Deconvolving Diffraction for Fast Imaging of Sparse Scenes

Mark Sheinin, Matthew O'Toole, and Srinivasa G. Narasimhan

Abstract—Most computer vision techniques rely on cameras which uniformly sample the 2D image plane. However, there exists a class of applications for which the standard uniform 2D sampling of the image plane is sub-optimal. This class consists of applications where the scene points of interest occupy the image plane sparsely (*e.g.*, marker-based motion capture), and thus most pixels of the 2D camera sensor would be wasted. Recently, diffractive optics were used in conjunction with sparse (*e.g.*, line) sensors to achieve high-speed capture of such sparse scenes. One such approach, called "Diffraction Line Imaging", relies on the use of diffraction gratings to *spread* the point-spread-function (PSF) of scene points from a point to a color-coded shape (*e.g.*, a horizontal line) whose intersection with a line sensor enables point positioning. In this paper, we extend this approach for arbitrary diffractive optical elements and arbitrary sampling of the sensor plane using a convolution-based image formation model. Sparse scenes are then recovered by formulating a convolutional coding inverse problem that can resolve mixtures of diffraction PSFs without the use of multiple sensors, extending the application of diffraction-based imaging to a new class of significantly denser scenes. For the case of a single-axis diffraction grating, we provide an approach to determine the minimal required sensor sub-sampling for accurate scene recovery. Compared to methods that use a speckle PSF from a narrow-band source or a diffuser-based PSF with a rolling shutter sensor, our approach uses spectrally-coded PSFs from broad-band sources and allows arbitrary sensor sampling, respectively. We demonstrate that the presented combination of the imaging approach and scene recovery method is well suited for high-speed marker based motion capture and particle image velocimetry (PIV) over long periods.

Index Terms—Computational Photography, Motion Capture, PIV, Diffraction

1 Introduction

Despite significant advances in sensor technologies, capturing high speed videos at high spatial resolution remains a challenge. Even with sufficient SNR, the main limitation is the bandwidth to read out, digitize, transfer, and store a large volume of sensor data. Thus, most imaging sensors sacrifice spatial resolution for temporal resolution, or vice-versa. That said, there are several applications where the fast moving scenes are sparse (Fig. 1). For example, retro-reflective markers or LEDs for motion capture [1], [2], [3], reflective particles for fluid velocimetry [4], [5], [6], [7], combustible particles, the headlights and tail-lights of moving vehicles, or the decorative lighting and street lamps viewed from a fast moving vehicle [8], [9]. Using full-frame 2D sensing for these scenes is overkill and limits the achievable temporal resolution. This work addresses capturing sparse scenes at high speeds (several kHz) with sensors that are rated for only 30-120 Hz at full-frame resolutions.

One way to address this problem is to modify the electronics and read-out circuitry of the sensors. Event cameras [7], [10], [11] are designed to measure changes in intensity at every pixel and transmit only "large" changes to save bandwidth. But the sensors used in these cameras typically have low spatial resolution and suffer from low fill-factors due to the additional electronics needed at each pixel [7]. Traditional image sensors, on the other hand, have high fill-factors and can capture images at high frame rates by specifying small regions of interest (ROIs). But these ROIs are not guaranteed to capture the sparse points occurring anywhere in the field of view. Positional information of motion capture scene markers can be facilitated by temporal modulation of the markers [1], [2], [12], [13], but these techniques are limited to specialized markers, and may require synchronization.

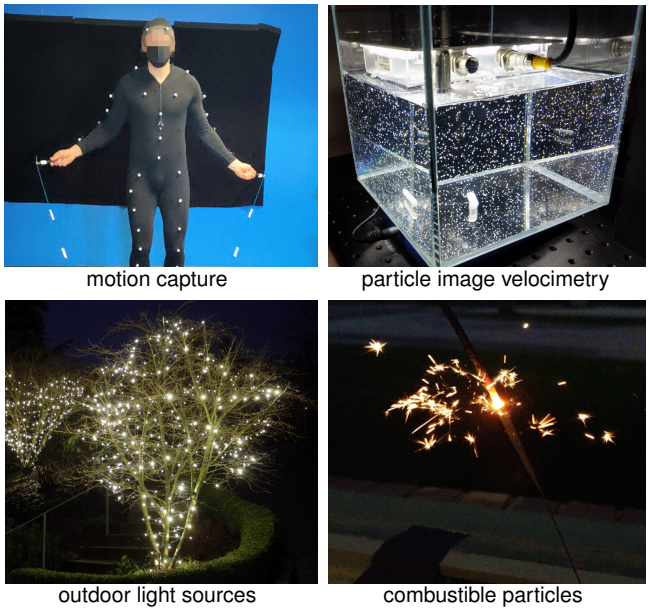


Fig. 1. Many imaging scenarios involving detecting sparse sources, such as the retro-reflective markers on a motion capture suit, tracer particles suspended in fluid, the thermal reaction from sparklers, and night light sources. In this work, we propose to recover images of such scenes at high speeds through diffraction-based imaging.

• *M. Sheinin, M. O'Toole, and S. Narasimhan are with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. E-mail: marksheinin@gmail.com*

Alternatively, imaging efficiency can be increased by optically multiplexing the incident illumination and computationally decoding an image of the scene. Engineering the PSF of scene sources was used for 3D cellular-imaging and tracking [14], [15], [16], [17], [18], [19], multi-color localization [20], [21], and depth recovery [22], [23], [24]. For high-speed capture, the incident light can be multiplexed to recover the scene from a small number of sensor measurements [25], [26], [27], [28], [29]. For example, Antipa *et al*. [26] recover high-speed videos of the scene with the help of a custom-fabricated diffuser placed in front of a bare sensor. The diffuser produces a caustic pattern that changes in response to the scene. By using a rolling-shutter sensor, every row on the sensor samples the caustic at a different time, and a video can therefore be recovered from a sequence of row measurements. The optical setup does not encode the entire scene all at once however; instead, the setup effectively has a time-varying FOV that only captures part of the scene. As such, in each frame, their method misses the fast motion of individual points that lay outside the time-varying FOV. Conversely, our method is designed for sparse scenes and utilizes multiple fixed ROIs to recover all scene points in each captured frame.

Weinberg *et al*. [27] recover images of sparse sources by relying on a high-frequency speckle point spread function (PSF) created on the image plane, and imaged once again with a rolling-shutter sensor. The speckle PSF is created by imaging narrow-band (single color) point sources through a diffuser added at the pupil plane. While their method requires specialized narrow-band sources, our approach exploits broad-band (white) light sources.

Sheinin et al. [30] use a diffraction grating to spread incident light. Their method creates a spectrally-dependant PSF, where the point position encoding relies not only on the PSF shape, but also relies on the PSF's spectrum (color). This method allows point position encoding using a line sensor coupled with off-the-shelf diffraction gratings and light sources. Diffraction gratings are easily available and have been used for artistic effect [31], spectroscopy [32], [33], multi-spectral sensing [34], [35], [36], [37], and rainbow particle velocimetry [6]. But their work has three important limitations: (a) they focus only on a line diffraction pattern, (b) their reconstruction works for points that are very sparse and fails in situations (e.g., points on a uniform grid) where multiple scene points contribute different spectral intensities to the same pixel, and (c) they require multiple cameras to resolve the position of points reliably.

We extend the approach of Sheinin et al. [30] to arbitrary diffraction patterns and arbitrary sensor sampling of the sparse scene point distributions, while using only a single camera. The image formation can be modeled as a spatially invariant convolution of the diffraction pattern and the sparse scene. The point-spread function (PSF) of the diffraction pattern can be measured for any type of scene point or source. The image can be acquired using an arbitrary configuration of ROIs and is deconvolved by enforcing sparsity. For the case of a single-axis diffraction grating yielding a 'rainbow streak' PSF sampled with horizontal ROIs, we derive an approach to minimize the ROI configuration (and maximize frame rate) based on the diffraction PSF. The convolutional model is then extended to scenes with multiple spectra and the sparse scene is estimated using a known dictionary of PSFs. In contrast to Sheinin et al. [30], our approach produces images instead of a list of 2D points, does not require any training data, does not require multiple cameras, applies to more dense scenes, and is not limited to line diffraction gratings/sensors or particular sparse scene configurations.

We demonstrate our approach using examples in several application domains, such as motion capture of fast moving markers and particle image velocimetry (PIV). Our system prototype is composed of just an off-the-shelf single-axis or double axis diffraction grating mounted in front of a single, ordinary 120 Hz camera that is capable of outputting multiple horizontal ROIs. We use our prototype to accurately capture high speed motions (kilo-hertz) like skipping ropes, bullets from a toy gun, complex fluid vortices, and even sparklers. Please see supplementary videos for better visualizations. While we primarily demonstrate 2D position estimates of the sparse scene points at high speeds, this approach can be used with multiple cameras and standard 3D estimation pipelines.

