaluate the magnification of the system. The system ABCD matrix is:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1/f_\mu & 1 \end{bmatrix} \begin{bmatrix} A_G & B_G \\ C_G & D_G \end{bmatrix} \begin{bmatrix} 1 & Q \\ 0 & 1 \end{bmatrix} \tag{8.7}$$

and the system magnification, which is used in the lateral resolution derivation, is:

$$M = B = \left(1 - \frac{t}{f_\mu}\right) A_G + tC_G. \tag{8.8}$$

where A_G, B_G, C_G, & D_G are elements for the GRIN's ray transfer matrix ($A_G = 0.0725$, $B_G = 1.6931$, $C_G = -0.599$, and $D_G = 0.124$) and t is the distance from the phase mask to the sensor. Given that f_μ, the microlens focal length, ranges from 7 mm to 25 mm, combined with the small value for A_G, this results in the first term, $(1 - t/f_\mu)A_G$, being negligible and the magnification can be approximated simply as tC_G. This shows that for our system, the magnification is given by:

$$M \approx tC_G. \tag{8.9}$$

Substituting Eq. 8.9 into Eq. 8.6 and solving for Δ_{ML}, we get an expression for the microlens clear aperture needed for a target object resolution:

$$\Delta_{ML} = \frac{1.22\lambda t}{M\delta x} \approx \frac{1.22\lambda}{C_G \delta x} \tag{8.10}$$

Depth of Focus

We aim to determine the microlens depth-of-focus (DoF), defined as the distance that a point source in-focus can move axially before the blur spot on the camera sensor is bigger than a target circle of confusion radius, γ_c. To do so, we examine a single microlens' image in the GRIN entrance pupil for an object at distance z from the first principal plane of the GRIN. As the object moves axially by a distance d_{ML}, we can use similar triangles to derive (see Fig. 8.8 for variable definitions):

$$\frac{y}{d_{ML}} = \frac{\Delta_{EP}}{d_{ML} + z + L} \approx \frac{\Delta_{EP}}{L}, \tag{8.11}$$

where Δ_{EP} is the radius of the microlens' clear aperture in the entrance pupil (i.e. object side) of the GRIN and L is the distance from the first principal plane to the entrance pupil.

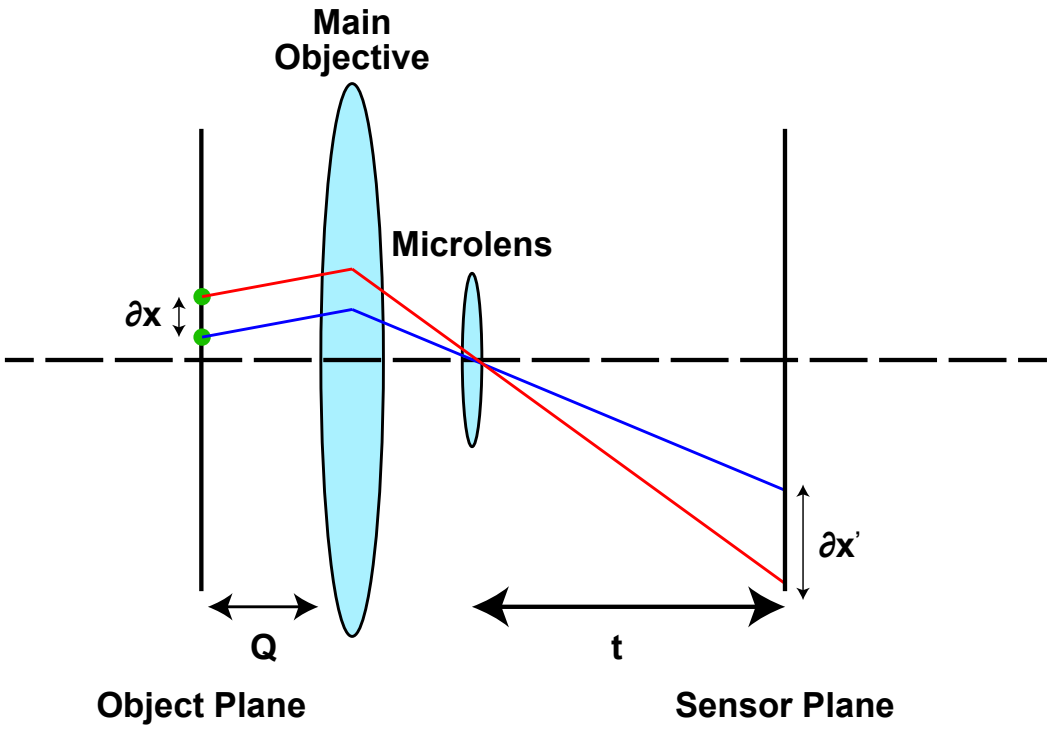


Figure 8.7: Lateral resolution derivation. Examining a single microlens placed immediately after the main objective.

Given that $L = 13\ mm$ is much larger than z, d_{ML}, which are on the order of 0.2 mm, we drop both z and d_{ML}. By substituting $y = \gamma_c/M$ into Eq. 8.11, we can solve for the microlens DoF as a function of our system parameters:

$$d_{ML} = \frac{\gamma_c L}{\Delta_{EP} M}. \tag{8.12}$$

Since the entrance pupil of the GRIN is very far from the object (it is approximately telecentric in object space), the object axial position is negligible in determining the microlens DoF. Designing for $\gamma_c = 12\ \mu m$, a circle-of-confusion smaller than the diffraction-limited spot size, $|M|\delta_x$, and using $\Delta_{EP} = 4\ mm$ (calculated using Zemax for a microlens with a clear aperture of 300 μm), we determine the DoF to be $\pm 20\ \mu m$.

Choice of Reconstruction Grid

To successfully reconstruct $\mathbf{v}$, we should define the reconstruction grid with sufficient sampling to realize the best resolution possible, but without oversampling, which increases computation and memory requirements. The theory above defines a band-limit for the measurements, so our goal is to use a sensor with a matching effective pixel size. In our architecture, increasing the sensor pixel size directly corresponds to increased lateral reconstruction voxel

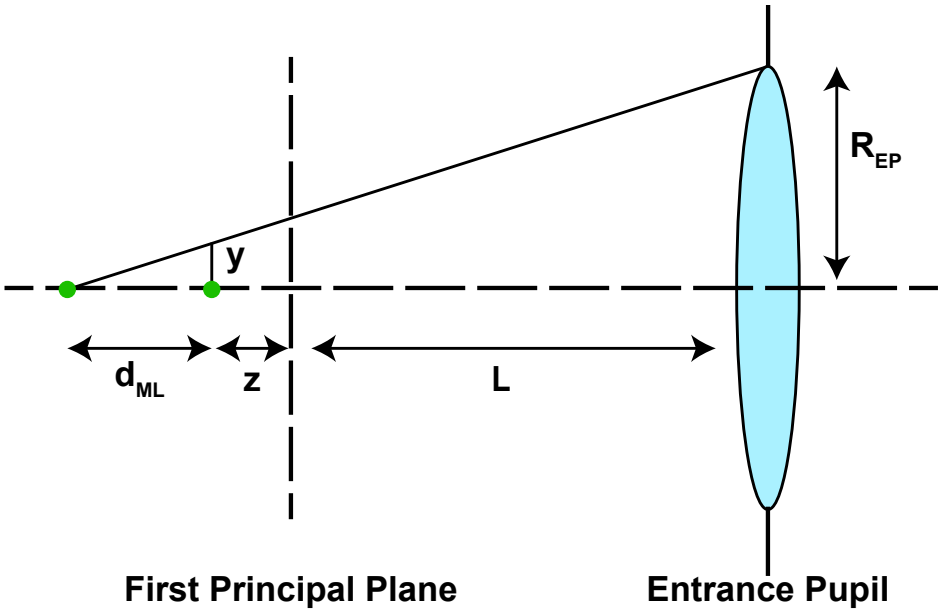


Figure 8.8: Depth-of-focus (DoF) derivation setup, with distance variables defined.

size and lower final resolution. Because of complicated interactions between nonlinear reconstructions and grid size, we determine our choice of lateral sampling empirically by binning the raw data from the resolution tests in the main paper *Results* section by 2×, 4×, and 8× and evaluating the final resolution. We find that the resolution begins to degrade between 4× and 8× binning, so we operate at 4× binning. This results in our sensor's effective object-space pixel size being 1.7 μm, which is sufficiently below the 2.76 μm minimum feature size that we observe experimentally. Note that the ability to use on-chip binning allows our approach to read data faster than a conventional LFM, which cannot use conventional on-chip binning without resolution loss. This allows us to achieve a 40 volume-per-second measurement rate using a low-cost USB 2.0 camera.

