Diffraction Line Imaging:
Supplementary Document

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

Carnegie Mellon University, Pittsburgh, PA 15213, USA

1 High-intensity Line Illumination System

In the main manuscript, we presented two 3D structured light systems that use line illumination: *(a)* a projector-based system where a line (plane is space) is swept over the scene, and *(b)* a static line scanner where objects are scanned by moving through the illuminated area. Here we provide additional technical details and results for the static line scanner experiment.

Fig. 10(a) of the main manuscript (reproduced here as Fig. 1[a]) shows our experimental static line scanner. We placed the line light approximately 7.5cm above a flat surface. The line light is an Advanced Illumination LL167 high intensity white line light. The camera system images the objects under scan. The camera was configured to readout three 8 × 2056 ROIs, and was set to operate at 1743 FPS with an exposure of 300us. The helper camera was set to the standard 60 FPS.

The system was calibrated by vertically moving a white planar object. For each planar position, we simultaneously recorded the diffracted signal along with the resulting projected line (using the helper camera). Ground truth disparity is computed similarly to the projector line calibration detailed in Section 6 of the main manuscript. Fig. 1(b) shows the measured signal from the three ROIs. The image in Fig. 1(b) is vertically stretched for better visualization. Figs. 1(c)-(e) show the raw recovered disparity from several objects superimposed on the helper camera image. All measurements here were captured at a rate of 1743 FPS, in regular lighting conditions (with room and sunlight ambient light present).

The supplementary video shows a scanning of fan rotating at 1300 RPM. The fan is simultaneously imaged using our system and the 2D helper camera for visualization. Here, the helper camera is not perfectly synchronized to the diffraction camera. Therefore, the disparity measurements might not match precisely due to a possible delay. As seen in the video, the 2D camera is unable to articulate the fan's motion due to its low capturing rate. The fan appears to rotate backwards due to aliasing in the sampling. On the other hand, our system is able to accurately capture the fan's motion.

As seen in Fig. 1e (middle apple), using a line illumination has a disadvantage over a laser beam or a projector. The latter can be approximated as a point source with respect to the scene and thus cause little to no vertical illumination overlap. However, the line illumination also illuminates parts of the object that

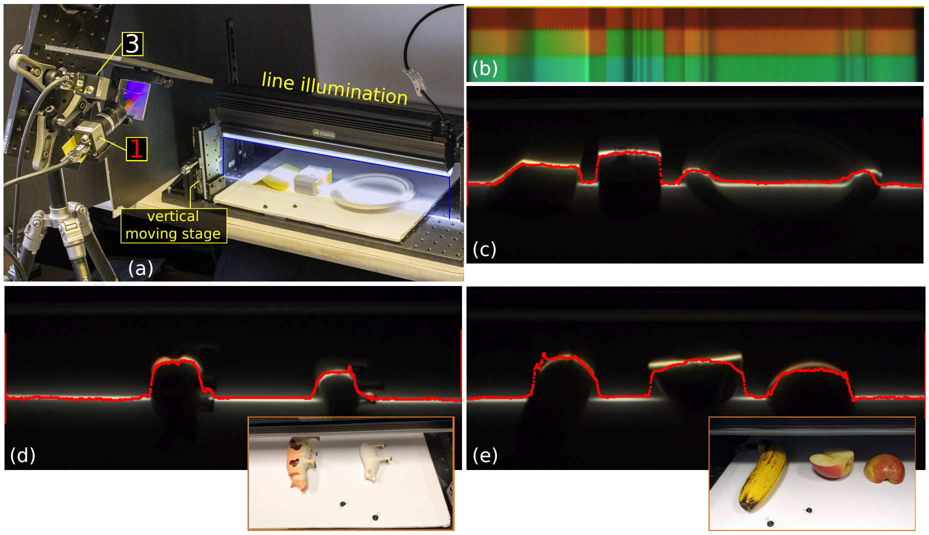

Fig. 1. Fast line-illumination scanning. The high intensity source enables very fast scan speeds, up to 1743 scan lines per second. **(a)** The experimental prototype, composed of a single diffraction system and a high intensity line illumination source. **(b)** The measured signal of scene (a) from three 8 × 2056 ROIs. The ROI image is vertically stretched for visualization. **(c)** The raw recovered disparity, superimposed on the ground truth helper camera. Observe the good correspondence between the recovered and ground truth disparity. **(d)-(e)** Additional scanning results.