2 Convolutional Image Formation Model

A diffraction grating is a thin optical element that disperses light as a function of wavelength. When illuminated with a wide spectrum (*e.g*., white light), the periodic micro-structure of a diffraction grating produces rainbow patterns, similar to the effect that a prism has on light. Simple periodic structures result in simple patterns that consist of lines, such as the single rainbow streak or the rainbow-colored star (as shown in Fig. 2[b]). In this section, we describe a convolutional image formation model for a sparse scene when viewed through such diffractive optics. Our model can also support diffractive optical elements (DOEs), which can produce almost any desired light distribution by carefully designing the micro-structure of the optical element [38], [39], [40].

Let $\boldsymbol{\delta}_{\mathbf{s}}$ denote the image of a single bright scene point mapped to a single pixel $\mathbf{s}$. Note that $\boldsymbol{\delta}_{\mathbf{s}}$ is an intensity-only image of the point, where the intensity value at $\mathbf{s}$ can take arbitrary values.[1] If the scene consists of multiple points that emit the same normalized spectrum of light, the resulting image (Fig. 2[a]) is given by:

$$\mathbf{I} = \sum_{\mathbf{s}} \boldsymbol{\delta}_{\mathbf{s}}. \tag{1}$$

When placed in front of a camera, the diffraction grating blurs the image with a rainbow-colored point spread function (PSF). If $\mathbf{s}$ is the center of the frame, the result of imaging $\boldsymbol{\delta}_{\mathbf{s}}$ is the PSF itself (Fig. 2[b]). When viewing the scene through the diffraction grating with a plurality of points, the image formation process can be modeled as a convolution between $\mathbf{I}$ and the PSF $\mathbf{h}_{\sigma}^{\text{diff}}$ of the diffraction grating:

$$\mathbf{I}_{\sigma}^{\text{diff}} = \mathbf{I} * \mathbf{h}_{\sigma}^{\text{diff}}, \tag{2}$$

where σ represents the different color channels of a camera. As shown in Fig. 2[c], the resulting "diffraction" image consists of a superposition of multiple copies of the diffraction

1. Not to be confused with a Kronecker delta function.

PSF, shifted to the scene point positions and scaled by the intensity of each point.

Capturing a full-frame video stream with a conventional sensor at high frame rates is limited by bandwidth. Instead, our approach is to recover a full frame image $\mathbf{I}$ from a sparse set of measurements on the sensor, modeled as:

$$\mathbf{I}_\sigma^{\text{sparse}} = \mathbf{W}_\sigma \odot \mathbf{I}_\sigma^{\text{diff}} = \mathbf{W}_\sigma \odot \left(\mathbf{I} * \mathbf{h}_\sigma^{\text{diff}}\right), \quad (3)$$

where, the operator $\odot$ denotes an element-wise Hadamard product, and the matrix $\mathbf{W}_\sigma$ is a binary sampling matrix with ones for sampled pixels and zeros otherwise. Note that $\mathbf{W}_\sigma$ can vary per color channel σ, *e.g.*, to encode the Bayer color filter array associated with most cameras. Although both $\mathbf{I}_\sigma^{\text{sparse}}$ and $\mathbf{I}$ are represented here as having the same size, the image $\mathbf{I}_\sigma^{\text{sparse}}$ requires far less bandwidth since $\mathbf{W}_\sigma$ is sparse, and only a small subset of camera pixels are needed to populate $\mathbf{I}_\sigma^{\text{sparse}}$. For example, Fig. 2(c-d) sub-samples 16 of 1542 rows, reducing the bandwidth of the video stream by a factor of nearly 100.

3 Recovering the Sparse Scene

We now seek to recover $\mathbf{I}$ from the sparsely sampled sensor measurements $\mathbf{I}_\sigma^{\text{sparse}}$. Here, we assume that the system PSF $\mathbf{h}_\sigma^{\text{diff}}$ is known, and acquired through a calibration procedure described in Section 4.2. Image $\mathbf{I}$ encodes two pieces of information about each scene point: intensity and position on the image plane. Some of the applications discussed in the paper, such as motion capture with markers and PIV, are concerned with only the point positions. In this section, we first provide a general method for recovering $\mathbf{I}$, followed by a theoretical discussion in Section 3.2 about the conditions for which the approach can yield precise point positioning.

In the general case, a plurality of scene points yield a mixture of point PSFs on the sensor. Coupled with image noise, this makes a direct deconvolution of Eq. (3) an underdetermined problem. Therefore, the reconstruction is formulated as an optimization:

$$\underset{\mathbf{I}}{\operatorname{argmin}} \frac{1}{2} \sum_\sigma \left\| \mathbf{W}_\sigma \odot \left(\mathbf{I} * \mathbf{h}_\sigma^{\text{diff}}\right) - \mathbf{I}_\sigma^{\text{sparse}} \right\|_2^2 + \gamma \left\|\mathbf{I}\right\|_1, \quad (4)$$

where, the first term is the data fidelity term, the second term is a sparsity enforcing regularization term, and the scalar γ controls the contribution of the regularizer. This is a convex optimization problem commonly used by compressed sensing methods [26], [27] and can be solved with readily available software packages [41], [42]. Fig 2 shows an example that illustrates the recovery process.

3.1 Extension to Multiple Spectra

We previously assumed that all scene points share the same reflectance spectrum. This allows the image formation model to be the convolution between an intensity-only image and a diffraction PSF (see Eq. (2)). However, general scenes can contain a variety of sources, each emitting or reflecting a unique spectrum. We therefore extend our image formation model to handle multiple spectra and generalize the corresponding reconstruction procedure.

Suppose the reflectance spectrum of each scene point can be described as a linear combination of K reflectance

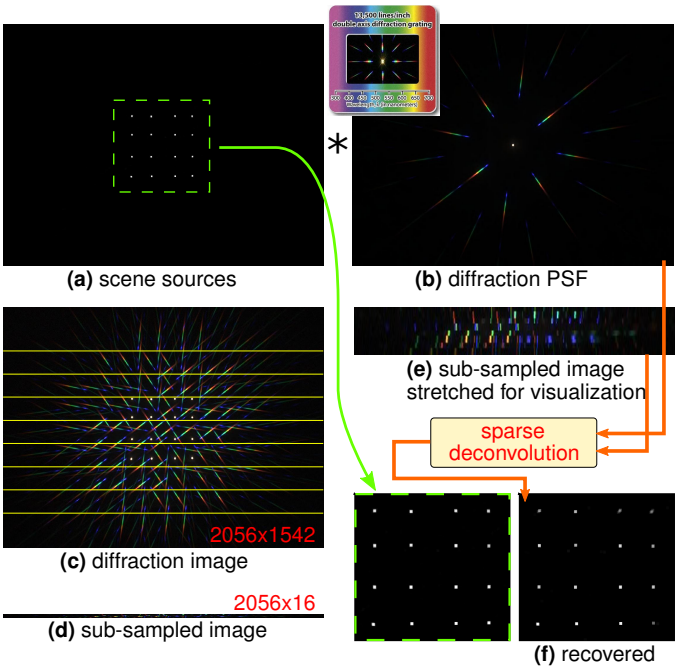


Fig. 2. Convolutional image formation and recovery. **(a)** An image of a 4×4 LED matrix captured using a standard 2D camera. **(b)** When imaged through a double-axis diffraction grating (inset), a single LED's point spread function (PSF) has the shape of a rainbow-colored star spanning a large part of the image domain. **(c)** The LED matrix of (a) is imaged through the diffraction grating. The resulting image can be modeled as a convolution of the standard 2D frame (a) with the individual LED PSF (b). **(d)** The full 2D image is then sub-sampled by taking eight narrow slices of size 2056×2 (marked by yellow lines) to yield this sub-sampled image. **(e)** The image in (d) vertically stretched for visualization. **(f)** The sub-sampled image is used along with the LED PSF to yield a reconstruction of the full 2D frame.

spectra. For example, the scene shown in Fig. 3 contains LEDs of four different colors, each having a unique spectrum. Let $\mathbf{h}_{\sigma,k}^{\text{diff}}$ denote the PSF of a scene point having reflectance spectra $k=1,2,\ldots,K$. Then the resulting sub-sampled image can be modeled as

$$\mathbf{I}_\sigma^{\text{sparse}} = \mathbf{W}_\sigma \odot \sum_{k=1}^{K} \left(\mathbf{I}_k * \mathbf{h}_{\sigma,k}^{\text{diff}}\right). \quad (5)$$

The image $\mathbf{I}_k$ denotes the contribution that spectrum k has to the signal at every pixel. In other words, each camera pixel can be dominated by a single spectrum coefficient k (see Fig. 3) or be a mixture of several coefficients. We can then reconstruct the signal by optimizing the following:

$$\underset{\mathbf{I}_1,\cdots,\mathbf{I}_K}{\operatorname{argmin}} \frac{1}{2} \sum_\sigma \left\| \mathbf{W}_\sigma \odot \left(\sum_{k=1}^{K} \mathbf{I}_k * \mathbf{h}_{\sigma,k}^{\text{diff}}\right) - \mathbf{I}_\sigma^{\text{sparse}} \right\|_2^2 + \gamma \sum_{k=1}^{K} \left\|\mathbf{I}_k\right\|_1. \quad (6)$$

An experiment showing a scene having $K=4$ sources with measured PSFs in shown in Fig. 3.