The choice of axial sampling informs our sampling interval during calibration (main-paper *Calibration* subsection). We measure every 5 μm, and perform axial binning (summing of consecutive PSFs) at 1×, 2×, and 4×. We find 1× yields the best results. The resulting 5 μm axial sampling is reasonable given the empirically observed 15 μm axial resolution. Hence our choice of grid balances fast frame rates and efficient reconstruction with image quality and resolution.

Choice of Regularization Parameter

One important parameter in our optimization problem is the regularization parameter τ. The regularization parameter sets the trade-off between the data fidelity term and our sparsity prior. In practice, this parameter sets the balance between preserving image details and noise reduction. Very small values of τ will preserve sharp details in our object; however,

the reconstructions can be noisy. Very large values will suppress noise, but also suppress the object's details with it.

To test the reconstruction quality as a function of the regularization parameter, we ran our 3D reconstruction algorithm on the experimental resolution target data at $z = 270\ \mu m$ with values of τ ranging from 10^{-14} to 10^{-1}. Figure 8.9(a) shows that the reconstructions and the data fidelity term are stable for a wide range of τ values. As expected, for very large values of τ, the Total Variation (TV) prior over-regularizes the image, resulting in smoothed out details.

Since the experimental data lacks ground truth to compare against, we simulate a raw measurement by running our 3D shift-varying forward model on a two-photon microscopy zebra fish 3D dataset with our measured PSFs and adding realistic additive white Gaussian noise. The measurement is then processed with values of τ ranging from 10^{-14} to 10^{-1}. Figure 8.9(b) shows a trend similar to experimental results - the mean-squared error is stable for a large range of τ values, with over-smoothed reconstructions as τ gets very large. We note that all the data shown in the main paper was processed using the same value of τ, which further show that once a good value for τ is found, it can be used to process different classes of objects. While it may be possible to fine-tune τ for each measurement to achieve better performance, it is, however, more practical for users to use the default value. If the user is to fine-tune τ, we recommend using the largest value of τ that still preserves the object's fine details.

2D Miniscope PSNR Comparison

Our Miniscope3D design is aimed at 3D imaging, but because it is smaller and lighter weight than 2D Miniscope, it might be useful in applications that only require 2D imaging. Because of the inherent aberrations in the GRIN lens, the 2D Miniscope does not achieve its full-aperture diffraction-limited resolution and our Miniscope3D resolution is only marginally worse than the 2D. However, we do suffer from reduced SNR as compared to the 2D Miniscope, because our PSFs spread the light over a larger area than a focused 2D Miniscope. To quantify this loss of SNR, we simulate measurements using on-axis PSFs from both our device and the 2D Miniscope (single lens with 2 μm blur). The simulation is performed at 3 light levels (100, 1000, and 10,000 photocounts) using a shift-invariant model with Poisson and read noise added. We use our reconstruction algorithm with an optimized τ value and display the results in Fig. 8.10. For a fair comparison, we show both the 2D Miniscope raw image and one reconstructed from an image deconvolution process. Our Miniscope3D system has better PNSR than the unprocessed 2D Miniscope data, but the deconvolved 2D Miniscope result performs the best, as expected. This is because our algorithm is denoising and deblurring. For a scene that does not fit our denoising priors, the processed results would perform worse. Also, note that the loss of PSNR in our system for 2D imaging is a neceessary sacrifice for gaining single-shot 3D imaging capability.

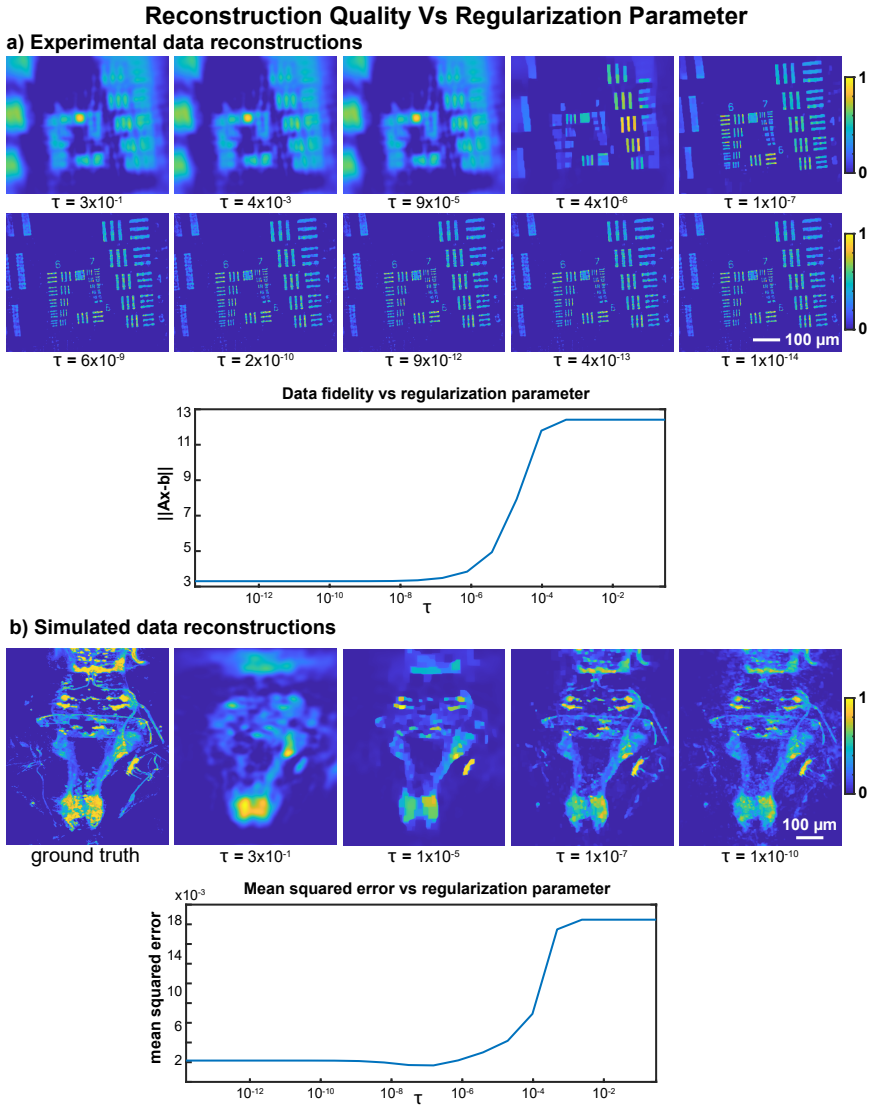


Figure 8.9: Reconstruction quality as a function of regularization parameter, τ. (a) Maximum intensity projections of an experimental volume reconstructed with different τ settings, along with a plot of the data fidelity term as a function of τ on a semi-log scale. (b) Maximum intensity projections of a simulated volume reconstructed with different τ settings, along with a plot of mean-squared error as a function of τ on a semi-log scale. The results demonstrate the stability of reconstructions for a large range of τ values.