would be shadowed when using a small point source, which may cause vertical color overlays at object edges and degrade disparity recovery quality. This can be mitigated by placing the objects on a black surface (instead of the white seen here).

2 Merging Measurement From Multiple Sensors

We propose using additional line sensors to improve point positioning accuracy. In this section, we describe how to merge the measurements captured by the multiple line sensors, used in our horizontal cylindrical lens system (Section 2.1) and the double-axis diffraction system (Section 2.2).

2.1 Horizontal Cylindrical Lens

Let $\tilde{x}$ denote the pixel coordinate of the cylindrical camera's image plane. Additionally, suppose that $\tilde{x} \equiv x^{\mathrm{v}}$, namely that the virtual camera's image plane horizontal axis and the cylindrical camera's image plane horizontal axis coincide. Denote the line image at coordinate $\tilde{x}$ by the pixel value $\mathbf{I}_{\mathrm{gray}}(\tilde{x})$. Here we assume a monochrome line image since no color information is required. Assuming no two scene points share the same x^{v}, detecting peaks on $\mathbf{I}_{\mathrm{gray}}(\tilde{x})$ yields N virtual image plane coordinates $\tilde{x}_k = \tilde{x}^{\mathrm{v}}_k$ indexed by $k = 1, 2, \ldots, N$.

Now, in addition to $(x^{\mathrm{v}}_n, y^{\mathrm{v}}_n)$ recovered by the diffraction camera, we have additional measurements $\tilde{x}^{\mathrm{v}}_k$ from this second camera. Note that the correspondences between indices n and k are generally unknown. Let

$$\mathcal{X} = \{\tilde{x}^{\mathrm{v}}_k | k = 1, 2, \ldots, N\} \tag{1}$$

be the set of all recovered coordinates from the camera with the cylindrical lens. We use $\mathcal{X}$ to improve x^{v}_n using the following reasoning. If point x^{v}_n is relatively accurate, there should exist a point in $\mathcal{X}$ which is *sufficiently* close to x^{v}_n. Then, this nearest neighbor in $\mathcal{X}$ is likely to belong to point n and can thus replace x^{v}_n, since generally $\tilde{x}^{\mathrm{v}}$ have less position uncertainty. Alternatively, if no point in $\mathcal{X}$ is sufficiently close to x^{v}_n, then there is a high probability that x^{v}_n is an inaccurate measurement and can thus be discarded.

Formally, for each point x^{v}_n, we solve this correspondence problem by computing

$$k_{\min} = \underset{k \in [1, N]}{\operatorname{argmin}} \, |x^{\mathrm{v}}_n - \tilde{x}^{\mathrm{v}}_k|, \tag{2}$$

Then, we perform the assignment

$$x^{\mathrm{v}}_n \leftarrow \tilde{x}^{\mathrm{v}}_{k_{\min}} \quad \text{if} \quad |x^{\mathrm{v}}_n - \tilde{x}^{\mathrm{v}}_{k_{\min}}| \leq E_{\mathrm{cyl}} \tag{3}$$

where $E_{\mathrm{cyl}} = 6$ is a predefined distance threshold. Points that do not meet this threshold are discarded. Fig. 8 of the main manuscript shows an example result comparing high-speed light source position recovery with and without the additional cylindrical lens camera. To mimic a cylindrical lens, we place a plano-convex cylindrical lens (Thorlabs N-BK7) in front of the helper camera's objective lens.

