Supplemental Material for SpinCam: High-Speed Imaging via a Rotating Point-Spread Function

Dorian Chan, Mark Sheinin, and Matthew O'Toole
Carnegie Mellon University, Pittsburgh, PA 15213, USA
dychan@andrew.cmu.edu, marksheinin@gmail.com, mpotoole@cmu.edu

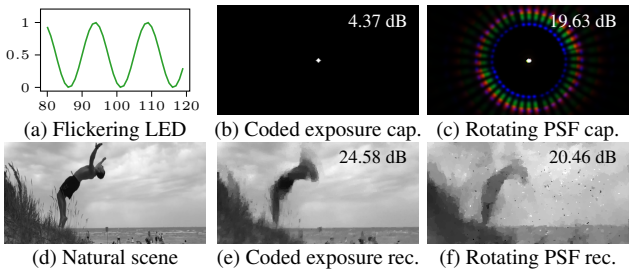
1. Introduction

We refer the reader to "index.html" for viewing the supplemental website, which includes videos of both results from the main paper as well as additional results, and contains the parameters for each experiment. We also include a DIY guide as part of the website. We additionally share our reconstruction code in "code/recover.ipynb". In Sec. 2, we start by exploring some of the key differences between coded exposure and our approach. In Sec. 3, we dive into an analysis of the space of potential PSFs for a rotating setup from the point-of-view of mutual coherence. In the rest of this document, we discuss the details of our real setup in more depth. In Sec. 4, we verify the reconstruction resolution of our system. In Sec. 5, we provide more information on our calibration and capture process. We use the color blue to refer figures and equations from the main paper.

2. Relationship to Coded Exposure

As we noted in the main paper, the convolution between the PSF and the scene at time t can be expressed in the Fourier domain as the elementwise multiplication of their corresponding Fourier transforms. The final computed image is the integration of all of these multiplications. In other words, a coded exposure process is performed on the Fourier transform of the image of the scene. Why might our approach then be useful when compared to traditional coded exposure?

To attempt to answer this question, we consider the simple case of a single bump binary shutter, a restriction often utilized in previous work for ease of practical implementation [5]. Under this scenario, at each time instant, a subset of the pixels of the scene will be captured under coded exposure, while a subset of the frequencies will be captured under our approach. As a result, scenes that are redundant in frequency are better captured by a time-varying PSF, while scenes that are redundant in space are better captured by a coded exposure. For example, a coded exposure setup may have a hard time resolving a spatially sparse scene consisting of a single point, while a time-varying PSF would easily resolve it—we illustrate this phenomenon in Fig. 1 in the case of a single flashing point. Conversely, a time-varying PSF would struggle to capture a scene consisting of a single spatial frequency, while a coded exposure approach would have no such problems. We note that many applications of high-speed imaging fall into the spatially sparse regime, which we demonstrate in the main paper and is argued by past work [14, 13].

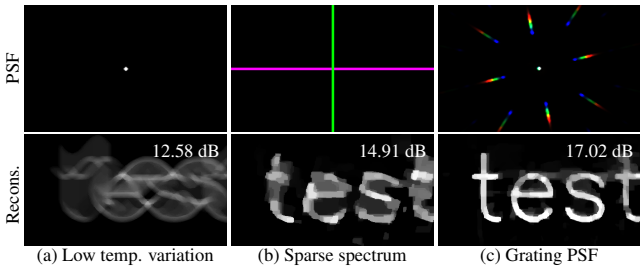


(a) Flickering LED (b) Coded exposure cap. (c) Rotating PSF cap.
(d) Natural scene (e) Coded exposure rec. (f) Rotating PSF rec.

Figure 1. **Simulated comparison with coded exposure**. For coded exposure, we implemented the shutter function of Hitomi *et al*. [5] with the same simple sparsity and variation priors as in Eq. (4). **(a)** A flickering point source. **(b)** The resulting coded-exposure image fails to capture the flicker dynamics due to the scene's spatial sparsity—all of temporal signal is captured in a single measurement. **(c)** The rotating PSF captures the flicker dynamics, as seen in the circle around the central points. **(d-f)** In dense scenes, coded exposure can outperform our method, especially under noise.