3.2 Domain of Position Recoverability

The above reconstruction algorithm handles an arbitrary sampling matrix, as well as an arbitrary PSF. However, not all PSF and sampling matrix combinations yield the same accuracy when recovering point positions from $\mathbf{I}$. Below, we discuss how the properties of the PSF constrain the sensor

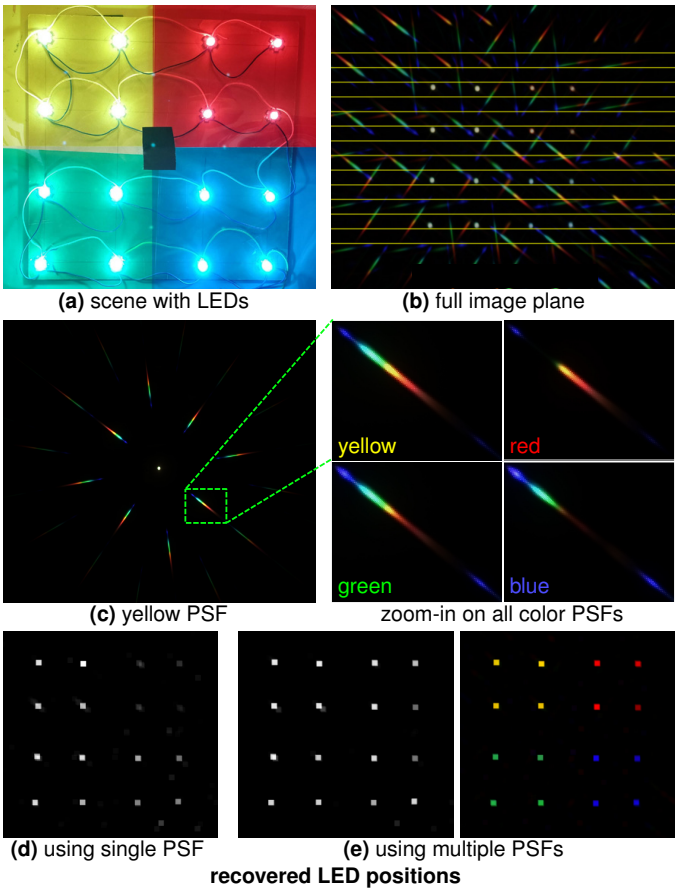


Fig. 3. Recovering sources of multiple spectra. **(a)** A scene with 16 LEDs of four different colors is imaged using a double-axis diffraction grating. **(b)** The 2D sensor plane is sampled using 14 two-pixel wide regions of interest (ROIs) illustrated by the yellow horizontal lines. **(c)** Each LED yields a star PSF in image (b) whose spectrum depends on the LED color. A close-up of a single diffraction mode of the four scene PSF spectra is shown on the right. **(d)** Scene recovery using Eq.(4) and a single PSF. **(e)** Scene recovery using a dictionary of the four scene PSFs with Eq. (6). Note that the single-dictionary recovery yields degraded results, while using multiple dictionaries improves the recovery. A by product of Eq. (6) are the individual coefficient weights of each PSF, visualized using the appropriate LED colors on the right.

sampling choice, and describe an approach to determine the recoverable point positions in the image domain for the case of a single-axis diffraction grating PSF sampled with horizontal ROIs.

Consider a scene containing a single point that maps to pixel $\mathbf{s}$. The point's image plane position can be uniquely recovered for an arbitrary PSF and sensor sampling matrix over an image domain Ω, if there exists an injective function f that maps the sampled and shifted PSF to its positional value $\mathbf{s}$ in this domain:

$$f\left(\mathbf{W}_\sigma \odot \left[\boldsymbol{\delta}_\mathbf{s} * \mathbf{h}_\sigma^{\text{diff}}\right]\right) \rightarrowtail \mathbf{s}, \quad \forall \mathbf{s} \in \Omega. \tag{7}$$

Fig. 4 illustrates various cases of the dependence of Ω on the PSF and $\mathbf{W}_\sigma$. Fig. 4[b-c] illustrate that, depending on $\mathbf{W}_\sigma$, the same PSF can yield an empty uniquely-recoverable domain or a uniquely-recoverable domain that spans the entire image plane. Specifically, the single pixel sampling in Fig. 4[b] can yield point positioning with an ambiguity of two horizontal pixels, while the line sampling shown in Fig. 4[c] yields no horizontal or vertical position ambiguity.

We seek a PSF and $\mathbf{W}_\sigma$ that maximize the support of Ω. This is a hard problem for a general PSF since Eq. (7) states an implicit condition for defining Ω but does not provide a method for computing it given the PSF and $\mathbf{W}_\sigma$. [2]

Now, we provide a method for computing Ω for the special case of a thin single-axis rainbow-streak PSF and horizontal ROIs. In a thin diffraction rainbow streak (Fig. 5[a]), the camera response function can yield unique color channel values per PSF row in response to the incident light spectrum. This means that a *subset* of the streak's rows define an injective function from the intensity-scaled color-channel values at the individual row to the streak's position.[3] Let $\mathbf{h}^{\text{inj}}$ denote a binary image that indicates the non-zero PSF pixels which belong to such 'injective' rows (see Fig. 5[a]). Given $\mathbf{h}^{\text{inj}}$, the recoverable domain can be estimated by computing its image domain support using $\mathbf{W}_\sigma$:

$$\Omega \approx \{\boldsymbol{x} \mid (\mathbf{W}_\sigma \star \mathbf{h}^{\text{inj}})[\boldsymbol{x}] > 0\}, \tag{8}$$

where $\star$ denotes the correlation operator. Intuitively, Eq. (8) computes the image pixels where a scene point's PSF will intersect at least one ROI with an injective PSF color.

Next, we describe how to compute $\mathbf{h}^{\text{inj}}$. Let r and c denote the image row and column indices, respectively. Let $\mathbf{W}_\sigma^r$ denote a sampling matrix having a single-row ROI at row r. Now let $\mathbf{I}_\sigma^r$ denote the PSF image sampled at row r:

$$\mathbf{I}_\sigma^r \equiv \mathbf{W}_\sigma^r \odot \mathbf{h}_\sigma^{\text{diff}}. \tag{9}$$

If the non-zero color channel values at row r of $\mathbf{h}_\sigma^{\text{diff}}$ are injective and there are no reconstruction errors, we expect that applying the reconstruction of Eq. (4) to $\mathbf{I}_\sigma^r$ would yield $\hat{\mathbf{I}} = \boldsymbol{\delta}_{\mathbf{s}^{\text{cent}}}$, where $\mathbf{s}^{\text{cent}}$ is the frame's center coordinate. Note that we expect to get $\boldsymbol{\delta}_{\mathbf{s}^{\text{cent}}}$ for all the injective rows r. Let $\hat{\mathbf{s}}_r$ denote the dominant pixel position by applying recovery to $\mathbf{I}_\sigma^r$, one for each r. In practice, the recovered point $\hat{\mathbf{s}}_r$ may slightly deviate from $\mathbf{s}^{\text{cent}}$ due to noise. Finally, $\mathbf{h}^{\text{inj}}$ is defined as

$$\mathbf{h}^{\text{inj}}[r,c] = \begin{cases} 1, & \text{if} \quad \mathbf{h}_{\text{int}}^{\text{diff}}[r,c] > t \quad \text{and} \quad |\hat{\mathbf{s}}_r - \mathbf{s}^{\text{cent}}| < d, \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