Sparsity Comparison

Our approach assumes the object to have a sparse representation in some domain. In this paper, we use a general TV sparsity prior to promote gradient sparsity. This is a commonly-used prior for fluorescent imaging for a number of reasons: (1) fluorescent samples are generally sparsely labeled. (2) Even if a 2D slice of the sample is not spatially sparse, it will be sparse when considered with respect to our full 3D volume. (3) If native sparsity does not hold, images are generally sparse in gradient or wavelet domain. (4) Time-priors can further

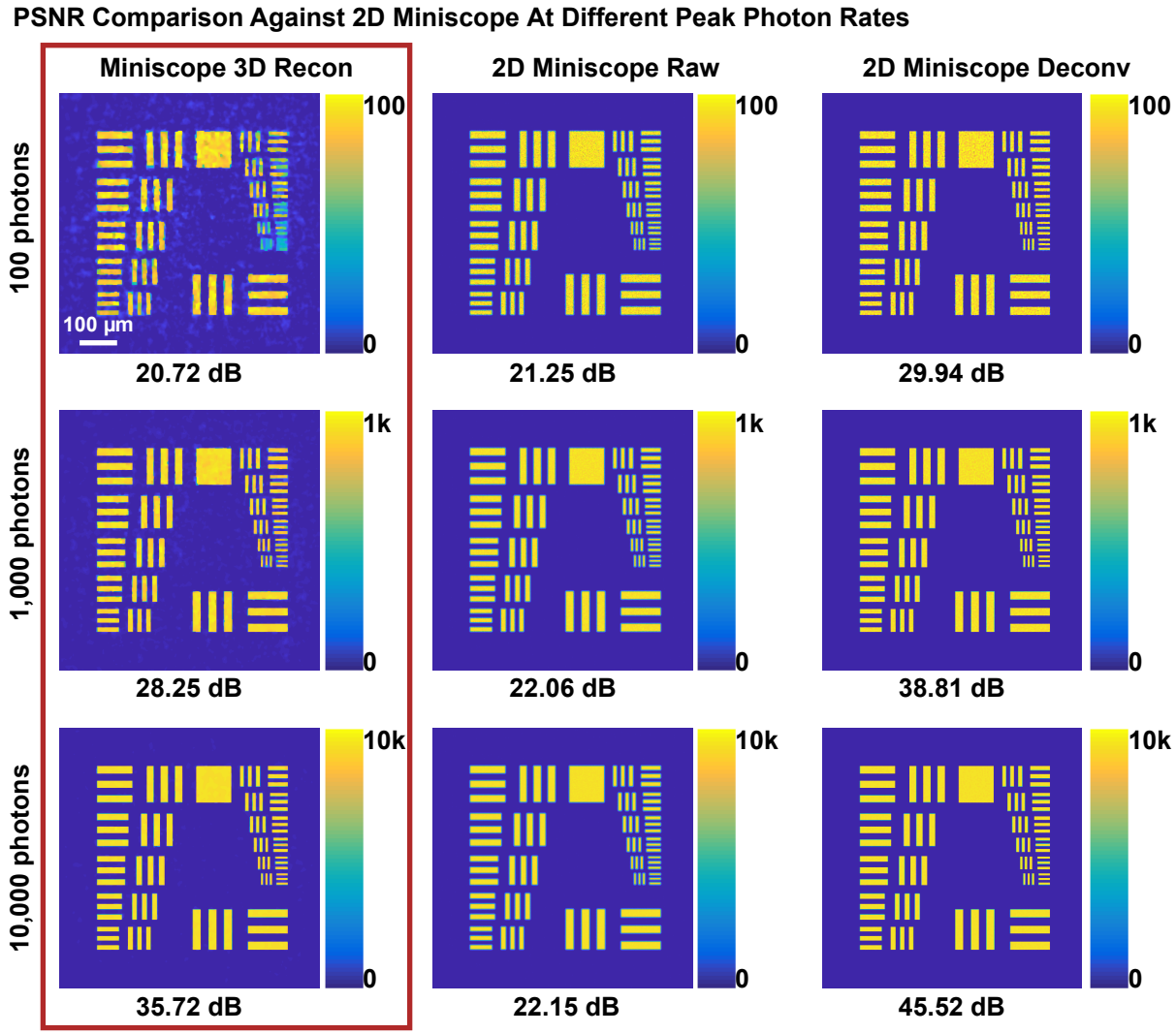


Figure 8.10: PSNR comparison of Miniscope3D and 2D Miniscope. (Left) Simulated reconstructions from our system at different light levels. (Middle) 2D Miniscope (simulated) raw measurement. (Right) 2D Miniscope deconvolved reconstructions. The multiplexing properties of our system that enable 3D capabilities result in a loss of PSNR.

render a volume sparse by only considering temporally-varying information (i.e. neural firings). While it is an NP hard problem to generate a phase transition curve for our system as it requires running a large number of reconstructions of many different classes of objects at each sparsity level, we give an example of how our system performs at different sparsity levels by thresholding a 3D volume to generate different sparsity levels and reporting mean-squared error (MSE) and PSNR. The simulated volume is of a 3D zebrafish dataset. The simulations are done using our 3D shift-varying model and the experimental PSFs from our system. Figure 8.11 shows MSE and PSNR for the reconstructed volume at different sparsity levels (33%, original volume, to 0.2%, thresholded volume). As expected, our system performs

better for sparser volumes. For denser volumes, our system recovers a lower-resolution version of the object and does not fail catastrophically.

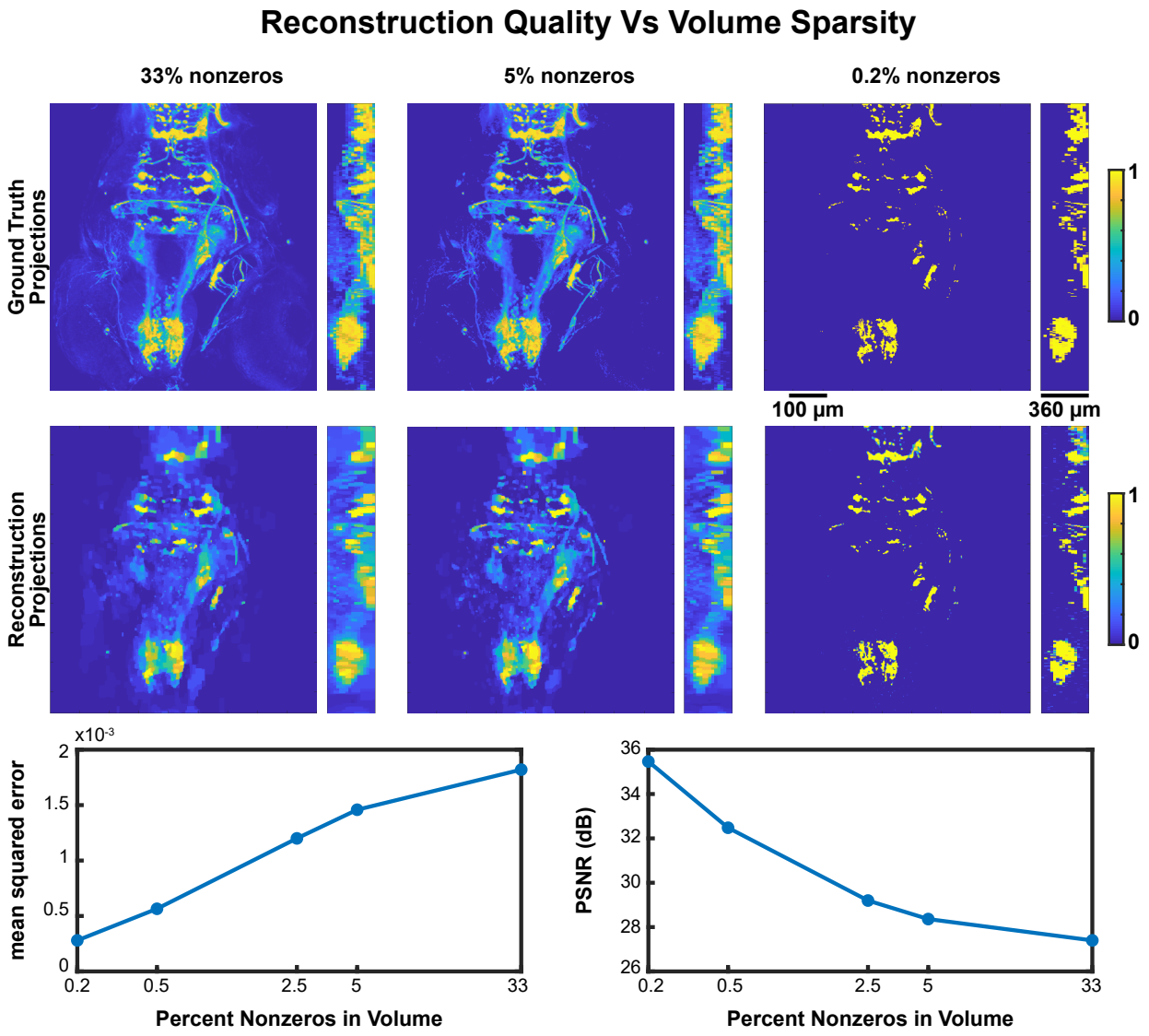


Figure 8.11: Simulations of reconstruction quality at different sparsity levels. Maximum intensity projections (y-x, z-x) show the quality of our reconstructions as compared to the ground truth at different sparsity levels. As the volume gets more dense, our reconstruction resolution degrades.