In general, generic scenes fall into neither category. Currently, as shown in the second row of Fig. 1, our method performs somewhat worse than coded exposure techniques on natural dense scenes. However, by leveraging strong priors like dictionary learning or machine learning like those applied in the coded exposure literature [9, 6, 10], it is likely the results of using a time-varying PSF can be significantly improved.

In addition, we note that in our current prototype system, the grating PSF codes the frequency spectrum with a general broadband pattern with few zeros (see Fig. 5). This allows our system to resolve more frequencies for every



(a) Low temp. variation (b) Sparse spectrum (c) Grating PSF

Figure 2. **Quantitative spatiotemporal resolution example**. The word "test" moves in a sinusoidal pattern. **(a)** A PSF that does not significantly vary with rotation results in poor temporal resolution & motion blur. **(b)** A PSF with more rotational variation offers better temporal resolution, but can result in spatial blur if not well designed. **(c)** A good PSF balances these two concerns.

time step, at the cost of having to solve a complex unmixing problem (Eq. (4)). In contrast, existing implementations of coded exposure techniques have been limited to binary patterns for ease of implementation [5, 9]. This extra degree of freedom provided by a PSF setup could be adapted to different target scenes. For example, for an extremely spatially sparse scene, a narrowband PSF with a sparse binary spectrum could be used to maximize temporal resolution, while a broadband PSF could be used for denser scenes to maximize spatial resolution.

As we mentioned in the main paper, traditional approaches for coded exposure are also limited in their temporal resolution and suffer from complex hardware. State-of-the-art approaches based on experimental sensors [7, 8, 9, 6, 11] or SLMs [5, 12] are limited by how quickly masks can be displayed by the corresponding hardware. SLM and piezoelectric stage-based approaches also require careful alignment and positioning in order to colocate the device with the sensor. In contrast, a rotating PSF setup can be easily implemented as shown by our DIY guide, just by placing a diffraction grating in front of the lens of a camera. Furthermore, our prototype can operate at 192,000 FPS on our off-the-shelf motor, and this rate can be increased just by speeding up the motor.

3. What is the Right PSF to Use?

In the main paper, we briefly touched on the spatiotemporal resolution provided by a few examples of different PSFs, and we show a quantitative example in Fig. 2. However, this begs the following important question—in general, what is the right PSF to use, such that the original high speed signal can be easily and efficiently recovered?

To explore this question, we turn to the compressed sensing literature. It is well known that the quantity known as the *mutual coherence* of a compressed sensing system [3, 4, 1], like that expressed in Eq. (4), limits the density of scenes that can be recovered. When the mutual coherence is high, only very sparse scenes can be recovered. When the mutual coherence is low, then much denser scenes can be reconstructed. Mathematically, assuming that the high speed video is spatiotemporally sparse, the mutual coherence μ of our system can be expressed as follows:

$$\mu(\mathbf{M}) = \max_{i \neq j, 1 \leq i,j \leq n} \frac{|m_i^T m_j|}{\|m_i\| \|m_j\|} \tag{1}$$

where $\mathbf{M}$ denotes the forward model of our rotating PSF in matrix form, n is the number of columns in $\mathbf{M}$, and m_i is the ith column of $\mathbf{M}$. This can also be written in matrix form:

$$\mu(\mathbf{M}) = \max_{i \neq j, 1 \leq i,j \leq n} |(\tilde{\mathbf{M}}^T \tilde{\mathbf{M}})_{i,j}| \tag{2}$$

where $\tilde{\mathbf{M}}$ is the column-normalized form of $\mathbf{M}$. In practice, μ merely constrains the *worst case* recovery of a compressed sensing system, and often does not reflect the real performance of a compressed sensing system [4]. As a result, the computation above is often relaxed in order to better reflect average performance [4, 1]—one particular form is given by:

$$\mu_{\text{avg}}(\mathbf{M}) = \frac{\sum_{i \neq j} |(\tilde{\mathbf{M}}^T \tilde{\mathbf{M}})_{i,j}|}{n(n-1)} \tag{3}$$

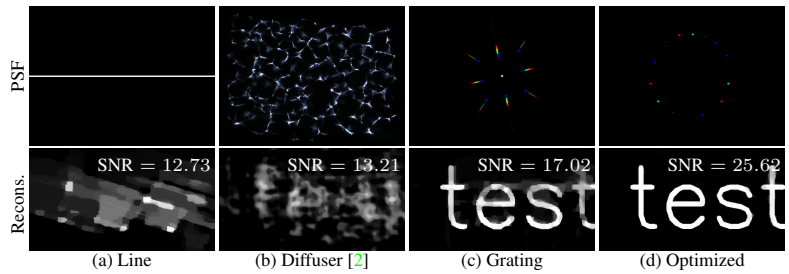
where an "average" mutual coherence is computed [4].

