Diffraction Line Imaging

Mark Sheinin(✉), Dinesh N. Reddy, Matthew O'Toole, and Srinivasa G. Narasimhan

Carnegie Mellon University, Pittsburgh, PA 15213, USA
marksheinin@gmail.com

Abstract. We present a novel computational imaging principle that combines diffractive optics with line (1D) sensing. When light passes through a diffraction grating, it disperses as a function of wavelength. We exploit this principle to recover 2D and even 3D positions from only line images. We derive a detailed image formation model and a learning-based algorithm for 2D position estimation. We show several extensions of our system to improve the accuracy of the 2D positioning and expand the effective field of view. We demonstrate our approach in two applications: (a) fast passive imaging of sparse light sources like street lamps, headlights at night and LED-based motion capture, and (b) structured light 3D scanning with line illumination *and* line sensing. Line imaging has several advantages over 2D sensors: high frame rate, high dynamic range, high fill-factor with additional on-chip computation, low cost beyond the visible spectrum, and high energy efficiency when used with line illumination. Thus, our system is able to achieve high-speed and high-accuracy 2D positioning of light sources and 3D scanning of scenes.

Keywords: Line sensor · Diffraction grating · 3D sensing · Motion capture · Computational imaging

1 Introduction

Artificial light sources are widely used in computer vision. Whether observed directly (Fig. 1[a–b]) or indirectly (Fig. 1[c]), artificial lights act as strong features to track [4,29], reconstruct [21,36], and interpret the scene and its objects. In this work, we rely on a key observation: these light sources occupy the image domain sparsely. But positioning sparse light sources with 2D sensors wastes pixel resources and limits the image acquisition rate.[1] Specifically, fast operation requires short exposures which leave most of the captured 2D image pixels completely dark (see Fig. 1[c]). Thus, most of the system's bandwidth is wasted.

[1] Event-based cameras known as dynamic vision sensors [8,14] output changes in intensity on a per-pixel basis, but the prototype sensors have limited spatial resolution.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58536-5_1) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12347, pp. 1–16, 2020.
https://doi.org/10.1007/978-3-030-58536-5_1

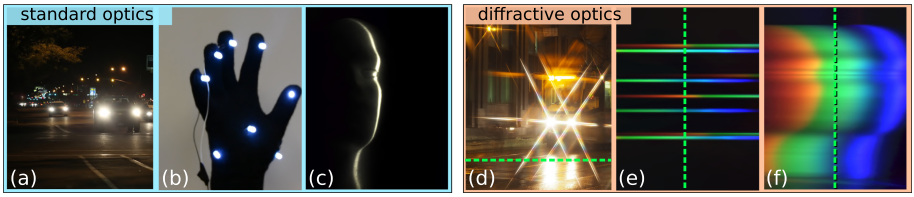

Fig. 1. When light sources or bright projections of light **(a-c)** are viewed through a transmissive diffraction grating, the incoming light is dispersed as a function of wavelength, creating streaks of light on the camera's sensor plane **(d-f)**. If a 1D color sensor (dotted green) or very few rows in a 2D sensor, intersects with these streaks, the measured colors can be used to efficiently determine the 2D spatial positions for each light source (or projected-line reflection) at high frame rates. (Color figure online)

Instead of using the full 2D sensor, we take a novel approach for saving bandwidth by imaging the scene using 1D (line) sensors.

Light passing through a diffraction grating is dispersed as a function of wavelength. When imaged by a 2D camera, the dispersed light manifests as colorful streaks. The micro-structure of the grating influences the shapes and number of streaks. And the brightness and appearance of the streaks depend on the 2D spatial locations of the sources in the image. Diffraction gratings are used to create artistic effects [16] (as in Fig. 1[d]), as well as in many imaging and scientific applications including spectroscopy [18,31], multi-spectral sensing [13,26], and rainbow particle velocimetry [35]. Unlike prior works, our method uses diffraction to encode the spatial position of scene light sources.

In this work, we introduce a novel class of imaging systems that use one or more diffraction gratings in conjunction with line (1D) sensors. We call this "Diffraction Line Imaging". We demonstrate how line sensors can yield 2D and even 3D information. Line sensors offer several advantages over 2D sensors: high frame rates, high dynamic range, and additional on-chip computation near the sensing area. Hence, this imaging approach results in fast 2D positioning and 3D reconstruction in many applications including night imaging, LED-based motion capture, and industrial product scanning.

We derive a detailed image formation model that maps a 3D point source to a 2D location on a virtual image plane. We then develop a learning-based algorithm to estimate the 2D locations of a sparse set of light sources or a line illumination projected onto a 3D scene. We numerically evaluate the uncertainty of the 2D positioning and the achieved field of view. To improve positioning accuracy significantly, the imaging system is extended to include multiple diffraction gratings and/or an additional cylindrical lens. Finally, we extend the approach to multiple line sensors (or multiple regions of interest in a 2D sensor) to increase the imaging system's field of view. Our approach can also be thought of as a variant of compressive sensing [2,3,32,34] with direct decoding of 2D positions, in place of computationally intensive and noise-prone decoding algorithms.

Our imaging systems are demonstrated in the following two applications.

Passive Imaging of Light Sources: Night scenes include a variety of light sources such as street lamps, vehicle headlamps, tail lights, turn signals, and bicycle lights (Fig. 1[a]). The flicker of sources can determine the AC phase of electrical circuits and even analyze power grids [27,28]. The glows around street lamps reveal the weather condition (fog, haze, rain) and visibility [19]. Finally, motion capture systems often attach light sources (LEDs) to estimate the object's motion in 3D (Fig. 1[b]) [22,24]. Our experiments show that we are able to estimate 2D light source positions from line images at high frame rates (up to 2220 fps in our camera). In contrast to previous works that use line sensors [22,24] or use spatio-temporal illumination to light a subject [15,25], our approach is based purely on passive observation of light sources without requiring special modulation or synchronization.