2.2 Double-axis Diffraction

Let $\hat{x}^{\mathrm{v}}_l, \hat{x}^{\mathrm{v}}_l$ denote the virtual image plane positions recovered by the horizontal diffraction grating sensor, where $l = 1, 2, .., L$. Due to horizontal or vertical point overlap, generally $L = N$ might not hold. We merge points from both sensors if they fall within a predefined distance $\mathcal{E}(l, n)$,

$$\mathcal{E}(l, n) \equiv \sqrt{(\hat{x}^{\mathrm{v}}_l - x^{\mathrm{v}}_n)^2 + (\hat{y}^{\mathrm{v}}_l - y^{\mathrm{v}}_n)^2}. \tag{4}$$

Duplicate detection are removed from the merged point set using the same threshold constraints. The full merging steps are described in Algorithm 1.

```
input  : $\{\hat{x}_l^{\mathrm{v}}, \hat{x}_l^{\mathrm{v}}\}_1^L$, $\{x_n^{\mathrm{v}}, x_n^{\mathrm{v}}\}_1^N$, predefined distance threshold $E_{\mathrm{double}}$
output: merged point set $\mathcal{M}$
set $\mathcal{M} = \emptyset$;
compute $\mathcal{E}(l,n)\ \forall n, l$;
while $\min_{l,n}[\mathcal{E}(l,n)] \leq E_{\mathrm{double}}$ do
    $l_{\min}, n_{\min} = \operatorname{argmin}_{l,n}[\mathcal{E}(l,n)]$;
    $x_{n_{\min}}^{\mathrm{v}} \leftarrow \hat{x}_{l_{\min}}^{\mathrm{v}}$;
    $\hat{y}_{l_{\min}}^{\mathrm{v}} \leftarrow y_{n_{\min}}^{\mathrm{v}}$;
    $\mathcal{E}(l,n) \leftarrow \infty$;
end
$\mathcal{M} \leftarrow \{\hat{x}_l^{\mathrm{v}}, \hat{x}_l^{\mathrm{v}}\}_1^L \bigcup \{x_n^{\mathrm{v}}, x_n^{\mathrm{v}}\}_1^N$;
remove duplicate points from $\mathcal{M}$;
```

Algorithm 1: Double-axis diffraction merging procedure.

3 Computing the FOV for Multiple ROIs

As detailed in the main manuscript, our system recovers 2D positions by measuring the intersection of the rainbow streaks with a vertical line sensor (as shown in Fig. 3 of the main manuscript). The field-of-view (FOV) of our system is determined by the angular range, with respect to the diffraction camera, for which such an intersections exist. Here we provide a straightforward analysis for the dependence of the FOV on the configured ROIs.

The system's FOV is computed using Eq. (3) of the main manuscript. As shown in Fig. 2(Right), in our prototype $\theta' \approx 45°$, $\lambda \in [400\mathrm{nm}, 800\mathrm{nm}]$, and $d = 833.3\mathrm{nm}$, which yields a range of possible incident angles:

$$\theta_i \in [-14.6°, 13.1°]. \tag{5}$$

Therefore, the FOV for a single-diffraction-single-ROI system is about 28°. For reference, the FOV of our helper camera, mounted with the 8mm lens, is approximately 48° (see Fig. 2[Left]). In our prototype, two (or more) ROIs expand the FOV up to 52°, which is wider than the corresponding FOV of our 2D helper camera (see Fig. 2[Right]).

4 Additional Experimental Details

Tracker In Fig. 5 of the main paper, we applied a simple tracker to the recovered points yielding motion trajectories. Our tracker uses a combination of a Kalman filter [1] for smoothing and the Hungarian algorithm [3] for keeping track of points between the frames. The current framework works well with such a simple tracking framework due to the high frame-rate of our system.