In the case of our imaging system, these expressions can be intuitively interpreted as measuring how unique the responses of different spatiotemporal points are. If each spatiotemporal scene point leaves very different responses compared to other points, the points will be easier to disambiguate and the mutual coherence is small. However, if they leave very similar responses, different points will be hard to separate and the mutual coherence is large.

For our imaging system, it turns out that under special conditions, these equations can be efficiently computed. In particular, note that $\tilde{\mathbf{M}}^T \tilde{\mathbf{M}}$ can be rewritten as:

$$\tilde{\mathbf{M}}^T \tilde{\mathbf{M}} = \begin{bmatrix} \tilde{\mathbf{C}}_1^T \\ \tilde{\mathbf{C}}_2^T \\ \cdots \\ \tilde{\mathbf{C}}_{N_E}^T \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{C}}_1 & \tilde{\mathbf{C}}_2 & \cdots & \tilde{\mathbf{C}}_{N_E} \end{bmatrix} \tag{4}$$

where $\tilde{\mathbf{C}}_i$ is the normalized version of $\mathbf{C}_i$, which denoted the convolution with the PSF at timestep i in Eq. (3). Therefore, row i of $\tilde{\mathbf{M}}^T \tilde{\mathbf{M}}$ can be efficiently computed by taking the response of spatiotemporal pixel i, and cross-correlating that function with the PSF at every timestep. Furthermore, if we ignore boundary conditions and assume that the responses of all pixels at some timestep t are identical, this computation can be reused for all pixels from time t, and the entries of $\tilde{\mathbf{M}}^T \tilde{\mathbf{M}}$ can be approximately computed with just N_E^2 convolutions. Instead of computing these metrics



(a) Line (b) Diffuser [2] (c) Grating (d) Optimized

Computed coherence values				
μ	0.9950	**0.9140**	0.9537	0.8763
μ_{avg}	0.0023	0.0370	**0.0007**	0.0001
μ_{spatial}	0.9950	0.8211	**0.6605**	0.5237
$\mu_{\text{avg, spatial}}$	0.0023	0.0370	**0.0006**	0.0001
μ_{temporal}	**0.8302**	0.9140	0.9537	0.8763
$\mu_{\text{avg, temporal}}$	**0.0481**	0.2340	0.1527	0.0242
	(a) Line	(b) Diffuser [2]	(c) Grating	(d) Optimized

Figure 3. **Evaluating real potential PSFs with mutual coherence measures**. **(a)** A line PSF, like that generated by a cylindrical lens, has poor spatial mutual coherence values, resulting in shoddy reconstructions. **(b)** The PSF from a diffuser [2] has better spatial mutual coherence but poor average-case coherence, once again resulting in poor results. **(c)** The PSF created by a dual-axis diffraction grating has slightly worse temporal mutual coherence, but better average case coherence in both the spatial and temporal cases, resulting in better reconstructions. **(d)** A PSF generated by an optimization procedure [1] results in the best coherence scores and the best reconstructions, but creating such a PSF is not easily physically realizable.

over all the spatiotemporal points, we can also restrict the computations to just points from the same timestamps but different spatial locations to get a sense of the spatial resolution, which we denote as μ_{spatial} and $\mu_{\text{avg, spatial}}$. We can apply the same procedure for points from the same spatial location but different timestamps to get a sense of the temporal resolution, which we term μ_{temporal} and $\mu_{\text{avg, temporal}}$.