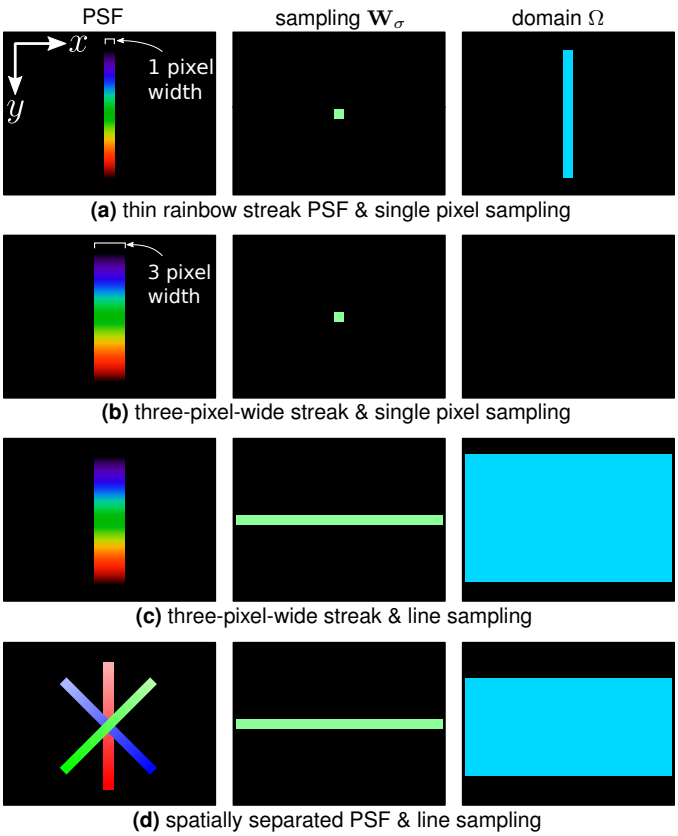
where $\mathbf{h}_{\text{int}}^{\text{diff}}$ is an intensity image of the PSF, t is an intensity threshold, and d is a pre-defined distance threshold (*e.g.*, two pixels). See supplementary material for an illustration of this process.

For multi-spectra scenes as in Section 3.1, we define the recoverable domain as the image domain which guarantees accurate recovery for a point belonging to any of the possible K PSFs. Thus, the recoverable domain Ω is the intersection of all the individual PSF domains Ω_k which are computed separately per PSF k using Eq. (8):

$$\Omega = \Omega_{\mathbf{h}_1^{\text{diff}}} \cap \Omega_{\mathbf{h}_2^{\text{diff}}} \ldots \cap \Omega_{\mathbf{h}_K^{\text{diff}}}. \tag{11}$$

2. The domain Ω can be found by exhaustively applying the reconstruction algorithm of Eq (4) for all possible $\mathbf{s}$ shifts.

3. For an standard color camera with RGB channels, the rainbow streak may not have unique RGB values and the ends of the visible spectrum, where the camera's response maps the deep blues and reds to $[0, 0, \alpha]$ and $[\alpha, 0, 0]$, respectively.



(a) thin rainbow streak PSF & single pixel sampling

(b) three-pixel-wide streak & single pixel sampling

(c) three-pixel-wide streak & line sampling

(d) spatially separated PSF & line sampling

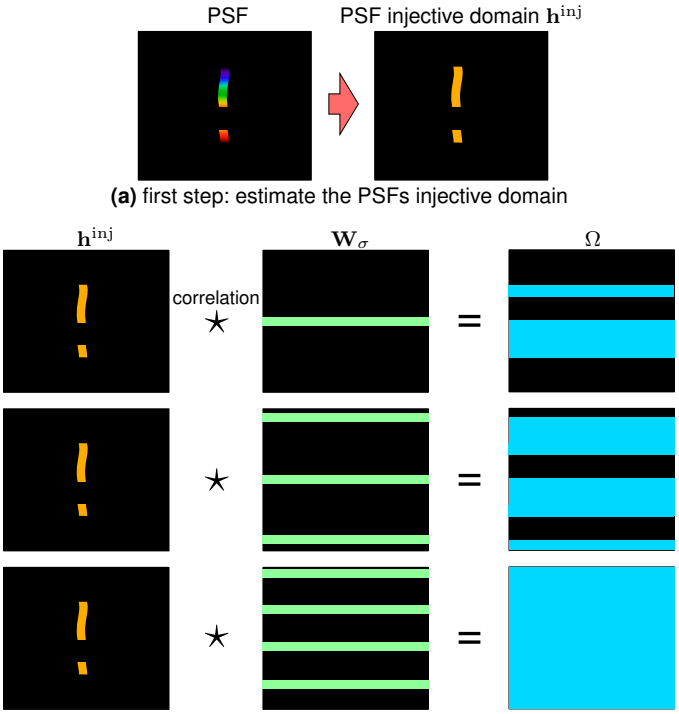
Fig. 4. Effect of PSF and sensor sampling on position recovery. The image domain Ω for which point positions can be recovered uniquely depends on the PSF and the sensor sampling matrix. **(a)** A vertical one-pixel wide rainbow PSF sampled using a single camera pixel defines a vertical recoverable domain with the same spatial support as the non-zero PSF pixels. **(b)** By increasing the PSF width to three pixels, a single pixel sampling can no longer determine $\mathbf{s}$ precisely anywhere in the image domain. This is because the vertical (y-axis) position can still be determined with certainty, but now a two pixel ambiguity exists in the horizontal position. **(c)** Sampling the entire row resolves the horizontal position ambiguity, showing that different $\mathbf{W}_\sigma$ yield different Ω for the same PSF. **(d)** The PSF does not have to be a single rainbow line. In this example, the PSF shift can be computed using the ratios between the different RGB channels as in (a-c). However, here these measurement are spatially separated across different columns of the sampling row.

3.3 Choosing $\mathbf{W}_\sigma$ using Horizontal Camera ROIs

Many machine vision cameras allow defining multiple regions of interests (ROIs) for which to perform sensor readout. The output image is then a concatenation of the ROIs of choice (see Fig. 2[c-d]). An ROI is defined as a rectangle in the image plane. As most cameras perform readout sequentially one row at a time, the capture speed usually depends on the total number of rows in all the defined ROIs.

Given these hardware constraints, our sampling matrix $\mathbf{W}_\sigma$ consists of multiple ROIs of size $2056 \times M$, where M is the vertical width of each ROI. The multiple ROIs are repeated vertically with a pitch of P pixels. Thus the image sub-sampling factor is P/M. Higher sub-sampling factors result in faster frame rates.

Maximizing the sub-sampling factor requires maximizing P while minimizing M. This can be achieved by setting M to its minimal allowable value (2 pixels for our cameras) while finding the highest $P = P_{\max}$ such that Ω covers the entire image domain (field of view). As illustrated in



(a) first step: estimate the PSFs injective domain

(b) second step: decrease ROI pitch until covers the entire image domain

Fig. 5. Estimating the uniquely recoverable domain. Given a PSF and a sensor sampling matrix $\mathbf{W}_\sigma$, the image uniquely-recoverable domain can be approximated by the following procedure: **(a)** First, estimate the PSF's injective pixel domain $\mathbf{h}^{\text{inj}}$. **(b)** Correlate $\mathbf{h}^{\text{inj}}$ with $\mathbf{W}_\sigma$ to yield Ω. In (b) we illustrate this process for choosing the ROIs' pitch P for a uniform horizontal sampling of the image plane.

Fig. 5[b], we find $P_{\max}$ by first estimating the PSF's injective domain $\mathbf{h}^{\text{inj}}$ as described in Section 3.2, and then using Eq. (8) to search for the maximal pitch $P_{\max}$.

4 Implementation Details

4.1 Hardware

Our prototype diffraction camera consists of an IDS UI-3070CP-C-HQ Rev.2 color camera mounted with a Fujinon 1.5MP 9mm lens. Our camera was limited to eight vertical ROIs. Therefore, we used eight ROIs in all high-speed experiments. We used two types of diffraction gratings in our experiments. For imaging emitters (*e.g.* LEDs), we used a simple double-axis diffraction grating tilted at 45 degrees (seen in Fig. 2[b]). For scenes with reflected light (motion capture, PIV), we used a Thorlabs 50mm 300 grooves/mm transmission grating (GT50-03). For motion capture, we used 14mm retro-reflective markers [43] and retro-reflective tape which were illuminated by an Advanced Illumination white ring light (RL-S052120) placed in front of the camera.

In the PIV experiment, we illuminated the top of the water tank with an Advanced Illumination spot light (SL-S100150). The water in the tank was mixed with Cospheric White Polyethylene Micro-spheres having a 1mm diameter which were stirred with a INTLLAB Magnetic Stirrer set to its maximum speed of 3000 RPMs.

4.2 System Calibration

We imaged a single scene point to acquire the PSFs needed for the optimization in Eq. (4) and (6). In the motion capture experiments, this amounted to imaging a single marker, while a single or multiple LEDs were imaged for experiments shown in Figs 2-3. In the PIV and sparklers experiments, placing a single steady point in the scene was impossible. Instead, we imaged a scene containing multiple sparse points and cropped the PSF of a single well-separated point. While generally the PSF may depend on depth, we observed no significant depth-dependent change in the PSF in our experiments. When using the double-axis diffraction grating with a single point spectrum, we set $P = 148$ with $M = 2$. For the single-axis 300 Grooves/mm diffraction grating we applied the procedure described in Section 3.3 which yielded a pitch $P_{\max} = 70$.