Structured Light Line Scanning: Structured light 3D scanning often projects or sweeps an illumination line across a scene [6]. A 2D camera observes the intersection of the line with the 3D scene. The 3D scene is then reconstructed by intersecting the pixel ray with the plane of illumination. We show how to accurately reconstruct the scene using a line light source and diffraction line imaging. Our image acquisition is mainly limited by the signal-to-noise ratio (SNR) (*i.e.*, exposure time) and not by the bandwidth. Interestingly, our system is the first instance of a structured light system with a 1D source and a 1D camera, since prior methods used either a 2D projector and/or a 2D camera [33]. Further, line illumination and imaging is significantly more energy efficient than 2D illumination and imaging [23]. Hence, bright light sources enable scan rates up to tens of thousands of lines per second, making this approach very useful in industrial/logistics scanning applications [5,11]. **See supplementary material for videos of results.**

2 Background on Diffraction Gratings

Our approach to light source positioning exploits *diffraction*: the wavelength-dependent optical phenomenon that causes light to bend (*i.e.*, diffract) when passing through and around obstacles and slits. A *diffraction grating* is an optical element that produces diffraction patterns, useful in scientific applications such as spectroscopy [18,31], holography [17], and hyperspectral imaging [13,26].

Figure 2(Left) shows a grating consisting of a periodic structure repeated every d microns. In a *transmissive* grating [17], incident light diffracts according to:

$$d[\sin(\theta_m) - \sin(\theta_i)] = m\lambda, \tag{1}$$

where θ_m is the angle of the m^{th} diffraction order, θ_i is the angle of incident light, and λ is the wavelength in microns.[2] In Eq. (1), $m = 0$ corresponds to the zeroth-order component which passes unbent through the grating, while each

[2] Eq. (1) assumes a collimated incident beam and an identical index of refraction for the medium on both sides of the grating.

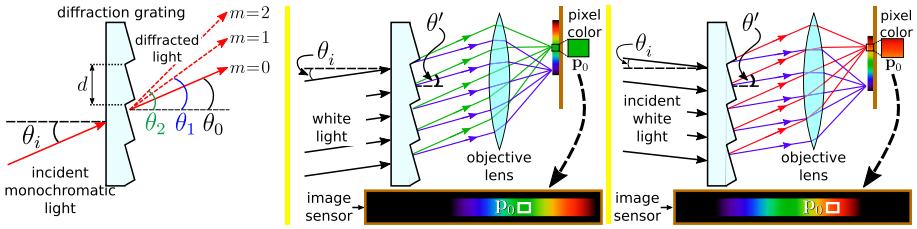

Fig. 2. Diffraction-based positioning. **Left:** A monochromatic collimated beam passes through a diffraction grating. The grating diffracts the beam yielding new beam directions according to Eq. (1). **Middle:** Per diffraction order m, each wavelength is diffracted in a different spatial direction. When imaged by a camera, this results in a horizontal rainbow pattern on the image plane. Here, pixel $\mathbf{p}_0$ measures the energy of some wavelength in the green range. **Right:** Shifting the incident angle θ_i results in a spatial shift of the spectral pattern on the image plane. Each θ_i maps a unique wavelength to $\mathbf{p}_0$. Thus, the color value at $\mathbf{p}_0$ provides information about θ_i. (Color figure online)

wavelength disperses into a unique angle for $m \neq 0$, producing the characteristic rainbow color streaks (see Fig. 1).

For a *fixed* θ_i, a camera imaging the exiting light maps each wavelength onto unique sensor position. This mapping is the basis for spectroscopy, whose purpose is the spectral analysis of the incident light. In contrast, we propose to *invert* this process, using color to efficiently and precisely recover the incident direction of light θ_i for unknown light sources.

Now, consider light exiting the diffraction grating at a fixed angle θ' belonging to the first diffraction order (*i.e.*, $m = 1$). Suppose that on the camera image plane, light exiting at θ' is focused at some camera pixel $\mathbf{p}_0$, as illustrated in Fig. 2(Middle). Then, from Eq. (1), the color (wavelength) measured at $\mathbf{p}_0$ is:

$$\lambda(\theta_i) = d[\sin(\theta') - \sin(\theta_i)], \tag{2}$$

and depends on the incident light direction θ_i. Thus, for a *fixed* θ', the measured color at $\mathbf{p}_0$ indicates the direction of incident light.

3 Diffraction-Based 2D Positioning

The diffraction equation (Eq. (2)) has two key properties that enable computing the direction of scene light sources. First, the function $\lambda(\theta_i)$ is an injective (*i.e.*, one-to-one) function across the domain $\theta_i \in [-pi/2, \theta']$. The inverse of $\lambda(\theta_i)$ is therefore well-defined and given by

$$\theta_i(\lambda) = \arcsin\left(\sin(\theta') - \frac{\lambda}{d}\right). \tag{3}$$

Second, Eq. (2) and its inverse do not depend on the intensity of light. This property makes our imaging system effectively invariant to the incident light intensity

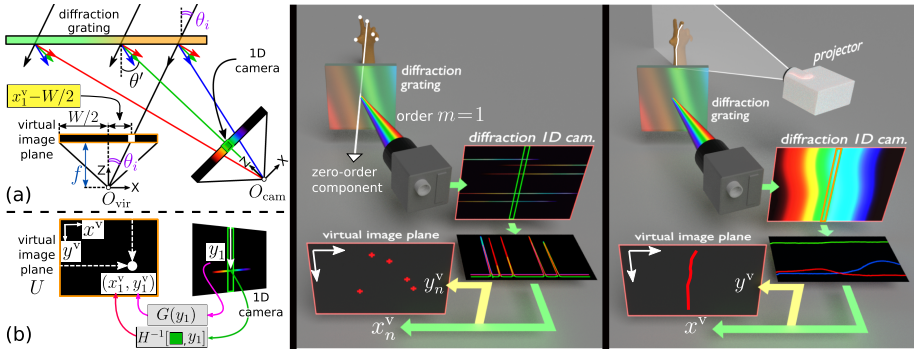

Fig. 3. Positioning system schematic. The first diffraction order is imaged by a RGB line scan camera, yielding 2D light source positions based on the measured color. **Left**: System schematic with **(a)** Top-view and **(b)** Front-view. **Middle**: Sparse light-sources yield intensity spikes on the line scan camera (green rectangle). Spike location along the vertical line scan camera encodes the vertical coordinate in the virtual image plane, while the spike color encodes the horizontal coordinate. **Right**: Line projection 3D scanning. The projected line yields a piece-wise continuous signal in the 1D camera. (Color figure online)

and, more specifically, its spectral characteristics (see derivation in Sect. 3.1). Note that this assumes no under- and over-saturated pixel measurements.