With these expressions in hand, we can apply them to possible PSFs we could use for our real world prototype. We show sample reconstructions along with the estimated numbers in Fig. 3, which we compute with a PSF resolution of 150×200 pixels and 150 timesteps over a 90° rotation. A line PSF like that generated by a cylindrical lens, as expected, has very poor mutual coherence, resulting in poor reconstructions with poor SNR. It has strong temporal resolution, as shown by the temporal coherence values, but poor spatial resolution as shown by the high spatial coherence numbers. A diffuser PSF, like the one used by Antipa *et al*. [2], has better mutual coherence, but poor average mutual coherence, again resulting in poor reconstructions. A grating PSF has slightly worse mutual coherence but much better average mutual coherence, resulting in higher quality reconstructions. While its temporal coherence does not match that of the line PSF, the reconstructions are visually improved thanks to the better spatial resolution. We note that when computing the coherence values for the grating, we clipped the brightness of the central DC spot to the maximum brightness of the color streaks, to better match real world usage where the central DC spot is allowed to be overexposed to better expose the rest of the PSF. Finally, we can also optimize for a PSF with a low mutual coherence—we follow the approach of Abolghasemi *et al*. [1] and minimize the related expression $|\mathbf{I} - \tilde{\mathbf{M}}^T\tilde{\mathbf{M}}|_F$ using gradient descent. This optimized PSF, though not easily physically realizable, provides both the best coherence scores as well as the best reconstructions.

To end this section, we emphasize that the above findings are heavily dependent on the priors utilized for the reconstruction process. The above derivations focused on the specific case of spatiotemporal sparsity, and the PSF that minimizes the mutual coherence may look very different if another prior is used. For example, if the scene is sparse under some other basis $\mathbf{B}$, the mutual coherence would be instead computed using the matrix $\mathbf{G} = \mathbf{MB}$. Furthermore, while μ possesses strong theoretical guarantees on the sparsity of scenes that can be recovered, the average case μ_{avg} lacks the same mathematical rigor, and our approximations for computing $\tilde{\mathbf{M}}^T\tilde{\mathbf{M}}$ introduce further uncertainty. Appropriately, the results in this section should then be used as a general guideline for the right PSF rather than a definitive rulebook.

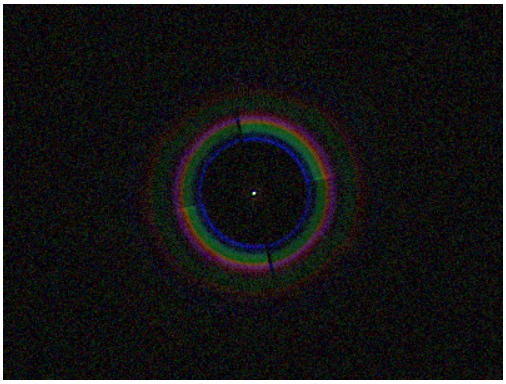
4. Reconstruction Resolution

In our results, we reconstructed 146 frames from each 300×400 pixel image. We verified that our system can adequately resolve such temporal resolution in Fig. 4 in the presence of shot and read noise. It is likely that with more GPU memory, larger images can be input into our reconstruction method and even higher temporal resolution can be resolved.

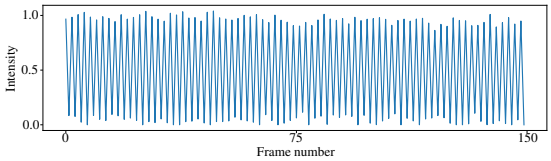
5. Calibration and Capture

In order to apply Eq. (4), we need to identify the time-varying point-spread function of our system. To simplify this process, we assume that the PSF varies purely as a function of rotation, and there is no distortion or otherwise non-ideal effects that might be caused in deformations or misalignment in the grating. Secondly, we assume that the PSF rotates at a constant rate over the camera exposure.

With these two simple assumptions, for a particular capture, we then just need to know the stationary PSF $\mathbf{k}$ of our system, and the range of angles that are rotated over during the camera exposure. We examine each element in more



(a) Noisy source image



(b) Recovered signal

Figure 4. Reconstructing a simulated 150 frame video encoded in a 90° rotation of the point-spread function in a 300×400 pixel image. Our setup is able to resolve a point source flashing every other frame. Here, we set $\lambda_{sparsity} = 1e-7$, $\lambda_{dx} = \lambda_{dy} = 0$, and $\lambda_{dt} = 0$.

detail in the rest of this section, as well as potential future hardware implementations.