While this paper mainly illustrates the recovery of 2D images at high frame rates from one camera, we also use a stereo camera pair to evaluate the 3D positions of markers. For extrinsic and intrinsic calibration of the stereo pair in Fig. 6 we built a checkerboard pattern with retro-reflective tape at the square's corners (Fig. 6[b]). Since the calibration does not require high-speed capture, the pattern was imaged simultaneously by both stereo cameras in full-frame mode. The checkerboard corners were extracted using Eq. (4) and inputted to a standard calibration pipeline [44].

4.3 Optimization Details

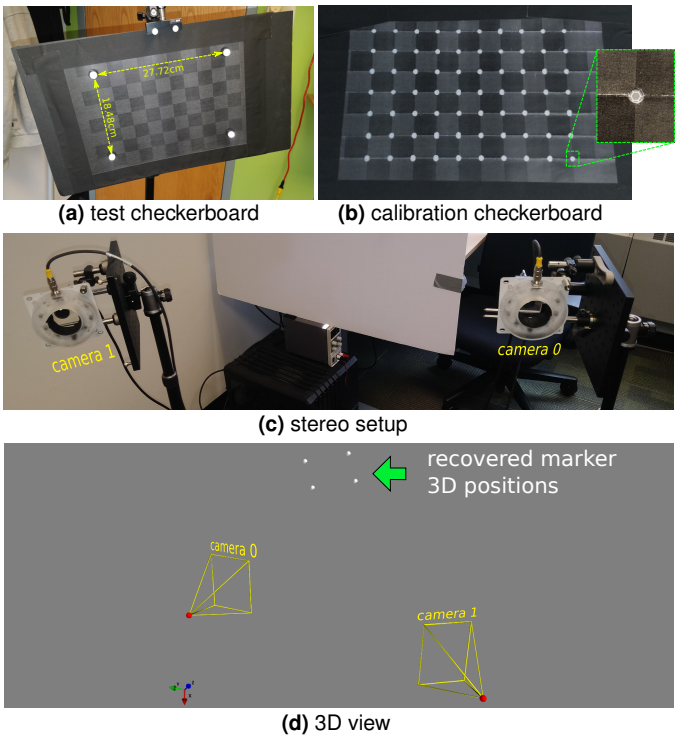
We solved the optimization problem in Eqs (4) and (6) using the SPORCO Python package [42], [45], [46]. We use a PGM solver along with Additive Mask Simulation (AMS) boundary handling technique to apply our sampling matrix $\mathbf{W}_\sigma$ [47], [48]. The solver has a complexity of $\mathcal{O}(KN \log N)$ per iteration per frame, where N and K are the number of frame pixels and scene PSFs, respectively. We set $\gamma = 0.01$ in Eq (4). Recovering a single frame at full resolution and a maximum of 1000 iterations takes approximately 60 seconds.

5 Experimental Evaluation

We have conducted an extensive experimental evaluation of our method. We showed that the method is readily applicable in many scenarios including motion capture (Section 5.2) and PIV (Section 5.3). Our method's output $\mathbf{I}$ is a sparse representation, sometimes resulting in a single pixel with a non-zero value per scene point, and thus hard to visualize. Therefore, for better visualization, all figures show $\mathbf{I} * \mathbf{h}_\sigma^{\text{spot}}$ instead of $\mathbf{I}$, where $\mathbf{h}_\sigma^{\text{spot}}$ is a small kernel intended to enlarge the recovered result.

5.1 Point Positioning Accuracy Experiment

We tested the accuracy of 3D marker positioning by placing four retro-reflective markers at known positions on a calibration checkerboard pattern (see Fig. 6[a]). The marker-checkerboard was then imaged by a diffraction-camera stereo pair (see Fig. 6[c]) whose intrinsic and extrinsic parameters were pre-calibrated as described in Section 4.2. The sensor sub-sampling factor was $P/M = 70/2 = 35$ in this experiment. Both cameras were tilted by 90 degrees. We captured several video streams of the marker-checkerboard



(a) test checkerboard **(b)** calibration checkerboard

(c) stereo setup

(d) 3D view

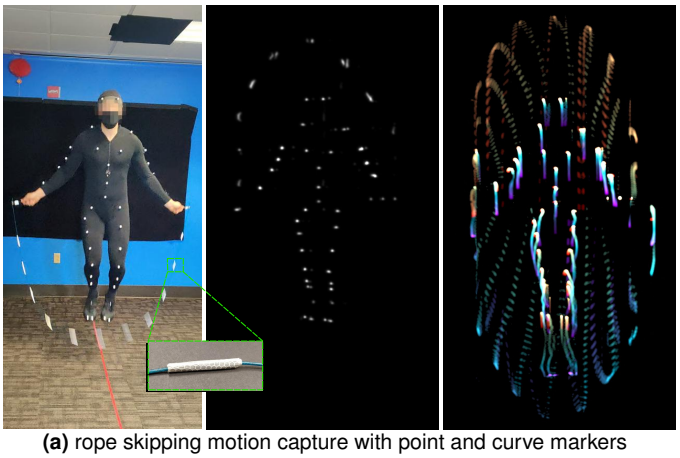
Fig. 6. Testing marker positioning accuracy. **(a)** Four markers are placed in known position on a planer object. **(b)** A retro-reflective 6x9 symmetric checkerboard pattern used for the system geometric calibration. **(c)** The stereo setup used in the experiment. **(d)** A 3D view showing the calibrated camera positions as well as the marker positions for a single video frame. The mean absolute positioning error and standard deviation in this experiment over multiple frames were 4mm and 4.3mm, respectively.

moving around in the scene (see video in supplementary material). The 2D positions of the four markers were recovered using our method (Fig. 6[d]) and were used to triangulate their 3D positions in space. From the retrieved 3D marker positions we computed the four distances between the markers and compared them to the ground truth distances (marked in yellow in Fig. 6[a]). The mean absolute error for all distance measurements was computed over 100 frames was 4mm with a standard deviation of 4.3mm.

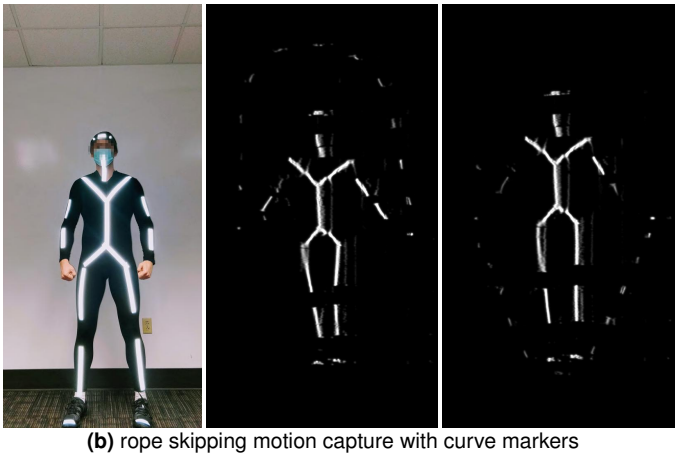
5.2 Motion Capture Experiments

We captured high-speed videos of an actor performing rope skipping (see Fig. 7). The actor was performing a fast rope skipping exercise, swinging the rope twice underneath his feet on every jump. The rope was partially covered with retro-reflective tape to enable the recovery of its position in the frames. In Fig. 7(b), the actor's suit was fitted with retro-reflective tape instead of point markers. The reconstructed frames show the actor's skeleton outlined by the tape. The capture speed in Fig. 7 was 1000FPS with a sub-sampling factor of $P/M = 70/12 = 5.8$. Please see the supplementary videos for additional results.

In Fig. 8 we capture the fast motion of darts fired from a toy gun. The toy gun and darts are covered with retro-reflective tape and markers for tracking. The gun is able to fire 5 darts per second. Our reconstructed frames visualize the fast dart trajectories. Here, the capturing speed was 1000FPS with a sub-sampling factor of $P/M = 70/2 = 35$.