A basic schematic of our optical system is shown in Fig. 3(Left). The system consists of a diffraction grating and a color line scan camera. The line scan camera is positioned vertically, containing pixels along the Y-direction (in the camera's reference frame) denoted by the single coordinate y. We use the terms 'line scan camera' and '1D camera' interchangeably. For ease of explanation, we sometimes refer to the line scan camera's image plane as the full 2D image, had the camera been equipped with a standard 2D sensor.

The proposed system can then track both 2D spatial coordinates. For each point source, the vertical source coordinate is trivially measured by the rainbow's streak position along the 1D vertical sensor (*i.e.*, y_1 in Fig. 3[Left]b). Computing the source's horizontal coordinate amounts to computing its θ_i angle, which involves three simple steps: (1) measure the response in RGB space, (2) compute the dominant wavelength λ that produces the RGB signal, and (3) evaluate the inverse function $\theta_i(\lambda)$. Note that this procedure requires the light sources to be sufficiently broadband, such that the RGB measurement is non-zero for the given incident angle θ_i and the corresponding wavelength $\lambda(\theta_i)$.

As mentioned in the introduction, we tackle two types of imaging regimes. (a) *Sparse light sources*: scenes having a sparse set of point sources distributed across the image plane (Fig. 3[Middle]), and (b) *Structured light line scanning*: scenes illuminated by a vertically projected line, which yields a vertical curve on the image plane (Fig. 3[Right]). The projected line (*i.e.*, plane in space) in (b) can either be swept across a static object by a projector, or can have a fixed direction while the object moves through the plane (*e.g.*, by placing the object on a conveyor belt or turn table).

3.1 Image Formation Model

We derive the model which connects the projection of light from 3D positions in space onto a 2D virtual camera image plane. The line scan camera, modeled as a pinhole, is set to image the first-order diffracted light from the scene. Let O_{cam} denote the line scan camera's position and orientation. As shown in Fig. 3(Left), we define a virtual image plane U with pixel coordinates $\mathbf{p}^{\text{v}} = (x^{\text{v}}, y^{\text{v}})$, that belong to a virtual 2D camera positioned at O_{vir}.

For simplicity, we begin by describing the model for a single point source indexed by $n=1$, having homogeneous world coordinates $\mathbf{w}_1 \equiv [X, Y, Z, 1]^T$. On the line scan camera's image plane, the point source creates a holographic image in the shape of a *horizontal* rainbow line (see Figs. 1, 4). The camera's line sensor is orthogonal to the rainbow line, intersecting it over a few pixels centered at y_1. The geometric relationship between y_1 and virtual image coordinate y_1^{v} is:

$$y_1^{\text{v}} \equiv G(y_1). \tag{4}$$

Neglecting distortions that arise from light entering at highly oblique angles [10], the rainbow line's y_1 coordinate is given by standard projective geometry:

$$\gamma \begin{pmatrix} y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{P}\mathbf{w}_1, \tag{5}$$

where $\mathbf{P}$ is the camera's projection matrix and γ is an arbitrary scale factor [9]. In turn, since the Y-axes of both the virtual and line sensors are identical, G can be approximated by an affine transformation.

As seen in Fig. 3(a), the angle of the incident light with respect to the grating can be expressed as:

$$\theta_i(x_1^{\text{v}}) = \arctan([x_1^{\text{v}} - W/2]/f), \tag{6}$$

where f the virtual camera's focal length in pixel units and W is the virtual image plane width. Combining this with Eq. (3) yields:

$$\lambda(x_1^{\text{v}}) = d\left(\sin(\theta') - \sin[\arctan([x_1^{\text{v}} - W/2]/f)]\right). \tag{7}$$

The RGB intensity measured by the camera is modeled as:

$$\mathbf{I}_\sigma(y_1) = T\mathbf{c}_\sigma[\lambda(x_1^{\text{v}})]\ s[\lambda(x_1^{\text{v}})], \tag{8}$$

where T is the exposure time, $\mathbf{c}_\sigma(\lambda)$ is the camera's spectral response function [grayscale/Joule] for every color channel $\sigma \in \{R, G, B\}$, and $s(\lambda)$ represents the source's spectral radiant flux [Joule/sec] falling on y_1. In Eq. (8) we assumed that each pixel y integrates a very narrow band of wavelengths due to its the very narrow field of view.

Normalizing the 3x1 vector $\mathbf{I}_\sigma(y_1)$ by its L^2 norm removes the dependence on exposure time and source spectral flux:

$$\bar{\mathbf{I}}_\sigma(y_1) = \frac{\mathbf{I}_\sigma(y_1)}{||\mathbf{I}_\sigma(y_1)||_2} = \frac{\mathbf{c}_\sigma[\lambda(\theta', x_1^{\text{v}})]}{||\mathbf{c}_\sigma[\lambda(\theta', x_1^{\text{v}})]||_2} \equiv H_\sigma(x_1^{\text{v}}). \tag{9}$$

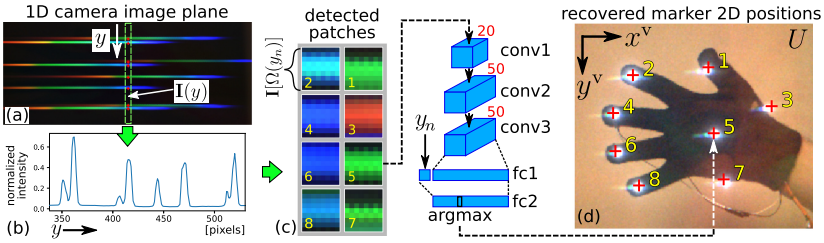

Fig. 4. Recovering sparse point sources. **(a)** Rainbow streaks incident on the line scan camera image plane. **(b)** The vertical rainbow locations along the line scan camera y_n are computed by detecting peaks in the averaged grayscale intensity along y. **(c)** For each detected peak y_n, a small 8×9 patch is extracted within sensor's narrow region of interest (ROI). **(d)** Each patch and its corresponding y_n is processed by a CNN to obtain x_n^{v}. Coordinate y_n^{v} is computed directly from y_n using Eq. (4). The 2D glove image from the helper camera is shown for reference only.