5.1. Calibrating the stationary PSF

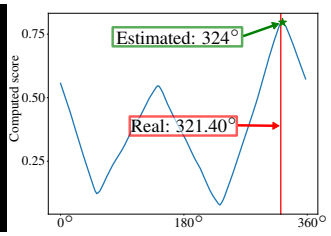
To calibrate $\mathbf{k}$, we place a small sample of the target material in front of a black background in the field-of-view of the optical system, and capture an image with the grating at a single fixed orientation. This image can be used directly for the PSF, but a sharper PSF can potentially be created in postprocessing. In short, we create a DC-only version of this image by cropping out the rainbow streaks created by the diffraction grating, and then perform a sparse deconvolution between the DC-only image and the original captured image to recover a sharp PSF:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \quad \|\mathbf{I}_{capture} - \mathbf{I}_{dc} * \mathbf{k}\|^2 + \mathcal{I}_{\mathbb{R}_+}(\mathbf{k}) + \lambda_{sparsity}\|\mathbf{k}\|_1 \tag{5}$$

where $\mathcal{I}_{\mathbb{R}_+}$ denotes a non-negativity prior, and $\lambda_{sparsity}$ weights a sparsity regularization term. Instead of using a DC-only image, one could also capture an image of the scene with and without the grating present — however, we found this process was difficult without significantly disturbing the optical setup.

Once we have computed a sharp PSF, we can then synthetically rotate this PSF to create the kernel at any desired angle. While we could capture separate images for a large set of rotations of the grating, we empirically find that such a methodology produces similar results to the synthetic rotation while requiring significantly more calibration effort and computation time.

(a) Simulated measurements



(b) Estimated θ_{start}

Figure 5. Validating our coarse approach for determining the start angle θ_{start} on a simulated scene. In short, we solve a coarse version of Eq. (4) using a dictionary that spans all possible rotations. We use this coarse reconstruction to estimate a score for every possible start angle, based on the energy contained in the subsequent 90° of reconstructed frames. Note the bimodal nature of the graph - the utilized grating PSF is roughly 180° symmetric. Here, $\lambda_{sparsity} = \lambda_{dx} = \lambda_{dy} = \lambda_{dt} = 5 \times 10^{-4}$.

5.2. Identifying the dictionary of angles

Determining what range of angles the PSF spans in a particular exposure is a more complicated problem. Assuming that the PSF spins at a uniform rate, this requires determining the start angle θ_{start} and end angle θ_{end}. We rely on an unstructured approach that simply requires that the scene contains content at the beginning and at the end of the camera exposure. First, we preconfigure the speed of the spinning disk and the camera exposure, such that the disk rotates some $\theta_{speed} = \theta_{end} - \theta_{start}$ in every captured image. To do this, for our diffraction grating, we fix the motor voltage, and then tune the exposure time of the camera such that $\theta_{speed} = 90^\circ$ — we compute θ_{speed} by correlating a image of a small retroreflector with different rotations of the previously calibrated $\mathbf{k}$.

Then, with θ_{speed}, we solve a coarse version of Eq. (4), using a dictionary of N_E PSFs $\mathbf{k}$ spanning angles between 0 and 360 degrees in order. We typically use around 2000 iterations. We take the maximum value of each of the output frames, and then compute a score for each angle consisting of the mean of the maximum value of the next θ_{speed} reconstructions. We then set θ_{start} to the largest score. Intuitively, in the reconstructed video, the correct angles will be the brightest frames, while the incorrect angles will be darker. We demonstrate the validity of this approach on simulated measurements in Fig. 5. Note that the plot is bimodal — the PSF of our diffraction grating is roughly 180° symmetric.

Once we have estimated θ_{start} and θ_{end}, we again solve Eq. (4) with a new dictionary consisting of equally spaced angles θ_{start} between θ_{end}. Because the motor may change speed over time, we then use a manual tuning step to refine θ_{start} and θ_{end}. For example, if there are dark frames at the

beginning and end of the sequence, we increase θ_{start} and decrease θ_{end}. However, if there is significant reconstructed brightness in the first and last frames, we decrease θ_{start} and increase θ_{end}.