Guide to Different Designs Using Our Theory

Our theory is general and enables other users to design their own optimized 3D microscope targeting different resolutions or volumes-of-interest. To do so, users should implement the following design process:

- For a target lateral resolution, determine the microlens' average clear aperture needed to support that resolution (main paper Sec. *Lateral Resolution*). This also determines the number of microlenses in the phase mask.
- For a target depth range, distribute the focal lengths dioptrically across the depth range.
- Using our optimization criterion, optimize the microlenses positions and aberrations to further enhance the 3D performance.
- Fabricate the phase mask using our adaptive stitching algorithm with a Nanoscribe 3D printer.

Adaptive Stitching

The Nanoscribe 3D printer can only print across a field-of-view (FoV) of 350 μm, and so the 1.8 mm sized phase mask must be printed in multiple stitched blocks, with the mask translating between them. Due to the optical requirements on the microlenses, care needs to be taken when dividing the microlens array into blocks for printing with Nanoscribe. Our adaptive stitching approach aims to print each lens with minimal stitching artifacts. As the clear aperture for each lens is of the same order of magnitude as the maximum printing block size of Nanoscribe, each stitching block will correspond approximately to a single microlens. The center location of each microlens is known, so the problem reduces to dividing the plane in a number of regions, with each region attributed to one of the microlens centres. Preferably, the stitching lines should then fall in the overlapping region of two (or more) microlenses. We assume that such a division will result in the best possible optical quality.

This problem definition is quite similar to the basic Voronoi segmentation, where we are given a set of points in a plane and the task is to attribute each location in the plane to one of the given points. That problem is solved as follows. For each location in the plane, the distance to all centres is calculated. Attribution to one centre is then decided by it being the closest one (minimum search). As a result, a dividing line is defined by the fact that the distance to two or more centres is equal. The question now is, how can this be adapted to take into account finite shapes?

Rephrasing, we need to define a smooth function in the plane for each microlens followed by attributing locations to microlenses based on a (minimum) search over these different functions. To this end, we will use the height function for each microlens individually and then do a maximum search for the attribution. As a result, segmentation lines would fall exactly at those locations where the height of two or more microlenses are equal (see Fig.6 of main-paper). This is precisely what we want to achieve.

The resulting height-based segmentation is shown in Fig. 8.12. Here, different slices are shown (50 to 53 μm height). Colored regions need to be printed by Nanoscribe as a single FoV. The different colors correspond to different stitching blocks.

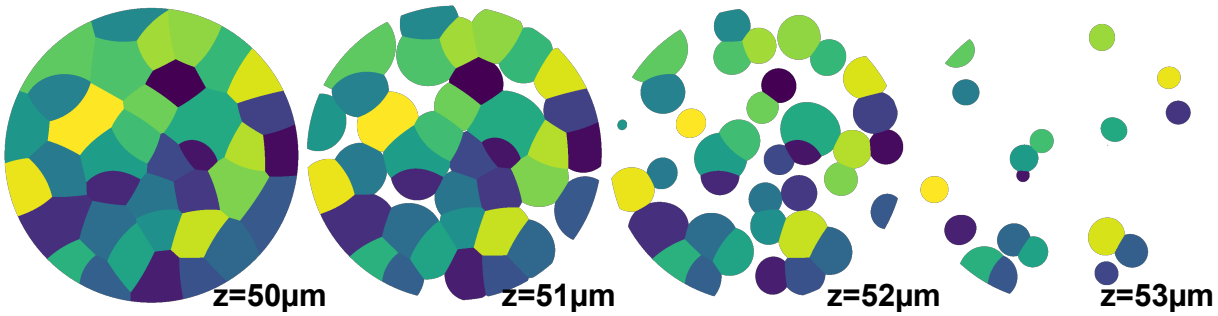


Figure 8.12: Different slices are shown, with different colors corresponding to different stitching blocks.

Bibliography

[1] Jesse K Adams, Vivek Boominathan, Benjamin W Avants, Daniel G Vercosa, Fan Ye, Richard G Baraniuk, Jacob T Robinson, and Ashok Veeraraghavan. "Single-frame 3D fluorescence microscopy with ultraminiature lensless FlatScope". In: *Science Advances* 3.12 (2017), e1701548.

[2] Edward H Adelson and James R Bergen. *The plenoptic function and the elements of early vision*. Vol. 2. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of ..., 1991.

[3] Edward H Adelson and John Y A Wang. "Single lens stereo with a plenoptic camera". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 14.2 (1992), pp. 99–106.

[4] M V Afonso, J M Bioucas-Dias, and M A T Figueiredo. "Fast image recovery using variable splitting and constrained optimization". In: *IEEE Transactions on Image Processing* 19.9 (Sept. 2010), pp. 2345–2356.

[5] M S C Almeida and M Figueiredo. "Deconvolving images with unknown boundaries using the alternating direction method of multipliers". In: *IEEE Transactions on Image processing* 22.8 (2013), pp. 3074–3086.

[6] Percival F Almoro, Laura Waller, Mostafa Agour, Claas Falldorf, Giancarlo Pedrini, Wolfgang Osten, and Steen G Hanson. "Enhanced deterministic phase retrieval using a partially developed speckle field". In: *Optics Letters* 37.11 (June 2012), pp. 2088–2090. DOI: 10.1364/OL.37.002088. URL: http://ol.osa.org/abstract.cfm?URI=ol-37-11-2088.

[7] Arun Anand, Vani K Chhaniwal, Percival Almoro, Giancarlo Pedrini, and Wolfgang Osten. "Shape and deformation measurements of 3 D objects using volume speckle field and phase retrieval". In: *Optics Letters* 34.10 (2009), pp. 1522–1524.

[8] N Antipa, S Necula, R Ng, and L Waller. "Single-shot diffuser-encoded light field imaging". In: *2016 IEEE International Conference on Computational Photography (ICCP)*. May 2016, pp. 1–11. DOI: 10.1109/ICCPHOT.2016.7492880.

[9] Nick Antipa. *Supplemental material: Video from Stills: Lensless Imaging with Rolling Shutter*. Apr. 2019. DOI: 10.6084/m9.figshare.7961138.v1. URL: https://figshare.com/articles/media/Video_from_Stills_Lensless_Imaging_with_Rolling_Shutter/7961138%20https://ndownloader.figshare.com/files/14819612%20https://ndownloader.figshare.com/files/14819615%20https://ndownloader.figshare.com/files/14819618%20https://ndo.

[10] Nick Antipa, Grace Kuo, Reinhard Heckel, Ben Mildenhall, Emrah Bostan, Ren Ng, and Laura Waller. *DiffuserCam*. url: https://waller-lab.github.io/DiffuserCam/. 2017.

[11] Nick Antipa, Grace Kuo, Reinhard Heckel, Ben Mildenhall, Emrah Bostan, Ren Ng, and Laura Waller. " DiffuserCam: lensless single-exposure 3D imaging". In: *Optica* 5.1 (2018), pp. 1–9.

[12] Nick Antipa, Patrick Oare, Emrah Bostan, Ren Ng, and Laura Waller. "Video from stills: Lensless imaging with rolling shutter". In: *2019 IEEE International Conference on Computational Photography (ICCP)*. IEEE. 2019, pp. 1–8.

[13] M Salman Asif, Ali Ayremlou, Aswin Sankaranarayanan, Ashok Veeraraghavan, and Richard G Baraniuk. "Flatcam: Thin, lensless cameras using coded aperture and computation". In: *IEEE Transactions on Computational Imaging* 3.3 (2017), pp. 384–397.