Finally, x_1^{v} is given by inverting Eq. (9):

$$x_1^{\mathrm{v}} = H^{-1}[\bar{\mathbf{I}}_\sigma(y_1)]. \tag{10}$$

The model in Eqs. (6–10) assumes that only a single source is predominately projected onto every line-camera pixel. Recovering x_1^{v} is described next.

3.2 Learning to Recover Horizontal Coordinates

Computing H^{-1} in practice requires accounting for various factors—sources not truly at infinity, sources having a small (non-point) projected surface area, sensor-diffraction grating geometry, the line scan camera's radiometric response, sensor saturation, and more. And, due to camera lens distortions, sensor-diffraction grating misalignment, and deviations to Eq. (1) due to incident light in oblique angles, H^{-1} generally depends on y_1 as well.[3] These factors make computing H^{-1} directly a laborious task. Instead, we adopt a learning-based approach.

We train a neural network to approximate H^{-1}, denoted by $\hat{H}^{-1}$. The network receives sensor color measurements in RGB space along with y, termed RGBy, and outputs the x^{v} coordinates. We tailored two different network architectures for our two recovery regimes: sparse points and vertical line projection.

Recovering Sparse Point Sources. For sparse sources, we train a convolutional neural network (CNN). The network, denoted by H_{point}^{-1}, maps RGBy values from a small patch into a predefined discrete set of possible x^{v} coordinates (see Fig. 4). Our network consists of three convolutional layers (channel widths 20, 50, 50 respectively) followed by two fully connected layers; see Fig. 4. Consider a scene with N point sources, indexed by $n = 1, 2, \ldots, N$. Let $\mathbf{I}(y)$

[3] Equation (1) strictly holds when the grating groves are perpendicular to the incidence plane. The incidence plane is the plane containing the beam direction and the normal to diffraction grating plane.

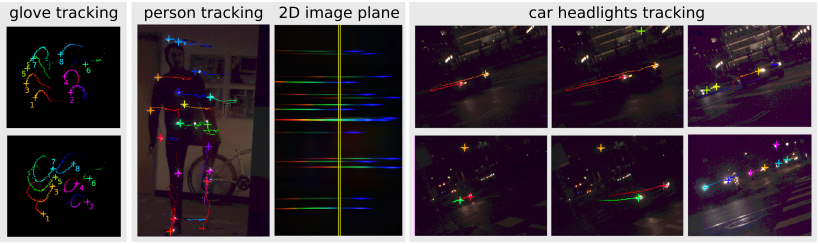

Fig. 5. Detection and tracking of light sources on a fast moving glove fitted with 8 LEDs, a suit fitted with 18 LEDs and headlamps of vehicles at multiple road intersections. The low-exposure background image from the helper camera is only shown to aid visualization. See supplementary material for videos and tracker details. (Color figure online)

denote the $8 \times Q$ color image from our line scan camera. Here, 8 denotes the number of image columns, since line scan cameras can be a few pixels wide in the horizontal direction as well (*e.g.*, due to RGB Bayer color filter).

The first step is to identify the rainbow line positions from peaks in $\mathbf{I}(y)$. Then, for every detected rainbow with coordinate y_n, we provide the CNN with a normalized small 8×9 image patch $\Omega(y_n)$ vertically centered at y_n, and concatenate the coordinate y_n with the input to the first fully connected layer.[4] The network outputs a $W \times 1$ vector $\mathbf{s}_n$ with scores for every possible horizontal virtual image location x^{v}. Then, x_n^{v} are recovered as:

$$x_n^{\mathrm{v}} = \arg\max(S[\mathbf{s}_n]), \tag{11}$$

where S is a softmax function.

The training dataset consists of ground truth 2D coordinates captured with a 2D helper camera, line scan camera coordinates y_m, and the corresponding image patches $\bar{\mathbf{I}}[\Omega(y_m)]$:

$$\left\{x_m^{\mathrm{v,GT}}\right\}_{m=1}^{M} \longleftrightarrow \left\{\bar{\mathbf{I}}[\Omega(y_m)], y_m\right\}_{m=1}^{M}, \tag{12}$$

where M is the number of training examples. The training loss is given by:

$$L = \frac{1}{M}\sum_{m=1}^{M} \mathrm{BCE}(S[\mathbf{s}_m], D[x_m^{\mathrm{v,GT}}, \sigma]), \tag{13}$$

where $BCE(\cdot,\cdot)$ is the Binary Cross Entropy function and function $D(\cdot)$ generates a Gaussian probability distribution with mean $x_m^{\mathrm{v,GT}}$ and standard deviation σ. Intuitively, the Binary Cross Entropy function drives the output distribution $S[\mathbf{s}_m]$ to match a narrow Gaussian centered at $x_m^{\mathrm{v,GT}}$. Using a Gaussian instead of a Delta function for $D(\cdot)$ provides a tolerance for small deviation due to image noise. See additional results in Fig. 5.

[4] Normalizing a 8×9 consists of dividing all RGB pixel values by a scalar. For example, for L^{∞}, $\bar{\mathbf{I}}[\Omega(y_m)] = \mathbf{I}[\Omega(y_m)]/\max_{\mathrm{RGB}}$, where $\max_{\mathrm{RGB}}$ is the maximum value across all patch pixels and color channels.