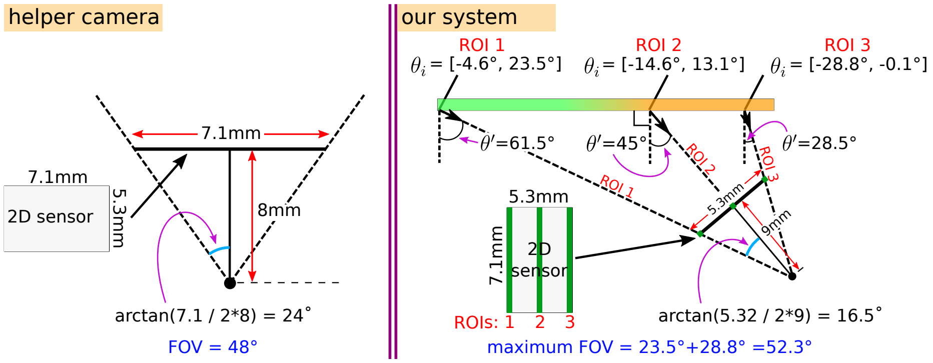

Fig. 2. FOV calculation for multiple ROIs. Both diffraction and helper cameras are equipped with sensors of optical size 7.1mm × 5.3mm. **Left:** The helper camera is equipped with a 8mm lens and thus has a FOV of approximately 48°. **Right:** Our prototype uses multiple ROIs of the 2D sensor. The sensor is rotated by 90° since acquisition speed depends on the number of sensor rows being read out. The camera is positioned at $\theta' = 45°$ with respect to the diffraction grating. Using a single ROI configured at the center of the sensor (marked here as ROI 2) allows detecting light incident in angles $\theta_i \in [-14.6°, 13.1°]$. Thus, the FOV for a single central ROI is about 28°. Defining two additional ROIs at the edges of the sensor (marked here as ROIs 1 and 3) extends the system's FOV to 52.3°. Intuitively, ROIs 1 and 3 'catch' the signal from rainbow streaks whose position on the 2D camera image plane does not intersect ROI 2.

Calibrating The Horizontal Cylindrical Lens System For the 'cylindrical lens' system variation, the virtual image plane's x^{v} axis is set to the cylindrical camera's 1D horizontal image plane $\tilde{x} \equiv x^{\mathrm{v}}$, while the vertical coordinate is set by the diffraction camera $y^{\mathrm{v}} \equiv y$. As described in Section 3 of the main manuscript, calibration proceeds with gathering data pairs $\{x_1^{\mathrm{v,GT}}\} \leftrightarrow \{\mathbf{I}[\Omega(y_1)], y_1\}$ and training the network to approximate H^{-1}. Note that since $y^{\mathrm{v}} \equiv y$, function G in Eq. (4) is now the identity function.

Geometric calibration for 'cylindrical lens' system is done by imaging an LED checkerboard instead of a standard black-and-white checkerboard usually used for 2D camera calibration (see Fig 3) [2]. Similarly to the standard black-and-white checkerboard, the LED checkerboard consists of LEDs arranged at known positions on a plane in 3D space. Geometric calibration consists of imaging the checkerboard's LEDs, thus retrieving their corresponding 2D projections on the virtual image plane. To avoid point position ambiguity (vertical overlap) the checkerboard LEDs are imaged column by column as shown in Fig. 3.

Comparison to Heuristic Classification Fig. 4(Top) here shows a performance comparison between our trained network and a heuristic method for recovering horizontal position using a color patch. We found the network to outperform the heuristic approaches.

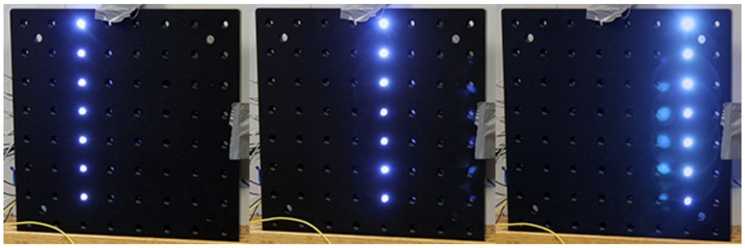

Fig. 3. LED ‘checkerboard’.