[14] M Salman Asif, Ali Ayremlou, Ashok Veeraraghavan, Richard Baraniuk, and Aswin Sankaranarayanan. "Flatcam: Replacing lenses with masks and computation". In: *Computer Vision Workshop (ICCVW), 2015 IEEE International Conference on*. IEEE. 2015, pp. 663–666.

[15] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202.

[16] Amir Beck and Marc Teboulle. "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems". In: *IEEE Transactions on Image Processing* 18.11 (2009), pp. 2419–2434.

[17] J M Bioucas-Dias and M A T Figueiredo. "A New TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithms for Image Restoration". In: *Image Processing, IEEE Transactions on* 16.12 (Dec. 2007), pp. 2992–3004. ISSN: 1057-7149. DOI: 10.1109/TIP.2007.909319.

[18] Waheb Bishara, Ting-Wei Su, Ahmet F Coskun, and Aydogan Ozcan. "Lensfree on-chip microscopy over a wide field-of-view using pixel super-resolution". In: *Optics Express* 18.11 (2010), pp. 11181–11191.

[19] Vivek Boominathan, Kaushik Mitra, and Ashok Veeraraghavan. "Improving resolution and depth-of-field of light field cameras using a hybrid imaging system". In: *Computational Photography (ICCP), 2014 IEEE International Conference on*. IEEE. 2014, pp. 1–10.

[20] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Cambridge University Press, 1999.

[21] E Bostan, U S Kamilov, M Nilchian, and M Unser. "Sparse Stochastic Processes and Discretization of Linear Inverse Problems". In: *IEEE Transactions on Image Processing* 22.7 (July 2013), pp. 2699–2710.

[22] S Boyd, N Parikh, E Chu, B Peleato, and J Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends in Machine Learning* 3.1 (2011), pp. 1–122.

[23] David Brady, Kerkil Choi, Daniel Marks, Ryoichi Horisaki, and Sehoon Lim. "Compressive Holography". In: *Opt. Express* 17.15 (July 2009), pp. 13040–13049. DOI: 10.1364/OE.17.013040. URL: http://www.opticsexpress.org/abstract.cfm?URI=oe-17-15-13040.

[24] Michael Broxton, Logan Grosenick, Samuel Yang, Noy Cohen, Aaron Andalman, Karl Deisseroth, and Marc Levoy. "Wave optics theory and 3-D deconvolution for the light field microscope". In: *Optics Express* 21.21 (2013), pp. 25418–25439. DOI: 10.1364/OE.21.025418. URL: http://www.opticsexpress.org/abstract.cfm?URI=oe-21-21-25418.

[25] Emmanuel J Candès and Carlos Fernandez-Granda. "Towards a mathematical theory of super-resolution". In: *Communications on pure and applied Mathematics* 67.6 (2014), pp. 906–956.

[26] Emmanuel J Candès and Michael B Wakin. "An introduction to compressive sampling". In: *IEEE Signal Processing Magazine* 25.2 (2008), pp. 21–30.

[27] Wanli Chi and Nicholas George. "Optical imaging with phase-coded aperture". In: *Optics Express* 19.5 (2011), pp. 4294–4300.

[28] Michał J Cieślak, Kelum A A Gamage, and Robert Glover. "Coded-aperture imaging systems: Past, present and future development–A review". In: *Radiation Measurements* 92 (2016), pp. 59–71. ISSN: 1350-4487.

[29] O Cossairt, C Zhou, and S K Nayar. " Diffusion Coding Photography for Extended Depth of Field". In: *ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH)* (Aug. 2010).

[30] Oliver Cossairt, Mohit Gupta, and Shree K Nayar. "When does computational imaging improve performance?" In: *IEEE transactions on image processing* 22.2 (2012), pp. 447–458. ISSN: 1057-7149.

[31] Yuchao Dai, Hongdong Li, Mingyi He, and Chunhua Shen. "Smooth Approximation of L_infinity-Norm for Multi-view Geometry". In: *2009 Digital Image Computing: Techniques and Applications*. IEEE. 2009, pp. 339–346.

[32] S Dehaeck, B Scheid, and P Lambert. "Adaptive stitching for meso-scale printing with two-photon lithography". In: *Additive Manufacturing* 21 (2018), pp. 589–597. ISSN: 2214-8604. DOI: https://doi.org/10.1016/j.addma.2018.03.026. URL: http://www.sciencedirect.com/science/article/pii/S2214860417305766.

[33] Winfried Denk, James Strickler, Watt Webb, et al. "Two-photon laser scanning fluorescence microscopy". In: *Science* 248.4951 (1990), pp. 73–76.

[34] Edward R Dowski and W Thomas Cathey. "Extended depth of field through wavefront coding". In: *Applied Optics* 34.11 (Apr. 1995), pp. 1859–1866. DOI: 10.1364/AO.34.001859. URL: http://ao.osa.org/abstract.cfm?URI=ao-34-11-1859.

[35] Marco F Duarte, Mark A Davenport, Dharmpal Takbar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. "Single-pixel imaging via compressive sampling". In: *IEEE signal processing magazine* 25.2 (2008), pp. 83–91.

[36] Eitan Edrei and Giuliano Scarcelli. "Memory-effect based deconvolution microscopy for super-resolution imaging through scattering media". In: *Scientific Reports* 6 (2016).

[37] Christoph J Engelbrecht, Fabian Voigt, and Fritjof Helmchen. "Miniaturized selective plane illumination microscopy for high-contrast in vivo fluorescence imaging". In: *Optics Letters* 35.9 (2010), pp. 1413–1415.

[38] H M L Faulkner and J M Rodenburg. "Movable aperture lensless transmission microscopy: a novel phase retrieval algorithm". In: *Physical Review Letters* 93.2 (2004), p. 23903.

[39] Shechao Feng, Charles Kane, Patrick A Lee, and A Douglas Stone. "Correlations and fluctuations of coherent wave transmission through disordered media". In: *Physical review letters* 61.7 (1988), p. 834.

[40] Rob Fergus, Antonio Torralba, and William T Freeman. *Random Lens Imaging*. Tech. rep. Massachusetts Institute of Technology, 2006. URL: http://hdl.handle.net/1721.1/33962.

[41] Juliet Fiss, Brian Curless, and Richard Szeliski. "Refocusing plenoptic images using depth-adaptive splatting". In: *Computational Photography (ICCP), 2014 IEEE International Conference on*. IEEE. 2014, pp. 1–9.

[42] Ralf C Flicker and François J Rigaut. "Anisoplanatic deconvolution of adaptive optics images". In: *JOSA A* 22.3 (2005), pp. 504–513.

[43] David Chin-Lung Fong and Michael Saunders. " LSMR: An iterative algorithm for sparse least-squares problems". In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2950–2971.

[44] Todor Georgiev, Ke Colin Zheng, Brian Curless, David Salesin, Shree Nayar, and Chintan Intwala. "Spatio-Angular Resolution Tradeoffs in Integral Photography." In: *Rendering Techniques* 2006 (2006), pp. 263–272.

[45] Kunal K Ghosh, Laurie D Burns, Eric D Cocker, Axel Nimmerjahn, Yaniv Ziv, Abbas El Gamal, and Mark J Schnitzer. "Miniaturized integration of a fluorescence microscope". In: *Nature Methods* 8.10 (2011), p. 871.

[46] Patrick R Gill, James Tringali, Alex Schneider, Salman Kabir, David G Stork, Evan Erickson, and Mark Kellam. "Thermal Escher Sensors: Pixel-efficient Lensless Imagers Based on Tiled Optics". In: *Computational Optical Sensing and Imaging*. Optical Society of America. 2017, CTu3B–3.

[47] Joseph W Goodman. *Introduction to Fourier optics*. Roberts and Company Publishers, 2005.

[48] Joseph W Goodman. "Statistical properties of laser speckle patterns". In: *Laser speckle and related phenomena*. Springer, 1975, pp. 9–75.

[49] Andres de Groot, Bastijn J G van den Boom, Romano M van Genderen, Joris Coppens, John van Veldhuijzen, Joop Bos, Hugo Hoedemaker, Mario Negrello, Ingo Willuhn, Chris I De Zeeuw, et al. "NINscope, a versatile miniscope for multi-region circuit investigations". In: *eLife* 9 (2020), e49987.