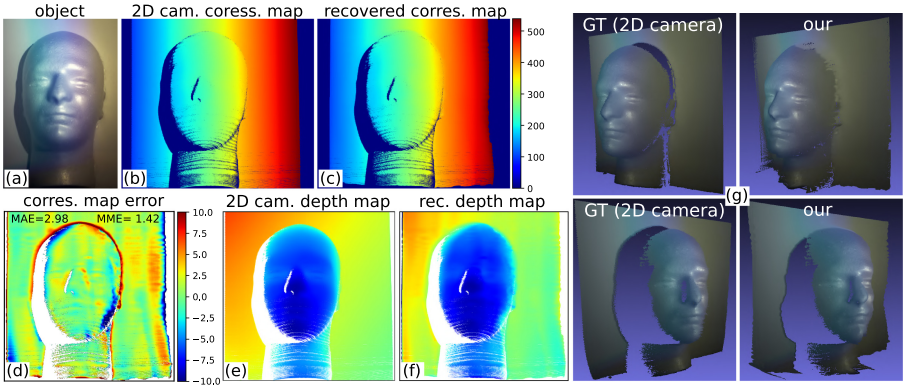

Fig. 6. Structured light with a 1D sensor and a 1D illumination. An object **(a)** is scanned using line projection. The recovered lines using a 2D camera (estimated ground truth) **(b)** and our method **(c)** are visualized with a correspondence map, in which each color indicates the projected column index. **(d)** Correspondence map error. Depth maps recovered using a 2D camera **(e)** and our method **(f)**. Displayed range 50.6 mm to 71.1 mm. **(g)** Recovered dense point clouds. We believe that this is the first instance of structured light scanning with both a 1D sensor and 1D camera. (Color figure online)

Recovering a Vertically Projected Line. As detailed in Sect. 7, 3D reconstruction relies on an aggregate of many continuous measurements through interpolation. Thus, here the network implements regression. Let the 3×1 RGB vector $\mathbf{u}_y$ denote the mean color of patch $\Omega(y)$. Namely, $\mathbf{u}_y$ is computed by averaging $\Omega(y)$ over the spatial domain.

The network H_{line}^{-1} is fed with a normalized $\mathbf{u}_y$ along with y, and outputs a continuous scalar approximation for x^{v}:

$$x^{\text{v}} = \begin{cases} H_{\text{line}}^{-1}\left(\frac{\mathbf{u}_y}{||\mathbf{u}_y||_2}, y\right), & \text{if } \mathbf{u}_y \in \mathcal{A} \\ \text{none}, & \text{otherwise}, \end{cases} \tag{14}$$

where $\mathcal{A}$ is a subspace of 'valid' RGB input values such that

$$\mathcal{A} = \{\mathbf{u}_y : ||\mathbf{u}_y||_{\text{inf}} > t,\ ||\mathbf{u}_y||_0 > 1\},$$

where t is a predefined intensity threshold. Subspace $\mathcal{A}$ excludes low intensity (thus noisy) measurements as well as wavelengths that map to only a single color channel and thus can't be distinguished after normalization. The latter occurs at the edges of the rainbow streak (*i.e.*, deep reds and blues).

The network consists of four fully connected layers of size 300, and is trained using the Huber loss [12]. The training set is $\left\{x_m^{\text{v,GT}}\right\}_{m=1}^{M} \longleftrightarrow \{\mathbf{u}_m, y_m\}_{m=1}^{M}$, and was obtained by scanning a vertical line on a white wall. See Sect. 6 for more calibration details and Fig. 6 for example result.

4 Expanding the FOV Using Multiple ROIs

Our original prototype's readout was set to a single ROI consisting of a few columns (rows in a 90° rotated camera) at the center of the camera's image

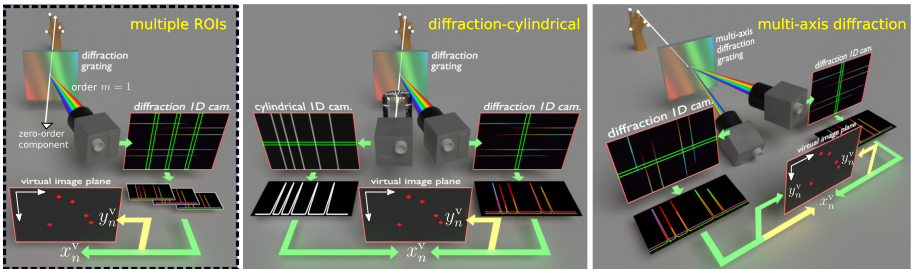

Fig. 7. Left: Using a 2D camera with multiple ROIs (three here) extends the system's FOV by 'catching' streaks outside the central ROI's domain. **Middle**: The additional left camera images the zeroth-order diffracted light through a cylindrical lens which focuses the sources onto horizontal white streaks. The intersection of the white streaks with the left line sensor yields additional horizontal point coordinate measurements which improve source positioning. **Right**: A multi-axis diffraction grating additionally yields horizontal rainbow streaks. Capturing these streaks using an additional line sensor mitigates the vertically-overlapping sources ambiguity.

plane. However, reading multiple ROIs from different spatial locations on the sensor, as shown in Fig. 7(Left), increases the horizontal FOV without sacrificing acquisition speed too much. This is because multiple ROIs can catch rainbow streaks whose visible spectrum does not intersect with the center column. Multiple ROIs also reduce recovery uncertainty since they may increase the signal per point, when the point's rainbow streak has sufficient signal over several ROIs.

Let $\mathbf{I}_\sigma^r$ denote the camera image from ROIs indexed $r=1,2,..,R$. For sparse points, we concatenate the R patches $\mathbf{I}_\sigma^r[\Omega(y_n)]$ from y_n to form an extended $8\times(9R)$ color patch and feed it to the network along with y_n. For vertical line projection, we similarly concatenate the RGB measurements $\mathbf{u}_y^r$ and feed the resulting $(3R+1)\times 1$ vector to the network. As in Eq. (14), only 'valid' y's are considered, where now a measurement at y is valid if any one of the R terms $\mathbf{u}_y^r$ is valid. To preserve the network's invariance to object albedo, we augment each individual $\mathbf{u}_y^r$ during training (before concatenation) by a random scale factor, followed by adding simulated sensor noise. Figure 6 shows a result where $R=5$. See supplementary for FOV analysis.

5 Reducing Sparse Point Uncertainty

The system described so far has two limitations: (1) uncertainty in the horizontal coordinates is typically higher than the vertical one, and (2) ambiguities in position estimates for certain source alignments (vertically overlapping sources). Here we describe two hardware variations that mitigate these limitations, for the sparse point case: (1) using two line sensors with a diffraction grating and a cylindrical lens, respectively (Sect. 5.1), and (2) using two line sensors with a double-axis diffraction grating (Sect. 5.2).