Network Training Both our networks (sparse points H^{-1}_{point} and structured light line scanning H^{-1}_{line}) were trained using the ADAM optimizer for 500 epochs and a batch size of 512. When multiple ROIs are used, each ROI (patch or RGB vector) is randomly scaled by a constant. This helps keeping H^{-1}_{point} invariant to point source spectrum and H^{-1}_{line} invariant to object albedo. Training takes between 5 minuets and an hour, depending on the network being trained and the size of the dataset.

3D Reconstruction Details Here we provide additional details about the 3D reconstruction method used in the experiment of Fig. 6 of the main manuscript. Prior to scanning, we perform intrinsic and extrinsic virtual-camera-projector calibration using a standard checkerboard [2]. Now, to extract 3D scene information, we seek the correspondence between the virtual camera pixels $\mathbf{p}^{\text{v}} = (x^{\text{v}}, y^{\text{v}})$ and projector pixels, denoted by $\mathbf{p}^{\text{p}} \equiv (x^{\text{p}}, y^{\text{p}})$. The correspondence is represented by two $A \times W$ matrices $\mathbf{X}^{\text{p}}$ and $\mathbf{Y}^{\text{p}}$, where A and W are the height and width of the virtual image. The value at $\mathbf{X}^{\text{p}}[y^{\text{v}}_0, x^{\text{v}}_0]$ holds the x^{p} coordinate that corresponds to $(x^{\text{v}}_0, y^{\text{v}}_0)$. Similarly, $\mathbf{Y}^{\text{p}}[y^{\text{v}}_0, x^{\text{v}}_0]$ holds the y^{p} coordinate that corresponds to $(x^{\text{v}}_0, y^{\text{v}}_0)$. 3D reconstruction using $\mathbf{X}^{\text{p}}$ and $\mathbf{Y}^{\text{p}}$ follows from triangulation [2].

During the scan, the projector illuminates the object using 440 of its columns, staring at column $x^{\text{p}} = 100$ and ending at $x^{\text{p}} = 540$. Scanning across all projector columns yields a large set of data points $\mathcal{S} \equiv \{x^{\text{v}}_j, y^{\text{v}}_j, x^{\text{p}}_j\}^{J}_{j=1}$, where J is the number of total valid measurement. Note that H^{-1}_{point} yield continuous coordinates $(x^{\text{v}}_j, y^{\text{v}}_j)$. The set $\mathcal{S}$ is then used to linearly interpolate the entries of $\mathbf{X}^{\text{p}}$ (over its regular grid) yielding the correspondence seen in Fig. 6 of the main paper. The resulting correspondence map is then filtered using Bilateral filtering. Matrix $\mathbf{Y}^{\text{p}}$ is then computed from $\mathbf{X}^{\text{p}}$ using epipolar geometry [2].

Light Intensity vs. Accuracy Analysis Our system’s performance depends on the available imaging signal-to-noise ratio (SNR). The SNR depends on the signal intensity that is measured by the camera. We numerically evaluated our neural network’s performance (positioning accuracy) with respect to the signal intensity.