[50] J Gu, Y Hitomi, T Mitsunaga, and S Nayar. "Coded rolling shutter photography: Flexible space-time sampling". In: *IEEE International Conference on Computational Photography (ICCP)*. Mar. 2010, pp. 1–8.

[51] Changliang Guo, Wenhao Liu, Xuanwen Hua, Haoyu Li, and Shu Jia. "Fourier light-field microscopy". In: *Optics Express* 27.18 (2019), pp. 25573–25594.

[52] T E Gureyev, A Pogany, D M Paganin, and S W Wilkins. "Linear algorithms for phase retrieval in the Fresnel region". In: *Optics Communications* 231 (2004), pp. 53–70. ISSN: 0030-4018. DOI: 10.1016/j.optcom.2003.12.020. URL: http://www.sciencedirect.com/science/article/pii/S0030401803023320.

[53] Walter Harm, Clemens Roider, Alexander Jesacher, Stefan Bernet, and Monika Ritsch-Marte. "Lensless imaging through thin diffusive media". In: *Optics Express* 22.18 (2014), pp. 22146–22156.

[54] Zachary T Harmany, Roummel F Marcia, and Rebecca M Willett. "Spatio-temporal compressed sensing with coded apertures and keyed exposures". In: *arXiv preprint arXiv:1111.7247* (2011).

[55] Fritjof Helmchen, Michale S Fee, David W Tank, and Winfried Denk. "A miniature head-mounted two-photon microscope: high-resolution brain imaging in freely moving animals". In: *Neuron* 31.6 (2001), pp. 903–912.

[56] Michele Hirsch, Sriram Sivaramakrishnan, Suhada Jayasuriya, Aiping Wang, Adrienn Molnar, Ramesh Raskar, and Gordon Wetzstein. "A switchable light field camera architecture with Angle Sensitive Pixels and dictionary-based sparse coding". In: *Computational Photography (ICCP), 2014 IEEE International Conference on*. IEEE. 2014, pp. 1–10.

[57] Terrence F Holekamp, Diwakar Turaga, and Timothy E Holy. "Fast three-dimensional fluorescence imaging of activity in neural populations by objective-coupled planar illumination microscopy". In: *Neuron* 57.5 (2008), pp. 661–672.

[58] Ryoichi Horisaki, Satoru Irie, Yusuke Ogura, and Jun Tanida. "Three-Dimensional Information Acquisition Using a Compound Imaging System". In: *Optical Review* 14.5 (2007), pp. 347–350. ISSN: 1349-9432. DOI: 10.1007/s10043-007-0347-z. URL: http://dx.doi.org/10.1007/s10043-007-0347-z.

[59] Frederic E Ives. *Parallax stereogram and process of making same.* 1903.

[60] J. Liang, L. Zhu, and L V Wang. "Single-shot real-time femtosecond imaging of temporal focusing". In: *Light: Science & Applications* 7.1 (2018), p. 42.

[61] Alexander D Jacob, Adam I Ramsaran, Andrew J Mocle, Lina M Tran, Chen Yan, Paul W Frankland, and Sheena A Josselyn. "A Compact Head-Mounted Endoscope for In Vivo Calcium Imaging in Freely Behaving Mice". In: *Current Protocols in Neuroscience* 84.1 (2018), e51.

[62] Zhong Jingshan, Rene A Claus, Justin Dauwels, Lei Tian, and Laura Waller. "Transport of Intensity phase imaging by intensity spectrum fitting of exponentially spaced defocus planes". In: *Optics Express* 22.9 (May 2014), pp. 10661–10674. DOI: 10.1364/OE.22.010661. URL: http://www.opticsexpress.org/abstract.cfm?URI=oe-22-9-10661.

[63] K.Tajima, T Shimano, Y Nakamura, M Sao, and T Hoshizawa. "Lensless light-field imaging with multi-phased fresnel zone aperture". In: *2017 IEEE International Conference on Computational Photography (ICCP)*. May 2017, pp. 76–82.

[64] Mahdad Hosseini Kamal, Barmak Heshmat, Ramesh Raskar, Pierre Vandergheynst, and Gordon Wetzstein. "Tensor low-rank and sparse light field photography". In: *Computer Vision and Image Understanding* 145 (2016), pp. 172–181. ISSN: 1077-3142. DOI: http://dx.doi.org/10.1016/j.cviu.2015.11.004. URL: http://www.sciencedirect.com/science/article/pii/S1077314215002465.

[65] Ulugbek S Kamilov. "A parallel proximal algorithm for anisotropic total variation minimization". In: *IEEE Transactions on Image Processing* 26.2 (2016), pp. 539–548.

[66] Ulugbek S Kamilov. "A parallel proximal algorithm for anisotropic total variation minimization". In: *IEEE Transactions on Image Processing* 26.2 (2017), pp. 539–548.

[67] Yuval Kashter, A Vijayakumar, and Joseph Rosen. "Resolving images by blurring: superresolution method with a scattering mask between the observed objects and the hologram recorder". In: *Optica* 4.8 (Aug. 2017), pp. 932–939. DOI: 10.1364/OPTICA.4.000932. URL: http://www.osapublishing.org/optica/abstract.cfm?URI=optica-4-8-932.

[68] Ori Katz, Pierre Heidmann, Mathias Fink, and Sylvain Gigan. "Non-invasive single-shot imaging through scattering layers and around corners via speckle correlations". In: *Nature Photonics* 8.10 (2014), pp. 784–790.

[69] Roman Koller, Lukas Schmid, Nathan Matsuda, Thomas Niederberger, Leonidas Spinoulas, Oliver Cossairt, Guido Schuster, and Aggelos K Katsaggelos. "High spatio-temporal resolution video with compressed sensing". In: *Optics Express* 23.12 (June 2015), pp. 15992–16007.

[70] F Krahmer, S Mendelson, and H Rauhut. "Suprema of Chaos Processes and the Restricted Isometry Property". In: *Commun. Pur. Appl. Math.* 67.11 (2014), pp. 1877–1904.

[71] Grace Kuo, Nick Antipa, Ren Ng, and Laura Waller. " 3D fluorescence microscopy with diffusercam". In: *Computational Optical Sensing and Imaging*. Optical Society of America. 2018, CM3E–3.

[72] Grace Kuo, Nick Antipa, Ren Ng, and Laura Waller. "DiffuserCam: Diffuser-Based Lensless Cameras". In: *Computational Optical Sensing and Imaging*. Optical Society of America. 2017, CTu3B–2.

[73] Grace Kuo, Fanglin Linda Liu, Irene Grossrubatscher, Ren Ng, and Laura Waller. "On-chip fluorescence microscopy with a random microlens diffuser". In: *Optics Express* 28.6 (2020), pp. 8384–8399.

[74] L. Gao, J. Liang, C. Li, and L W Wang. "Single-shot compressed ultrafast photography at one hundred billion frames per second". In: *Nature* 516.7529 (2014), pp. 74–77.

[75] KyeoReh Lee and YongKeun Park. "Exploiting the speckle-correlation scattering matrix for a compact reference-free holographic image sensor". In: *Nature Communications* 7 (2016).

[76] Marc Levoy, Ren Ng, Andrew Adams, Matthew Footer, and Mark Horowitz. "Light Field Microscopy". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 25.3 (2006).

[77] C Liang, L Chang, and H H Chen. "Analysis and Compensation of Rolling Shutter Effect". In: *IEEE Transactions on Image Processing* 17.8 (Aug. 2008), pp. 1323–1330.

[78] Chia-Kai Liang, Tai-Hsu Lin, Bing-Yi Wong, Chi Liu, and Homer H Chen. "Programmable aperture photography: multiplexed light field acquisition". In: *ACM Transactions on Graphics (TOG)* 27.3 (2008), p. 55.

[79] Chia-Kai Liang and Ravi Ramamoorthi. "A light transport framework for lenslet light field cameras". In: *ACM Transactions on Graphics (TOG)* 34.2 (2015), p. 16.

[80] William A Liberti III, L Nathan Perkins, Daniel P Leman, and Timothy J Gardner. "An open source, wireless capable miniature microscope system". In: *Journal of Neural Engineering* 14.4 (2017), p. 45001.