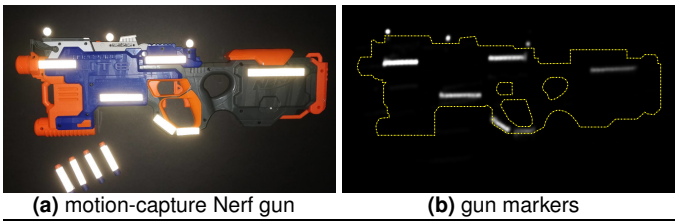


(a) rope skipping motion capture with point and curve markers

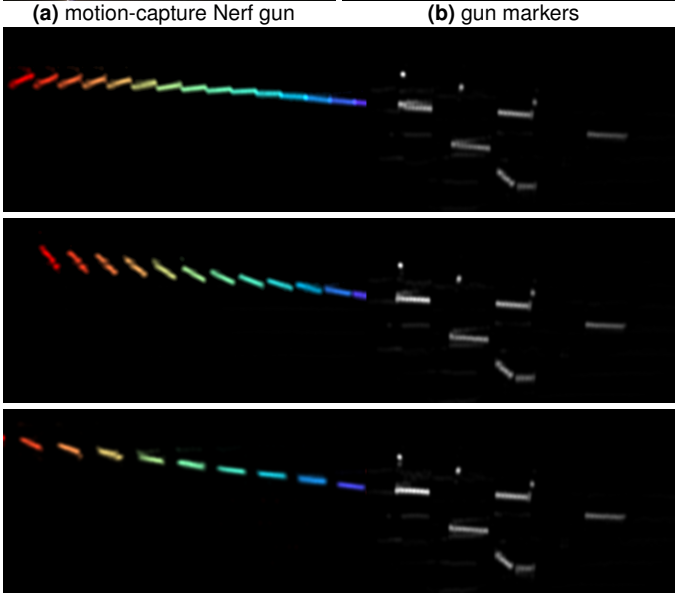


(b) rope skipping motion capture with curve markers

Fig. 7. High-speed motion capture. **(a)** An actor wearing a suit fitted with retro-reflective markers performs "double unders" rope skipping, where the rope is swung twice under the feet on every jump. The jump rope is periodically marked with retro-reflective tape. Our system captures the motion at 1000FPS. The middle subplot shows a single reconstructed frame. The right subplot visualizes a 300ms time duration using 100 frames spaced 3ms apart (taking every third frame). Observe that the actor completes a full rope revolution while being mid jump, followed by an additional revolution completed before the next jump. **(b)** Our method is not limited to points. The actor is fitted with retro-reflective tape which outlines the actor's skeleton. The middle and right subplot show two frames belonging to the same rope revolution.



(a) motion-capture Nerf gun **(b)** gun markers



(c) bullet flight reconstruction

Fig. 8. Capturing darts in flight. **(a)** A toy gun was fitted with retro-reflective markers and tape. The gun's plastic darts were also wrapped with retro-reflective tape. The gun, able to fire five darts per second, is imaged using our method at 1000 FPS. **(b)** The gun's recovered retro-reflective markers. **(c)** Bullet flight reconstruction. Each subplot shows a superposition of multiple frames belonging to the trajectory of a single (different) dart. The bullet positions in different frames is visualized using a different color. Here we visualize the darts by superimposing every seventh frame, which corresponds to a time interval of 7 milliseconds. Notice that the retro-reflective tape added to the darts disturbs their intended weight balance, thereby altering the dart's intended 'straight line' trajectories.

5.3 Particle image velocimetry

In Fig 9[a-b], the camera views a water tank seeded with white 1mm-diameter micro-spheres. The water is stirred using a magnetic stirrer set to its maximum speed of 3000 RPM. The tank was illuminated from above using a spot illumination. Our camera captured a 2D projection of the particles' 3D flow at 1000FPS. Fig 9(c) shows an illustration of the flow created in the tank by the stirrer (see supplementary video). In our tank, the stirrer created an axial flow whose motion can be described as: going from the stirrer, to the sides of tank and returning back to the stirrer from the top of the tank. Fig 9(d) shows a single recovered frame from the high-speed video. Fig 9(e) shows the recovered diffraction frame of Fig 9(d). The used ROIs are shown in yellow. Observe the high density of the diffraction image. Fig 9(f) shows a visualization of the flow for a time span of 150ms. The visualization was created by superimposing 15 frames spaced 10 frames apart. We used an open-source PIV package to compute the 2D projected flow [49]. Fig 9(g) shows the normalized flow computed using two frames spaced 6ms apart which is consistent with the perceived flow in the tank.

5.4 Comparison to Diffraction Line Imaging [30]

Our method extends diffraction line imaging to a whole new class of scenes, *e.g.*, where multiple sources are positioned on the same column or row. Figs. 2 and 3 show examples for sparse scenes where the scene sources overlap both horizontally and vertically. Applying the method of Sheinin *et al.* [30] fails in these cases (see Fig. 10[a-c]). Figs. 7 and 8 contain a mixture of retro-reflective point markers and 'curve' markers made with retro-reflective tape. The curve markers often yield horizontal overlap between themselves or with other point markers. As seen in the figures, our method is able to correctly resolve such frames. Finally, the high point density in Fig. 9 yields multiple points in almost all image columns and rows per frame, ubiquitously breaking the assumption of Sheinin *et al.* [30] and causing recovery to fail (see Fig. 10[d]). Nevertheless, Sheinin *et*

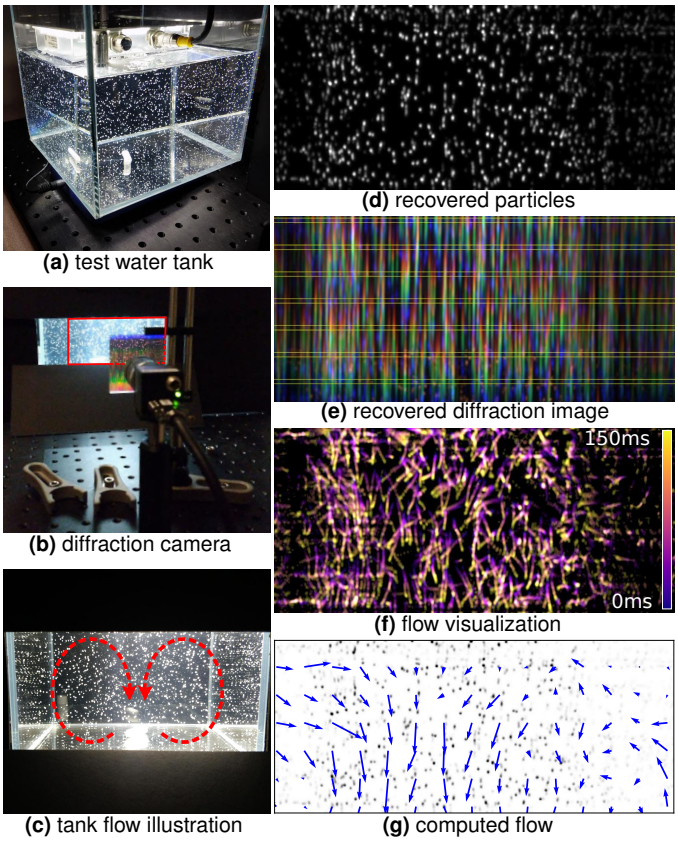


Fig. 9. Particle image velocimetry setup. **(a)** A water tank seeded with white diffuse PIV particles is placed on a magnetic stirrer. **(b)** The particle flow is imaged using our diffraction camera at 1000FPS. **(c)** The average stationary Axial flow created by the stirrer can be described as looping from the stirrer, to the sides of the tank upward, and returning from the tank's center. **(d)** A single recovered camera frame. **(e)** A 2D diffraction image, computed by the recovered $\mathbf{I}$ and the scene PSF. **(f)** A visualization of the flow computed by superimposing 15 frames spaced 10ms apart. **(g)** Normalized flow field from two frames spaced 6ms apart, computed using an open-source PIV package. Observe that the computed average flow is consistent with the observed flow.

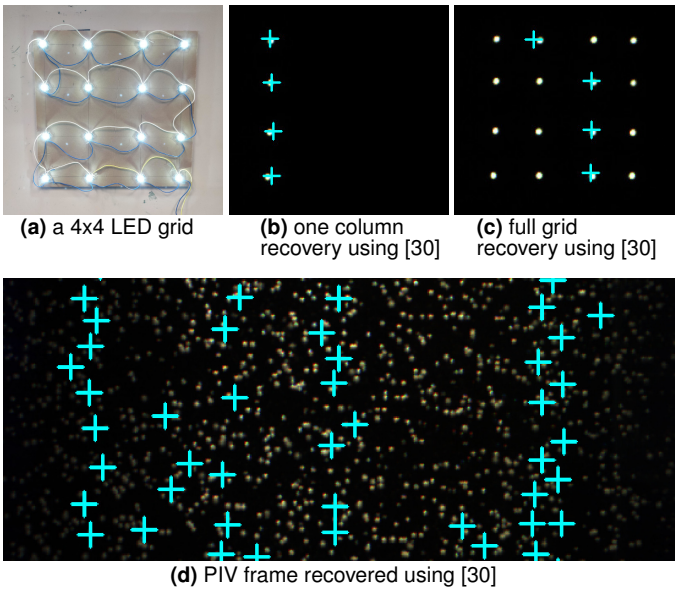


Fig. 10. Comparison to [30]. **(a)** Input scene. **(b)** The method in [30] maps the intersection of vertical diffraction rainbow streak with horizontal ROIs. Thus the method works as long as there are no two point that share the same image row. **(c)** When imaging the full grid of LEDs, the method of [30] fails to recover the grid points, even when using multiple ROIs. **(d)** The high point density in the PIV experiment of Fig. 9 yields multiple sources in each column and row of the captured video frames. Hence, the method of [30] completely fails to recover the source's positions (yellow=ground truth, cyan=recovered positions).

al. [30]'s method has the advantage of not assuming prior knowledge of scene source spectra as well as a faster run speed suitable for real-time operation.