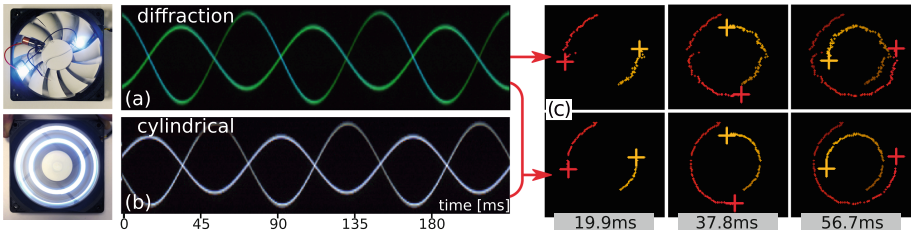

Fig. 8. High speed operation. A fan is mounted with two LEDs at different radii and is rotated at 780 RPM. A diffraction-cylindrical system (Sect. 5.1) images the fan at 2220 FPS. **(a)** Spectral space-time diagram for the diffraction camera where columns show the 1D signal (along the rows) at different times. **(b)** Space-time diagram for the additional cylindrical lens camera. **(c)** Recovered source positions and trajectories at three times within the first revolution. Top row shows results using only the diffraction signal. Bottom row shows the improvement by incorporating cylindrical measurements.

5.1 Horizontal Cylindrical Lens

To improve the accuracy of x^{v}, we place an additional *horizontal* line scan camera to image the zero-order component through a cylindrical lens (see Fig. 7[Middle]). The cylindrical lens focuses scene points onto vertical lines which intersect the horizontal line sensor. These white streaks yield precise measurement for coordinates x^{v}. However, unlike the rainbow streaks, the white streaks provide no information about y^{v}. Nevertheless, this data can improve 2D positioning.

Merging the recovered coordinates from the cylindrical lens sensor is fully detailed in the supplementary material. Here we describe the basic idea. Let $\tilde{x}^{\mathrm{v}}_k$ denote the recovered cylindrical lens coordinates, indexed by $k = 1, 2, .., K$. For every recovered point $(x^{\mathrm{v}}_n, y^{\mathrm{v}}_n)$, we compute the distance of its x^{v}_n coordinates to all $\tilde{x}^{\mathrm{v}}_k$. If for any k, $\tilde{x}^{\mathrm{v}}_k$ is very 'close' to x^{v}_n (*e.g.* below four pixels), we assume that measurement k originated from point n, and thus replace $x^{\mathrm{v}}_n \leftarrow \tilde{x}^{\mathrm{v}}_k$. Otherwise, we discard $(x^{\mathrm{v}}_n, y^{\mathrm{v}}_n)$ as an inaccurate measurement. See Fig. 8 for example result.

5.2 Double-Axis Diffraction

The transmissive grating of Sect. 2 has spatial variation in only one axis, thus it diffracts light along said axis. A *double-axis* grating has spatial variation in both axes (*i.e.*, a star filter) and thus diffracts light both horizontally and vertically. Using a double-axis grating allows for a direct generalization of Sect. 3.

We replace the single-axis grating in Fig. 7(Left) with a double-axis grating and add an additional line sensor camera to image the *vertical* first-order diffraction component as well (see Fig. 7[Right]). The vertical diffraction component creates a vertical rainbow streak on the second line sensor camera (which now requires no tilting). Each line sensor now provides a separate measurement for $(x^{\mathrm{v}}, y^{\mathrm{v}})$. Merging these pair of measurements follows the same logic as in Sect. 5.1. Namely, if two points fall within the predefined distance on the virtual image plane, they are assumed to originate from the same scene point and are

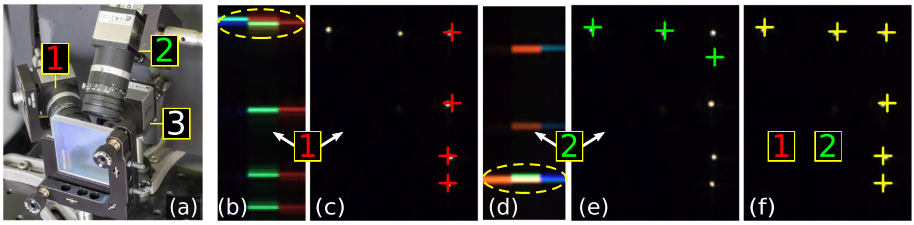

Fig. 9. A double-axis diffraction system with 3-ROIs **(a)** yields signals from both vertical (cam 1) **(b)**, and horizontal (cam 2) **(d)** diffraction. Recovered points from the vertical **(c)** and horizontal **(e)** diffraction superimposed on a 2D helper camera (ground truth, cam 3). In **(c)**, the three top points share the same y and thus not fully recovered. Similarly, in **(e)** the right vertical points share the same x and thus are not fully recovered. **(f)** Merging both measurements yields correct positions for all points.

merged by taking y^{v} from the vertical sensor, while taking x^{v} from the horizontal one. See the supplementary for details. See example result in Fig. 9.

6 Hardware and Calibration

Our prototype is shown in Fig. 9(a). The setup consists of three IDS UI-3070CP-C-HQ Rev.2 color cameras, which support up to 64 ROIs. Camera 1 and 2 capture the vertical and horizontal diffraction streaks, respectively. Camera 3 is the helper camera, used for gathering ground-truth training data and visualizations. Camera 3 is also used in Sect. 5.1 in conjunction with a cylindrical lens. At least two rows are required per ROIs to yield color measurements (due to Bayer filter). Camera 1 is rotated by 90° to obtain vertical 1D measurements. Cameras 1 and 2 were stripped of their built-in IR filter and mounted with Fujinon 1.5 MP 9 mm lenses. Camera 3 was mounted with a 8 mm M12 lens using an adapter. We used Thorlabs 50 mm 1200 grooves/mm transmission gratings (GT50-12). The double-axis diffraction system had two stacked gratings, where one is rotated 90° with respect to the other. Otherwise, a single grating was used. Our motion-capture suit and glove prototypes are fitted with 5 mm 12V white diffuse LEDs. A second glove was fitted with 20 mm 3.3V white LEDs. For 3D scanning, we used an InFocus IN124STa Short Throw DLP Projector.