[81] Gabriel Lippmann. "La photographie intégrale". In: *Comptes-Rendus, Académie des Sciences* 146 (1908), pp. 446–551.

[82] Fanglin Linda Liu, Vaishnavi Madhavan, Nick Antipa, Grace Kuo, Saul Kato, and Laura Waller. "Single-shot 3D fluorescence microscopy with Fourier DiffuserCam". In: *Novel Techniques in Microscopy*. Optical Society of America. 2019, NS2B–3.

[83] Hsiou-Yuan Liu, Eric Jonas, Lei Tian, Jingshan Zhong, Benjamin Recht, and Laura Waller. "3D imaging in volumetric scattering media using phase-space measurements". In: *Opt. Express* 23.11 (June 2015), pp. 14461–14471. DOI: 10.1364/OE.23.014461. URL: http://www.opticsexpress.org/abstract.cfm?URI=oe-23-11-14461.

[84] Hsiou-Yuan Liu, Eric Jonas, Lei Tian, Jingshan Zhong, Benjamin Recht, and Laura Waller. " 3D imaging in volumetric scattering media using phase-space measurements". In: *Optics Express* 23.11 (2015), pp. 14461–14471.

[85] Xianglei Liu, Jingdan Liu, Cheng Jiang, Fiorenzo Vetrone, and Jinyang Liang. "Single-shot compressed optical-streaking ultra-high-speed photography". In: *Optics letters* 44.6 (2019), pp. 1387–1390.

[86] Y Liu, X Yuan, J Suo, D Brady, and Q Dai. "Rank Minimization for Snapshot Compressive Imaging". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018), p. 1. ISSN: 0162-8828.

[87] Antoine Liutkus, David Martina, Sébastien Popoff, Gilles Chardon, Ori Katz, Geoffroy Lerosey, Sylvain Gigan, Laurent Daudet, and Igor Carron. "Imaging with nature: Compressive imaging using a multiply scattering medium". In: *Scientific Reports* 4 (2014).

[88] A Llavador, J Sola-Pikabea, G Saavedra, B Javidi, and M Martinez-Corral. "Resolution improvements in integral microscopy with Fourier plane recording". In: *Optics express* 24.18 (2016), pp. 20792–20798.

[89] Patrick Llull, Xuejun Liao, Xin Yuan, Jianbo Yang, David Kittle, Lawrence Carin, Guillermo Sapiro, and David J Brady. "Coded aperture compressive temporal imaging". In: *Optics express* 21.9 (2013), pp. 10526–10545.

[90] Luminit. *Technical Data and Downloads*. \url http://www.luminitco.com/downloads/datasheets. 2017.

[91] Andrew Lumsdaine and Todor Georgiev. "The focused plenoptic camera". In: *Computational Photography (ICCP), 2009 IEEE International Conference on*. IEEE. 2009, pp. 1–8.

[92] M Lustig, D L Donoho, J M Santos, and J M Pauly. "Compressed Sensing MRI". In: *IEEE Signal Processing Magazine* 25.2 (Mar. 2008), pp. 72–82. ISSN: 1053-5888. DOI: 10.1109/MSP.2007.914728.

[93] A M Maiden, G R Morrison, B Kaulich, A Gianoncelli, and J M Rodenburg. "Soft X-ray spectromicroscopy using ptychography with randomly phased illumination". In: *Nature communications* 4 (2013), p. 1669.

[94] K Marwah, G Wetzstein, Y Bando, and R Raskar. "Compressive Light Field Photography using Overcomplete Dictionaries and Optimized Projections". In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 32.4 (2013), pp. 1–11.

[95] A Matakos, S Ramani, and J A Fessler. "Accelerated edge-preserving image restoration without boundary artifacts". In: *IEEE Transactions on Image Processing* 22.5 (2013), pp. 2019–2029.

[96] Takeshi Naemura, T Yoshida, and H Harashima. "3-D computer graphics based on integral photography". In: *Optics Express* 8.4 (2001), pp. 255–262.

[97] Tomoya Nakamura, Keiichiro Kagawa, Shiho Torashima, and Masahiro Yamaguchi. "Super Field-of-View Lensless Camera by Coded Image Sensors". In: *Sensors* 19.6 (2019), p. 1329.

[98] Ren Ng, Marc Levoy, Mathieu Bredif, Gene Duval†, Mark Horowitz, and Pat Hanrahan. "Light Field Photography with a Hand-held Plenoptic Camera". In: *Stanford University Computer Science Tech Report* (Apr. 2005), pp. 3418–3421. URL: `https://graphics.stanford.edu/papers/lfcamera/lfcamera-150dpi.pdf`.

[99] Tobias Nöbauer, Oliver Skocek, Alejandro J Pernia-Andrade, Lukas Weilguny, Francisca Martinez Traub, Maxim I Molodtsov, and Alipasha Vaziri. "Video rate volumetric Ca 2+ imaging across cortex using seeded iterative demixing (SID) microscopy". In: *Nature Methods* 14.8 (2017), p. 811.

[100] J Nocedal and S J Wright. *Numerical Optimization*. Springer, 2006.

[101] Brendan O'Donoghue and Emmanuel Candes. "Adaptive restart for accelerated gradient schemes". In: *Foundations of computational mathematics* 15.3 (2015), pp. 715–732.

[102] Yusuke Oike and Abbas El Gamal. "A 256× 256 CMOS image sensor with ΔΣ-based single-shot compressed sensing". In: *2012 IEEE International Solid-State Circuits Conference*. IEEE. 2012, pp. 386–388.

[103] Fumio Okano, Jun Arai, Haruo Hoshino, and Ichiro Yuyama. "Three-dimensional video system based on integral photography". In: *Optical Engineering* 38.6 (1999), pp. 1072–1077.

[104] Sri Rama Prasanna Pavani and Rafael Piestun. "Three dimensional tracking of fluorescent microparticles using a photon-limited double-helix response system". In: *Optics Express* 16.26 (2008), pp. 22048–22057.

[105] Nicolas C Pégard, Hsiou-Yuan Liu, Nick Antipa, Maximillian Gerlock, Hillel Adesnik, and Laura Waller. "Compressive light-field microscopy for 3D neural activity recording". In: *Optica* 3.5 (2016), pp. 517–524.

[106] Eftychios A Pnevmatikakis and Andrea Giovannucci. "NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data". In: *Journal of Neuroscience Methods* 291 (2017), pp. 83–94.

[107] Jonathan Ragan-Kelley, Connelly Barnes, Andrew Adams, Sylvain Paris, Frédo Durand, and Saman Amarasinghe. "Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines". In: *ACM SIGPLAN Notices* 48.6 (2013), pp. 519–530.

[108] Justin Romberg. "Compressive sensing by random convolution". In: *SIAM Journal on Imaging Sciences* 2.4 (2009), pp. 1098–1128. ISSN: 19364954. DOI: 10.1137/08072975X.

[109] Leonid I Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: Nonlinear Phenomena* 60.1-4 (1992), pp. 259–268.

[110] S. Su and W Heidrich. "Rolling Shutter Motion Deblurring". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[111] Sam Dehaeck. *TipSlicer*. \url (https://github.com/SamDehaeck/TipSlicer).

[112] Olivier Saurer, Kevin Koser, Jean-Yves Bouguet, and Marc Pollefeys. "Rolling shutter stereo". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 465–472.

[113] Yoav Y Schechner and Shree K Nayar. "Generalized mosaicing". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 1. IEEE. 2001, pp. 17–24.

[114] G Scrofani, Jorge Sola-Pikabea, A Llavador, Emilio Sanchez-Ortiga, JC Barreiro, G Saavedra, J Garcia-Sucerquia, and Manuel Martinez-Corral. "FIMic: design for ultimate 3D-integral microscopy of in-vivo biological samples". In: *Biomedical optics express* 9.1 (2018), pp. 335–346.

[115] Mark Sheinin, Yoav Y Schechner, and Kiriakos N Kutulakos. "Rolling shutter imaging on the electric grid". In: *2018 IEEE International Conference on Computational Photography, ICCP 2018, Pittsburgh, PA, USA, May 4-6, 2018*. IEEE Computer Society, 2018, pp. 1–12. ISBN: 978-1-5386-2526-2. DOI: 10.1109/ICCPHOT.2018.8368472. URL: http://doi.ieeecomputersociety.org/10.1109/ICCPHOT.2018.8368472.