6 Analysis and Discussion

In this section we discuss how the design choices of our system affect the overall system performance, the method's limitations and future work.

6.1 System Design Considerations

Our experiments show a trade-off between the different choices of PSF, created by using different diffractive optics. A single-axis diffraction grating is energy efficient since most of the light's power is concerted in a single diffraction mode (*i.e.*, the rainbow streak or line). But, the single-axis PSF yields higher positioning uncertainty in the vertical axis [30], and has a limited injective domain, which reduces the sub-sampling factor. Conversely, the "star" PSF shown in Fig. 2 and Fig. 3 has a large spatial support and contains rainbow streaks in both axis. It therefore provides accurate localization in both spatial coordinates and enables high sensor sub-sampling factors, typically upwards of 100. However, its spatial spread comes at the cost of light efficiency, making it mostly useful for high-speed imaging of relatively bright sources such as LEDs and car headlights. Note that compared to methods that use a speckle PSF from a narrow-band source [27] or a diffuser-based PSF [26], in both the streak and star PSFs, the PSF color provides an additional cue that can potentially relax the need for generating spatially-sharp PSF features.

Diffraction-based imaging assumes sparsity in the scene points or sources. As the scene becomes denser (Fig. 9), a higher sensor sampling (lower sub-sampling factor) is required for accurate point positioning, since it increases the signal, making the optimization more robust to noise and signal saturation. Our high-speed experiments were limited to using 8 ROIs. We therefore decreased the sub-sampling factor by increasing the width of each ROI while still maintaining the desired capture speed.

Another notable design choice is sampling the scene with fixed sensor ROIs versus the short inter-row delay of a rolling shutter sensor [26], [27]. While using the rolling shutter may provide very high-speed bursts of recovered frames, such sampling does not support prolonged continuous high-speed capture. This is because the delay between two consecutive rolling shutter frames may be much longer than the inter-row delay used for the high-speed bursts [26]. Thus, the maximum amount of very high-speed frames in each burst is limited to the number of samples that 'fit' a single rolling shutter frame. Conversely, our ROI-based solution continuously outputs the video frames at high FPS and therefore is optimized for the long captures required by our applications.

6.2 Limitations and Future Work

Our method's performance depends on the system's hardware components: camera, diffractive optics, and illumination source. In low SNRs, the resulting diffraction PSF may not yield a uniform reconstruction accuracy across the image domain. This is because the PSF may have a non-uniform spectra intensity resulting in some wavelengths (or PSF colors) having relatively low image intensity compared to others. For example, the LED illumination used in our motion-capture experiments decreases sharply between the blue and green wavelengths. This causes artifacts in videos with relatively low SNR (*e.g.*, where the actor is far way from the camera), that appear as low recovered signal at fixed-location video columns (see supplementary video). Such artifacts can be alleviated by using light sources with a more uniform spectral power distribution, increasing the sub-sampling factor, or by post processing the recovered videos to compensate for the fixed-location low-signal regions.

Recent experimental cameras that allow an arbitrary sampling of the sensor plane may yield higher performance by designing more efficient sampling matrices [50]. Conversely, improved performance can be obtained by designing custom diffractive optical elements that yield a PSF with superior performance. For instance, the star PSF used here can be improved by concentrating most of the energy in just two diffraction modes (rainbow lines): one horizontal and one vertical, with minimal energy along the rest of the modes. Our system's dynamic range is determined by the dynamic range of the camera in our prototype (without the grating). However, our system could potentially extend the camera's dynamic range similarity to prior methods that use diffractive optics to yield HDR scenes [38], [51]. Finally, incorporating a learning-based component to position recovery may further improve point positioning by learning to denoise the PSF-specific artifacts that result in the reconstructed frames.

7 Conclusion

We have extended diffraction line imaging to handle an arbitrary sensor sampling combined with an arbitrary diffractive element. Our system is easy to assemble, requiring just a single diffractive optical element mounted in front of one camera. Calibration is fast and straightforward, requiring a single capture of the scene's PSF. Our reconstruction algorithm requires no prior data-set for training. In conclusion, we believe our method has the potential to allow high-speed capture of sparse scenes in various applications.

Acknowledgments

The authors would like to thank Brendt Wohlberg for the support with the SPORCO Python package, Justin Macey for lending us the motion capture suit and markers, Adithya Pediredla and Dinesh Reddy for help with the motion capture experiments. This work was supported in parts by NSF Grants IIS-1900821 and CCF-1730147. M. Sheinin was partly supported by the Andrew and Erna Finci Viterbi Foundation.

References

[1] "Optotrak certus," https://www.ndigital.com/msci/products/optotrak-certus/, Phase Space Inc., 2020.
[2] "Phase space inc." http://www.phasespace.com/, Phase Space Inc., 2020.
[3] "Vicon motion capture," https://www.vicon.com/, Vicon Motion Capture, 2020.
[4] L. Adrian, R. J. Adrian, and J. Westerweel, *Particle image velocimetry*. Cambridge university press, 2011, no. 30.
[5] L. Lourenco, A. Krothapalli, and C. Smith, "Particle image velocimetry," in *Advances in fluid mechanics measurements*. Springer, 1989, pp. 127–199.
[6] J. Xiong, R. Idoughi, A. A. Aguirre-Pablo, A. B. Aljedaani, X. Dun, Q. Fu, S. T. Thoroddsen, and W. Heidrich, "Rainbow particle imaging velocimetry for dense 3d fluid velocity imaging," *ACM TOG*, vol. 36, no. 4, p. 36, 2017.
[7] G. Gallego, T. Delbruck, G. M. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
[8] Y.-L. Chen, B.-F. Wu, H.-Y. Huang, and C.-J. Fan, "A real-time vision system for nighttime vehicle detection and traffic surveillance," *IEEE Trans. on Indust. Elect.*, vol. 58, no. 5, pp. 2030–2044, 2010.
[9] R. Tamburo, E. Nurvitadhi, A. Chugh, M. Chen, A. Rowe, T. Kanade, and S. G. Narasimhan, "Programmable automotive headlights," in *Proc. ECCV*, 2014, pp. 750–765.
[10] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *Proc. ECCV*. Springer, 2016, pp. 349–364.
[11] Y. Wang, R. Idoughi, and W. Heidrich, "Stereo event-based particle tracking velocimetry for 3d fluid flow reconstruction," in *European Conference on Computer Vision*. Springer, 2020, pp. 36–53.
[12] J. Kim, G. Han, H. Lim, S. Izadi, and A. Ghosh, "Thirdlight: Low-cost and high-speed 3d interaction using photosensor markers," in *Proc. CVMP*. ACM, 2017, p. 4.
[13] R. Raskar, H. Nii, B. de Decker, Y. Hashimoto, J. Summet, D. Moore, Y. Zhao, J. Westhues, P. H. Dietz, J. Barnwell, S. K. Nayar, M. Inami, P. Bekaert, M. Noland, V. Branzoi, and E. Bruns, "Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators," *ACM TOG*, vol. 26, no. 3, p. 36, 2007.
[14] B. Huang, W. Wang, M. Bates, and X. Zhuang, "Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy," *Science*, vol. 319, no. 5864, pp. 810–813, 2008.
[15] S. R. P. Pavani, M. A. Thompson, J. S. Biteen, S. J. Lord, N. Liu, R. J. Twieg, R. Piestun, and W. Moerner, "Three-dimensional, single-molecule fluorescence imaging beyond the diffraction limit by using a double-helix point spread function," *Proceedings of the National Academy of Sciences*, vol. 106, no. 9, pp. 2995–2999, 2009.
[16] Y. Shechtman, S. J. Sahl, A. S. Backer, and W. Moerner, "Optimal point spread function design for 3d imaging," *Physical review letters*, vol. 113, no. 13, p. 133902, 2014.
[17] Y. Shechtman, L. E. Weiss, A. S. Backer, S. J. Sahl, and W. Moerner, "Precise three-dimensional scan-free multiple-particle tracking over large axial ranges with tetrapod point spread functions," *Nano letters*, vol. 15, no. 6, pp. 4194–4199, 2015.
[18] S. Jia, J. C. Vaughan, and X. Zhuang, "Isotropic three-dimensional super-resolution imaging with a self-bending point spread function," *Nature photonics*, vol. 8, no. 4, pp. 302–306, 2014.
[19] M. R. Hatzvi and Y. Y. Schechner, "Three-dimensional optical transfer of rotating beams," *Optics letters*, vol. 37, no. 15, pp. 3207–3209, 2012.
[20] Y. Shechtman, L. E. Weiss, A. S. Backer, M. Y. Lee, and W. Moerner, "Multicolour localization microscopy by point-spread-function engineering," *Nature photonics*, vol. 10, no. 9, pp. 590–594, 2016.
[21] E. Hershko, L. E. Weiss, T. Michaeli, and Y. Shechtman, "Multicolor localization microscopy and point-spread-function engineering by deep learning," *Optics express*, vol. 27, no. 5, pp. 6158–6183, 2019.
[22] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM transactions on graphics (TOG)*, vol. 26, no. 3, pp. 70–es, 2007.
[23] A. Greengard, Y. Y. Schechner, and R. Piestun, "Depth from rotating point spread functions," in *Optical Information Systems II*, vol. 5557. International Society for Optics and Photonics, 2004, pp. 91–97.