Calibration: Calibration is done using camera 3, whose image plane serves as the virtual plane in Fig. 3. For sparse points, we simultaneously recorded about 40K samples of a single moving source using all cameras. For sources, we used three types of LEDs (two clear one diffuse). This procedure yields the training set of Eq. (12), and is used to compute H^{-1} and G. For 3D scanning, we sweep and image a vertical line on a white wall. For each projected line, we compute its line equation $y^{\mathrm{v}} = cx^{\mathrm{v}} + d$ and use it to determine the ground truth disparity $x^{\mathrm{v,GT}}(y) = [G(y) - d]/c$. Together, all lines yield about 700K RGBy/position samples. Helper-camera and projector intrinsic and extrinsic calibration is done using a checkerboard [9].

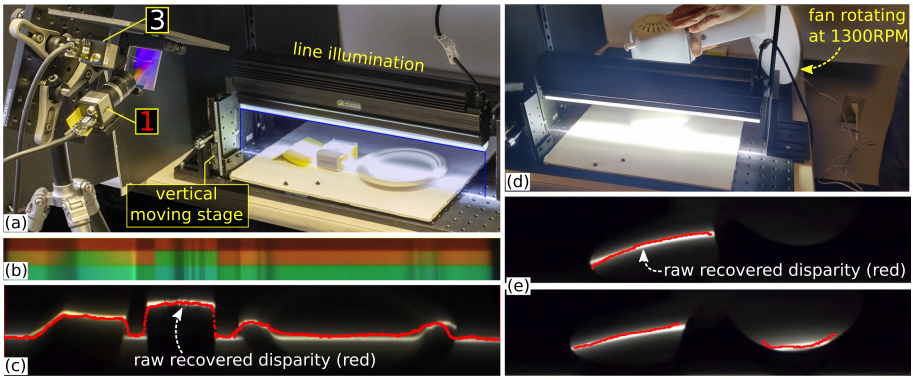

Fig. 10. Fast line-illumination scanning. **(a)** The experimental setup scans objects moving through the illuminated area, *e.g.*, for in-line inspection of objects on a conveyor belt. The bright source enables very fast scan speeds, up to 1743 scan lines per second. **(b)** The measured signal of scene (a) from the three ROIs. **(c)** The raw recovered disparity from (a), superimposed on the ground truth helper camera. **(d)** Fan rotating at 1300RPM. **(e)** Recovered disparity in two frames where the helper camera and our system temporally coincide. As shown in the supplementary video, the helper camera is 30x slower than our system and thus it is unable to capture the fan's motion.

7 Experimental Evaluation Details

Before using a neural network for H^{-1}, we tried various hand-crafted color-based methods (*e.g.*, mapping LAB or RGB values to position using optimization). The networks-based approach outperformed these methods. For sparse points, the mean absolute error (MAE) over the test set was 1.71 pixels with a 1.54 standard deviation. For line projection, MAE was 2.27 pixels with a 2.32 standard deviation. See supplementary for evaluation and network training details.

Point source detection and tracking is shown in Figs. 4, 5, and 8. Glove and person keypoints are accurately tracked for long durations even in the case of overlap in the diffraction readings. Figure 5(Right) shows tracking of headlights at multiple intersections. Observe that we are able to detect and track multiple light sources in the wild using a CNN trained on only three LEDs.

Figure 6 show 3D scanning using the setup shown in Fig. 3(Right). We used an off-the-shelf projector, and a 2D camera configured with five ROIs, each yielding a 8×2056 measurement. For each projected line, the algorithm yields up to 2056 continuous measurements, tracing that line in the virtual image. After imaging all lines, the measurements from all lines are used to interpolate the final correspondence map – a virtual camera image where each pixel is identified with one projected line index (or none). Then, 3D reconstruction follows from simple triangulation; see supplementary for more details.

In Fig. 6, the projector has limited contrast. Namely, when projecting a white vertical line, a significant amount of light was leaking to the 'dark' pixels outside the line. To compensate, we used longer exposures and averaged multiple frames

per line. In Fig. 6, we averaged 55 frames of 50ms exposure each. Additionally, we captured a black projector image prior to scanning and subtract it from all subsequently measurements.

Inter-reflections may degrade reconstruction quality by yielding a mixture of signal from multiple surface points on the same line. To effectively reduce their effect, we use high frequency illumination [20]. Specifically, we split each projected line into three high-frequency patterns, and extract the direct component as in [20]. High-frequency patterns are used in Fig. 6.

Figure 10 shows our experimental fast line illumination scanner. Applications for this setup include scanning products on rapidly moving conveyor-belts, or scanning objects rotated by a turntable. The camera was configured to readout three 8×2056 ROIs at 1743 FPS with an exposure of $300\,\mu s$. The system could compute disparity for fast moving objects (*e.g.*, fan rotating at 1300 RPM) captured under regular lighting conditions (with room and sunlight ambient light present). See supplementary for additional details.

8 Concluding Remarks

Diffraction line imaging is a novel computational imaging principle based on light diffraction and 1D sensing. Using the principle, we showed proof of concepts for applications like motion capture of uncontrolled and unmodulated LEDs, tracking car headlights, and 3D scanning using both 1D sensing and 1D illumination. Using line sensors significantly decreases bandwidth, which leads to speed. Speed is crucial for motion capture since it greatly eases tracking. In 3D scanning, speed is vital when scanning moving objects (*e.g.*, industrial conveyor belt).

Our prototype mimicked a line sensor using multiple rows from a conventional 2D sensor, resulting in fast readout rates of up to 2220 FPS. Faster line sensors [7] could reach up to 45,000 FPS (x20 faster than our prototype) and improve light efficiency with large pixels up to 4×32 microns in size (x10 larger). Conversely, using 2D sensors with multiple ROIs gives smooth control over the speed vs. quality trade-off, namely more ROIs reduce speed but increase accuracy.

As with any sensor though, our system's performance depends on the available SNR. Using hyper-spectral sensors may improve position decoding by raising the discrimination between signals from adjacent wavelengths. For 3D scanning, bright broadband sources, such as supercontinuum/swept-frequency lasers can additionally increase SNR many folds [1,30]. Learning-based or dictionary-based approaches may improve reconstruction quality by extracting multiple vertically overlapping points, which yield a linear color mixture on the 1D line scan sensor. Finally, we believe that our approach is an important step towards achieving a simple high-speed and low-cost solution to light source positioning with potential applications from vision to robotics.