[116] Jaewook Shin, Dung N Tran, Jasper R Stroud, Sang Chin, Trac D Tran, and Mark A Foster. "A minimally invasive lens-free computational microendoscope". In: *Science Advances* 5.12 (2019), eaaw5595.

[117] Alok Singh, Dinesh Naik, Giancarlo Pedrini, Mitsuo Takeda, and Wolfgang Osten. "Exploiting scattering media for exploring 3 D objects". In: *Light: Science & Applications* 6.2 (2017), e16219.

[118] Alok Singh, Dinesh Naik, Giancarlo Pedrini, Mitsuo Takeda, and Wolfgang Osten. "Looking through a diffuser and around an opaque surface: A holographic approach". In: *Optics Express* 22.7 (2014), pp. 7694–7701.

[119] Alok Singh, Giancarlo Pedrini, Mitsuo Takeda, and Wolfgang Osten. "Scatter-plate microscope for lensless microscopy with diffraction limited resolution". In: *Scientific Reports* 7.1 (2017), p. 10687.

[120] Ayan Sinha, Justin Lee, Shuai Li, and George Barbastathis. "Lensless computational imaging through deep learning". In: *Optica* 4.9 (Sept. 2017), pp. 1117–1125. DOI: 10.1364/OPTICA.4.001117. URL: http://www.osapublishing.org/optica/abstract.cfm?URI=optica-4-9-1117.

[121] Oliver Skocek, Tobias Nöbauer, Lukas Weilguny, Francisca Mart´ Traub, Chuying Naomi Xia, Maxim I Molodtsov, Abhinav Grama, Masahito Yamagata, Daniel Aharoni, David D Cox, et al. "High-speed volumetric imaging of neuronal activity in freely moving rodents". In: *Nature Methods* (2018), p. 1.

[122] D G Stork and P R Gill. "Optical, mathematical, and computational foundations of lensless ultra-miniature diffractive imagers and sensors". In: *International Journal on Advances in Systems and Measurements* 7.3 (2014), p. 4.

[123] David G Stork and Patrick R Gill. "Optical, mathematical, and computational foundations of lensless ultra-miniature diffractive imagers and sensors". In: *International Journal on Advances in Systems and Measurements* 7.3 (2014), p. 4.

[124] Abigail Stylianou and Robert Pless. "SparkleGeometry: Glitter Imaging for 3 D Point Tracking". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 10–17.

[125] Jun Tanida, Tomoya Kumagai, Kenji Yamada, Shigehiro Miyatake, Kouichi Ishida, Takashi Morimoto, Noriyuki Kondou, Daisuke Miyazaki, and Yoshiki Ichioka. "Thin observation module by bound optics: concept and experimental verification". In: *Applied Optics* 40.11 (2001), pp. 1806–1813.

[126] Jun Tanida, Tomoya Kumagai, Kenji Yamada, Shigehiro Miyatake, Kouichi Ishida, Takashi Morimoto, Noriyuki Kondou, Daisuke Miyazaki, and Yoshiki Ichioka. "Thin observation module by bound optics: concept and experimental verification". In: *Applied Optics* 40.11 (2001), pp. 1806–1813.

[127] Michael Teague. "Deterministic phase retrieval: a Green's function solution". In: *Journal of the Optical Society of America* 73.11 (Nov. 1983), pp. 1434–1441. DOI: 10.1364/JOSA.73.001434. URL: http://www.opticsinfobase.org/abstract.cfm?URI=josa-73-11-1434.

[128] S Thiele, K Arzenbacher, T Gissibl, H Giessen, and A M Herkommer. " 3D-printed eagle eye: Compound microlens system for foveated imaging". In: *Science Advances* 3.2 (2017), e1602655.

[129] Lei Tian, Jonathan C Petruccelli, Qin Miao, Haris Kudrolli, Vivek Nagarkar, and George Barbastathis. "Compressive x-ray phase tomography based on the transport of intensity equation". In: *Opt. Lett.* 38.17 (Sept. 2013), pp. 3418–3421. DOI: 10.1364/OL.38.003418. URL: http://ol.osa.org/abstract.cfm?URI=ol-38-17-3418.

[130] Lei Tian, Jingyan Wang, and Laura Waller. "3D differential phase-contrast microscopy with computational illumination using an LED array". In: *Optics Letters* 39.5 (2014), pp. 1326–1329.

[131] UCLA. *Miniscope*. url:(https://miniscope.org).

[132] A Veeraraghavan, D Reddy, and R Raskar. "Coded Strobing Photography: Compressive Sensing of High Speed Periodic Videos". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.4 (Apr. 2011), pp. 671–686. ISSN: 0162-8828.

[133] Ashok Veeraraghavan, Ramesh Raskar, Amit Agrawal, Ankit Mohan, and Jack Tumblin. "Dappled Photography: Mask Enhanced Cameras for Heterodyned Light Fields and Coded Aperture Refocusing". In: *ACM Trans. Graph.* 26.3 (July 2007). ISSN: 0730-0301. DOI: 10.1145/1276377.1276463. URL: http://doi.acm.org/10.1145/1276377.1276463.

[134] Kartik Venkataraman, Dan Lelescu, Jacques Duparré, Andrew McMahon, Gabriel Molina, Priyam Chatterjee, Robert Mullis, and Shree Nayar. "PiCam: An ultra-thin high performance monolithic camera array". In: *ACM Transactions on Graphics (TOG)* 32.6 (2013), p. 166.

[135] Bo Wahlberg, Stephen Boyd, Mariette Annergren, and Yang Wang. "An ADMM algorithm for a class of total variation regularized estimation problems". In: *IFAC Proceedings Volumes* 45.16 (2012), pp. 83–88. ISSN: 1474-6670.

[136] Albert Wang, Patrick R Gill, and Alyosha Molnar. "An angle-sensitive CMOS imager for single-sensor 3D photography". In: *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*. IEEE. 2011, pp. 412–414.

[137] Y Wang, J Yang, W Yin, and Y Zhang. "A New Alternating Minimization Algorithm for Total Variation Image Reconstruction". In: *SIAM Journal on Imaging Sciences* 1.3 (2008), pp. 248–272.

[138] Mian Wei, Navid Sarhangnejad, Zhengfan Xia, Nikita Gusev, Nikola Katic, Roman Genov, and Kiriakos N Kutulakos. "Coded Two-Bucket Cameras for Computer Vision". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 54–71.

[139] Li-Yi Wei, Chia-Kai Liang, Graham Myhre, Colvin Pitts, and Kurt Akeley. "Improving light field camera sample design with irregularity and aberration". In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), p. 152.

[140] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. "High performance imaging using large camera arrays". In: *ACM Transactions on Graphics (TOG)* 24.3 (2005), pp. 765–776.

[141] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S K Nayar. "Video from a single coded exposure photograph using a learned over-complete dictionary". In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2011, pp. 287–294.

[142] Kyrollos Yanny, Nick Antipa, Ren Ng, and Laura Waller. "Miniature 3D Fluorescence Microscope Using Random Microlenses". In: *Optics and the Brain*. Optical Society of America. 2019, BT3A–4.

[143] Xin Yuan, Patrick Llull, Xuejun Liao, Jianbo Yang, David J Brady, Guillermo Sapiro, and Lawrence Carin. "Low-cost compressive sensing for color video and depth". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3318–3325.

[144] Zhengyun Zhang and M Levoy. "Wigner distributions and how they relate to the light field". In: *Computational Photography (ICCP), 2009 IEEE International Conference on*. Apr. 2009, pp. 1–10. DOI: 10.1109/ICCPHOT.2009.5559007.

[145] Weijian Zong, Runlong Wu, Mingli Li, Yanhui Hu, Yijun Li, Jinghang Li, Hao Rong, Haitao Wu, Yangyang Xu, Yang Lu, et al. "Fast high-resolution miniature two-photon microscopy for brain imaging in freely behaving mice". In: *Nature Methods* 14.7 (2017), p. 713.