[24] M. Sheinin and Y. Y. Schechner, "Depth from texture integration," in *2019 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2019, pp. 1–10.
[25] N. Antipa, G. Kuo, R. Heckel, B. Mildenhall, E. Bostan, R. Ng, and L. Waller, "Diffusercam: lensless single-exposure 3d imaging," *Optica*, vol. 5, no. 1, pp. 1–9, 2018.
[26] N. Antipa, P. Oare, E. Bostan, R. Ng, and L. Waller, "Video from stills: Lensless imaging with rolling shutter," in *Proc. IEEE ICCP*, 2019, pp. 1–8.
[27] G. Weinberg et al., "100,000 frames-per-second compressive imaging with a conventional rolling-shutter camera by random point-spread-function engineering," *in Optics Express 2020*.
[28] J. Wang, M. Gupta, and A. C. Sankaranarayanan, "Lisens-a scalable architecture for video compressive sensing," in *Proc. IEEE ICCP*. IEEE, 2015, pp. 1–9.
[29] A. Liutkus, D. Martina, S. Popoff, G. Chardon, O. Katz, G. Lerosey, S. Gigan, L. Daudet, and I. Carron, "Imaging with nature: Compressive imaging using a multiply scattering medium," *Scientific reports*, vol. 4, no. 1, pp. 1–7, 2014.
[30] M. Sheinin, D. N. Reddy, M. O'Toole, and S. G. Narasimhan, "Diffraction line imaging," in *European Conference on Computer Vision*, 2020.
[31] D. Liu, H. Geng, T. Liu, and R. Klette, "Star-effect simulation for photography," *Computers & Graphics*, vol. 61, pp. 19–28, 2016.
[32] H. Nagaoka and T. Mishima, "A combination of a concave grating with a lummer-gehrcke plate or an echelon grating for examining fine structure of spectral lines," *The Astrophysical Journal*, vol. 57, p. 92, 1923.
[33] S. S. Vogt, S. L. Allen, B. C. Bigelow, L. Bresee, W. E. Brown, T. Cantrall, A. Conrad, M. Couture, C. Delaney, H. W. Epps *et al.*, "HIRES: the high-resolution echelle spectrometer on the Keck 10-m Telescope," in *Instru. in Astronomy VIII*, vol. 2198. Intern. Society for Optics and Photonics, 1994, pp. 362–375.
[34] D. S. Jeon, S.-H. Baek, S. Yi, Q. Fu, X. Dun, W. Heidrich, and M. H. Kim, "Compact snapshot hyperspectral imaging with diffracted rotation," *ACM TOG*, vol. 38, no. 4, p. 117, 2019.
[35] V. Saragadam and A. C. Sankaranarayanan, "KRISM: Krylov subspace-based optical computing of hyperspectral images," *ACM TOG*, vol. 38, no. 5, pp. 1–14, Oct. 2019.
[36] V. Saragadam and A. C. Sankaranaryanan, "Programmable Spectrometry: Per-pixel material classification using learned spectral filters," in *IEEE Intl. Conf. Computational Photography (ICCP)*, 2020.
[37] ——, "On space-spectrum uncertainty analysis for coded aperture systems," *Optics Express*, vol. 28, pp. 7771–7785, 2020.
[38] Q. Sun, E. Tseng, Q. Fu, W. Heidrich, and F. Heide, "Learning rank-1 diffractive optics for single-shot high dynamic range imaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1386–1396.
[39] S.-H. Baek, H. Ikoma, D. S. Jeon, Y. Li, W. Heidrich, G. Wetzstein, and M. H. Kim, "End-to-end hyperspectral-depth imaging with learned diffractive optics," *arXiv preprint arXiv:2009.00463*, 2020.
[40] C. A. Metzler, H. Ikoma, Y. Peng, and G. Wetzstein, "Deep optics for single-shot high-dynamic-range imaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1375–1385.
[41] F. Heide, S. Diamond, M. Nießner, J. Ragan-Kelley, W. Heidrich, and G. Wetzstein, "Proximal: Efficient image optimization using proximal algorithms," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–15, 2016.
[42] B. Wohlberg, "SParse Optimization Research COde (SPORCO)," Software library available from http://purl.org/brendt/software/sporco, 2016.
[43] "Mocap solutions markers," https://mocapsolutions.com/, MOCAP SOLUTIONS, 2020.
[44] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
[45] B. Wohlberg, "SPORCO: A Python package for standard and convolutional sparse representations," in *Proceedings of the 15th Python in Science Conference*, Austin, TX, USA, Jul. 2017, pp. 1–8.
[46] ——, "Convolutional sparse representation of color images," in *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, Mar. 2016, pp. 57–60.
[47] ——, "Boundary handling for convolutional sparse representations," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1833–1837.
[48] C. Garcia-Cardona and B. Wohlberg, "Convolutional dictionary learning: A comparative review and new algorithms," *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, pp. 366–381, Sep. 2018.
[49] A. Liberzon, D. Lasagna, M. Aubert, P. Bachant, J. Borg *et al.*, "Openpiv/openpiv-python: Updated pyprocess with extended area search method," 2016.
[50] M. Wei, N. Sarhangnejad, Z. Xia, N. Gusev, N. Katic, R. Genov, and K. N. Kutulakos, "Coded two-bucket cameras for computer vision," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 54–71.
[51] C. Metzler et al., "Deep optics for single-shot high-dynamic-range imaging," in *CVPR 2020*.

Mark Sheinin is a Post-doctoral Research Associate at Carnegie Mellon University's Robotic Institute at the Illumination and Imaging Laboratory. He received his Ph.D. in Electrical Engineering from the Technion - Israel Institue of Technology in 2019. His work has received a Best Student Paper Award at IEEE CVPR 2017. He is the recipient of the Porat Award for Outstanding Graduate Students, the Jacobs-Qualcomm Fellowship in 2017, and the Jacobs Distinguished Publication Award in 2018. His research interests include computational photography and computer vision.

Matthew O'Toole is an Assistant Professor with the Robotics Institute and Computer Science Department at Carnegie Mellon University (CMU). He earned his Ph.D. from the University of Toronto and was previously a Banting postdoctoral fellow with the Computational Imaging group at Stanford University. He is the recipient of the ACM SIGGRAPH 2017 Outstanding Thesis Honorable Mention award, and runner-up best paper awards at CVPR 2014 and ICCV 2007.

Srinivasa G. Narasimhan is the Interim Director and Professor of the Robotics Institute at Carnegie Mellon University. He obtained his PhD from Columbia University in Dec 2003. His group focuses on novel techniques for imaging and illumination to enable applications in vision, graphics, robotics, agriculture, intelligent transportation and medical imaging. His works have received 10 Best Paper or Best Demo or Honorable mention awards at major conferences [ICCV (2013), CVPR (2019, 2015, 2000), ICCP (2020, 2015, 2012), I3D (2013), CVPR/ICCV Workshops (2007, 2009)]. In addition, he has received the Ford URP Award (2013), Okawa Research Grant (2009) and the NSF CAREER Award (2007). His research has been covered in popular press including NY Times, BBC, PC magazine and IEEE Spectrum and is highlighted by NSF and NAE. He is the co-inventor of programmable headlights, Aqualux 3D display, Assorted-pixels, Motion-aware cameras, Episcan360, Episcan3D, EpiToF3D, and programmable triangulation light curtains. He co-chaired the International Symposium on Volumetric Scattering in Vision and Graphics in 2007, the IEEE Workshop on Projector-Camera Systems (PROCAMS) in 2010, and the IEEE International Conference on Computational Photography (ICCP) in 2011, co-edited a special journal issue on Computational Photography, and serves on the editorial board of the International Journal of Computer Vision and frequently as Area Chair of top computer vision conferences (CVPR, ICCV, ECCV, BMVC, ACCV).