5.3. Future hardware

In general, this above process is easy to implement, but lacks the rigor required for extremely accurate timing. In practice, there are a number of potential hardware solutions that could be utilized to improve the above system. For instance, a higher quality motor that rotates at a consistent, known rate would remove the need for the previously described manual tuning step.

Alternatively, one could simply place a calibration LED into the field of view of the camera. If the PSF and the location of the LED can be calibrated, then by examining the response present in the captured image the exact range of angles can be extracted. However, this LED leaves residual signal in the frame that needs to be removed or accounted for, which can potentially corrupt the reconstructions of the actual target scene.

Another potential option is to use a triggering system separate from the camera to directly capture the requisite start and end angles of the exposure. Under one potential implementation, a small dot could be painted onto the disk at a known angle θ_{dot} such that when it passes over a photodiode, the system can determine that the disk is at θ_{dot}. By tracking the amount of time required for subsequent activations of the photodiode, the speed of the disk can be identified on the fly. If the camera is triggered off these activations, then the start and end angles can also then be directly found if the trigger delay is calibrated.

Finally, a fast servo motor that allows careful control of the rotation speed and positioning would remedy all of the above problems, if properly synchronized with the camera. However, such a system would not allow for continuous, repeated capture, for long videos.

References

[1] Vahid Abolghasemi, Saideh Ferdowsi, Bahador Makkiabadi, and Saeid Sanei. On optimization of the measurement matrix for compressive sensing. In *2010 18th European Signal Processing Conference*, pages 427–431. IEEE, 2010. 2, 3

[2] 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)*, pages 1–8. IEEE, 2019. 3

[3] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006. 2

[4] Michael Elad. Optimized projections for compressed sensing. *IEEE Transactions on Signal Processing*, 55(12):5695–5702, 2007. 2

[5] Yasunobu Hitomi, Jinwei Gu, Mohit Gupta, Tomoo Mitsunaga, and Shree K Nayar. Video from a single coded exposure photograph using a learned over-complete dictionary. In *2011 International Conference on Computer Vision*, pages 287–294. IEEE, 2011. 1, 2

[6] Michael Iliadis, Leonidas Spinoulas, and Aggelos K Katsaggelos. Deepbinarymask: Learning a binary mask for video compressive sensing. *Digital Signal Processing*, 96:102591, 2020. 1, 2

[7] Yuqi Li, Miao Qi, Rahul Gulve, Mian Wei, Roman Genov, Kiriakos N Kutulakos, and Wolfgang Heidrich. End-to-end video compressive sensing using anderson-accelerated unrolled networks. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2020. 2

[8] Yi Luo, Jacky Jiang, Mengye Cai, and Shahriar Mirabbasi. Cmos computational camera with a two-tap coded exposure image sensor for single-shot spatial-temporal compressive sensing. *Optics express*, 27(22):31475–31489, 2019. 2

[9] Julien NP Martel, Lorenz K Mueller, Stephen J Carey, Piotr Dudek, and Gordon Wetzstein. Neural sensors: Learning pixel exposures for hdr imaging and video compressive sensing with programmable sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(7):1642–1653, 2020. 1, 2

[10] Cindy M Nguyen, Julien NP Martel, and Gordon Wetzstein. Learning spatially varying pixel exposures for motion deblurring. *arXiv preprint arXiv:2204.07267*, 2022. 1

[11] Travis Portz, Li Zhang, and Hongrui Jiang. Random coded sampling for high-speed hdr video. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2013. 2

[12] Dikpal Reddy, Ashok Veeraraghavan, and Rama Chellappa. P2c2: Programmable pixel compressive camera for high speed imaging. In *CVPR 2011*, pages 329–336. IEEE, 2011. 2

[13] Mark Sheinin, Matthew O'Toole, and Srinivasa G Narasimhan. Deconvolving diffraction for fast imaging of sparse scenes. In *2021 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10. IEEE, 2021. 1

[14] Mark Sheinin, Dinesh N Reddy, Matthew O'Toole, and Srinivasa G Narasimhan. Diffraction line imaging. In *European Conference on Computer Vision*, pages 1–16. Springer, 2020. 1