Acknowledgments. We thank A. Sankaranarayanan and V. Saragadam for help with building the hardware prototype and S. Panev and F. Moreno for neural network-related advice. We were supported in parts by NSF Grants IIS-1900821 and CCF-1730147 and DARPA REVEAL Contract HR0011-16-C-0025.

References

1. Alfano, R.R.: The Supercontinuum Laser Source. Springer, Heidelberg (1989). https://doi.org/10.1007/b106776
2. Antipa, N.: Diffusercam: lensless single-exposure 3D imaging. Optica **5**(1), 1–9 (2018)
3. Antipa, N., Oare, P., Bostan, E., Ng, R., Waller, L.: Video from stills: lensless imaging with rolling shutter. In: Proceedings of IEEE ICCP, pp. 1–8 (2019)
4. Chen, Y.L., Wu, B.F., Huang, H.Y., Fan, C.J.: A real-time vision system for nighttime vehicle detection and traffic surveillance. IEEE Trans. Ind. Elect. **58**(5), 2030–2044 (2010)
5. Loadscan: load management solutions (2020). https://www.loadscan.com/
6. Curless, B., Levoy, M.: Better optical triangulation through spacetime analysis. In: Proceedings of IEEE ICCV, pp. 987–994 (1995)
7. Dlis2k: ultra configurable digital output (2020). http://dynamax-imaging.com/products/line-scan-product/dlis2k-2/
8. Gallego, G., et al.: Event-based vision: a survey (2019). arXiv preprint arXiv:1904.08405
9. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
10. Harvey, J.E., Vernold, C.L.: Description of diffraction grating behavior in direction cosine space. Appl. Opt. **37**(34), 8158–8159 (1998)
11. Hossain, F., PK, M.K., Yousuf, M.A.: Hardware design and implementation of adaptive canny edge detection algorithm. Int. J. Comput. Appl. **124**(9), 31–38 (2015)
12. Huber, P.J.: Robust estimation of a location parameter. In: Kotz, S., Johnson, N.L. (eds.) Breakthroughs in statistics. Springer Series in Statistics (Perspectives in Statistics), pp. 492–518. Springer, Heidelberg (1992). https://doi.org/10.1007/978-1-4612-4380-9_35
13. Jeon, D.S., et al.: Compact snapshot hyperspectral imaging with diffracted rotation. ACM TOG **38**(4), 117 (2019)
14. Kim, H., Leutenegger, S., Davison, A.J.: Real-time 3D reconstruction and 6-DoF tracking with an event camera. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 349–364. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_21
15. Kim, J., Han, G., Lim, H., Izadi, S., Ghosh, A.: Thirdlight: Low-cost and high-speed 3D interaction using photosensor markers. In: Proceedingd of CVMP, p. 4. ACM (2017)
16. Liu, D., Geng, H., Liu, T., Klette, R.: Star-effect simulation for photography. Comput. Graph. **61**, 19–28 (2016)
17. Loewen, E.G., Popov, E.: Diffraction Gratings and Applications. CRC Press, Boca Raton (2018)
18. Nagaoka, H., Mishima, T.: A combination of a concave grating with a Lummer-Gehrcke plate or an echelon grating for examining fine structure of spectral lines. Astrophys. J. **57**, 92 (1923)
19. Narasimhan, S.G., Nayar, S.K.: Shedding light on the weather. In: Proceedings of IEEE CVPR, pp. 665–672 (2003)
20. Nayar, S.K., Krishnan, G., Grossberg, M.D., Raskar, R.: Fast separation of direct and global components of a scene using high frequency illumination. In: ACM SIGGRAPH, pp. 935–944 (2006)

21. Nelson, P., Churchill, W., Posner, I., Newman, P.: From dusk till dawn: localisation at night using artificial light sources. In: Proceedings of IEEE ICRA (2015)
22. Optotrak certus (2020). https://www.ndigital.com/msci/products/optotrak-certus/
23. O'Toole, M., Achar, S., Narasimhan, S.G., Kutulakos, K.N.: Homogeneous codes for energy-efficient illumination and imaging. ACM TOG **34**(4), 1–13 (2015)
24. Phase space inc. (2020). http://www.phasespace.com/
25. Raskar, R., et al.: Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators. ACM TOG **26**(3), 36 (2007)
26. Saragadam, V., Sankaranarayanan, A.C.: KRISM: Krylov subspace-based optical computing of hyperspectral images. ACM TOG **38**(5), 1–14 (2019)
27. Sheinin, M., Schechner, Y.Y., Kutulakos, K.N.: Computational imaging on the electric grid. In: Proceedings of IEEE CVPR, pp. 2363–2372 (2017)
28. Sheinin, M., Schechner, Y.Y., Kutulakos, K.N.: Rolling shutter imaging on the electric grid. In: Proceedings of IEEE ICCP, pp. 1–12 (2018)
29. Tamburo, R., et al.: Programmable automotive headlights. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 750–765. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_49
30. Vasilyev, A.: The optoelectronic swept-frequency laser and its applications in ranging, three-dimensional imaging, and coherent beam combining of chirped-seed amplifiers. Ph.D. thesis, Caltech (2013)
31. Vogt, S.S., et al.: HIRES: the high-resolution echelle spectrometer on the Keck 10-m Telescope. In: Instrumentation in Astronomy VIII, vol. 2198, pp. 362–375. International Society for Optics and Photonics (1994)
32. Wang, J., Gupta, M., Sankaranarayanan, A.C.: Lisens-a scalable architecture for video compressive sensing. In: Proceedings of IEEE ICCP, pp. 1–9. IEEE (2015)
33. Wang, J., Sankaranarayanan, A.C., Gupta, M., Narasimhan, S.G.: Dual structured light 3D using a 1D sensor. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 383–398. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_23
34. Weinberg, G., Katz, O.: 100,000 frames-per-second compressive imaging with a conventional rolling-shutter camera by random point-spread-function engineering. arXiv preprint arXiv:2004.09614 (2020)
35. Xiong, J., et al.: Rainbow particle imaging velocimetry for dense 3D fluid velocity imaging. ACM TOG **36**(4), 36 (2017)
36. Zhi, T., Pires, B.R., Hebert, M., Narasimhan, S.G.: Deep material-aware cross-spectral stereo matching. In: Proceedings of IEEE CVPR, pp. 1916–1925 (2018)