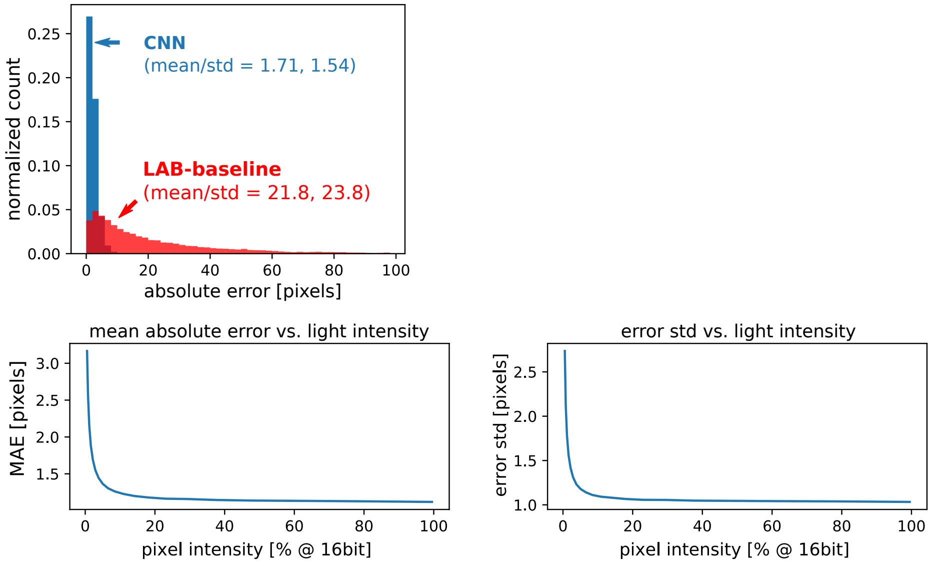

Fig. 4. System numerical evaluation results. **Top:** Histogram showing the absolute classification error in pixels for our proposed network (blue), and a heuristic classification based on color features in LAB color space. Both histograms were created using a dataset with 48K samples. The numbers below each method state the mean and standard deviation. **Bottom:** MAE vs. measured light intensity plot. The intensity is shown in percent of the maximum 16bit graylevel. The plot shows that for intensities as low as 3.3% of the maximum, the CNN performance is high–below 1.75 pixels MAE.

To achieve this we captured a large data-set consisting of approximately 70K diffraction-camera patch samples along with the corresponding ground truth helper-camera 2D positions. We then fed all dataset patches to our CNN. For all the recovered 2D points, we computed the mean-absolute-error (MAE) and standard deviation with respect to the ground-truth points. Using the same dataset, we repeated the procedure above while successively reducing the intensity for all dataset patches. The intensity was reduced while accounting for (adding) Poisson and read-noises to the patches. The added noise parameters were calibrated for our specific prototype camera.

Fig. 4(Bottom) shows the resulting MAE vs. measured light intensity plot. The plot shows an error below 1.75 pixels for intensities above 3.3%. At 0.5% the error is 3.2 pixels and increases as the signal decreases. Overall, the neural networks proves to be very robust to low signals.

Light-source Surface Area vs. Accuracy Analysis The analytical model described in Section 3 of the main manuscript assumes point light sources. Real-world sources however, have a small finite surface area on the projected image plane. This surface area depends on the source (bulb) geometry, the distance to the source, and the source's orientation with respect to the imaging system.

Our network was trained with real sources having such typical surface areas. Moreover, during calibration, we varied the source orientations and position with respect to the camera, yielding samples with a plurality of intensities and visible surface areas. Thus, our dataset had various intensities and surface areas and should be robust in this respect.

Furthermore, we numerically evaluated the robustness of our network for sources having a surface area that is larger than what is nominally found in our dataset. We used the 70K-point dataset described in the previous section to simulate sources with larger horizontal spreads.[1] For example, to simulate a point positioned at $(x^{\mathrm{v}}, y^{\mathrm{v}}) = (300, 500)$ having a dominant intensity which occupies three helper-camera pixels, we feed the CNN with a patch created by averaging three patches belonging to dataset points

$$(x^{\mathrm{v,GT}}, y^{\mathrm{v,GT}}) \in \{(299, 500), (300, 500), (301, 500)\}. \qquad (6)$$

We analyzed our system for sources that dominantly occupy up to 7 helper-camera pixels and saw no noticeable performance drop.[2]

References

1. Grewal, M.S.: Kalman filtering. Springer (2011)
2. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
3. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)

[1] Our system is unaffected by vertical spread since it does not affect the measured color.

[2] Our dataset did not contain enough combinations of adjacent ground-truth points to enable analysis beyond a 7-pixel spread.