Supplementary Material for "Eulerian Single-Photon Vision"

Contents

A. Nearly optimal phase recovery with the discrete Fourier transform

The Fourier coefficient is *not*, in general, the optimal estimator of sinusoid parameters – for that we would need to explicitly maximize the likelihood (effectively the signal reconstruction problem), but that does not have a closed-form solution for the single-photon imaging model. The utility of the DFT lies in its speed through the Fast Fourier Transform (FFT) algorithm. The general idea of side-stepping true maximum-likelihood estimation of sinusoid parameters in favor of approximate solutions appears frequently in the signal processing literature [7, 6, 12].

Notation We consider a 1D sinusoid $f[n]$ imaged by an ideal sensor (quantum efficiency $\eta = 1$ and dark counts $d = 0$) as $B[n]$, with a unit exposure time. We assume that a total of N samples are acquired, and that $f[n]$ is a non-negative single-tone sinusoid:

$$f[n] = c + a\cos\left(\frac{2\pi}{N}k_0 n + \phi\right) \tag{15}$$

where k_0 denotes the signal frequency (assumed integer and less than Nyquist rate), c the constant offset, a the amplitude, and ϕ the initial phase. We denote the discrete Fourier transform by $\mathcal{B}$:

$$\mathcal{B}[k] := \sum_{n=0}^{N-1} B[n]e^{-i\frac{2\pi}{N}kn}. \tag{16}$$

Cramér-Rao bounds We first compute the Cramér-Rao lower bound (CRLB) on the the variance of the sinusoid phase ϕ. We start by defining the log-likelihood function $\mathcal{L}$ given sinusoid parameters $\theta := \{c, k_0, a, \phi\}$ and an observed binary sequence $b[n;\theta]$ arising from that particular parameter setting:

$$\mathcal{L}(\theta';\theta,b) := \sum_n \mathcal{L}_n(\theta';b[n;\theta]), \tag{17}$$

$$\text{where } \mathcal{L}_n(\theta';b[n;\theta]) := \log \Pr\left(B[n;\theta'] = b[n;\theta]\right) \tag{18}$$

The CRLB is calculated using the *Fisher information matrix* (FIM) of the parameters θ, defined as

$$I_{\theta_k\theta_l} := -\mathrm{E}_B\left[\frac{\partial^2\mathcal{L}(\theta';\theta,b)}{\partial\theta'_k\partial\theta'_l}\right]_{\theta'=\theta} \tag{19}$$

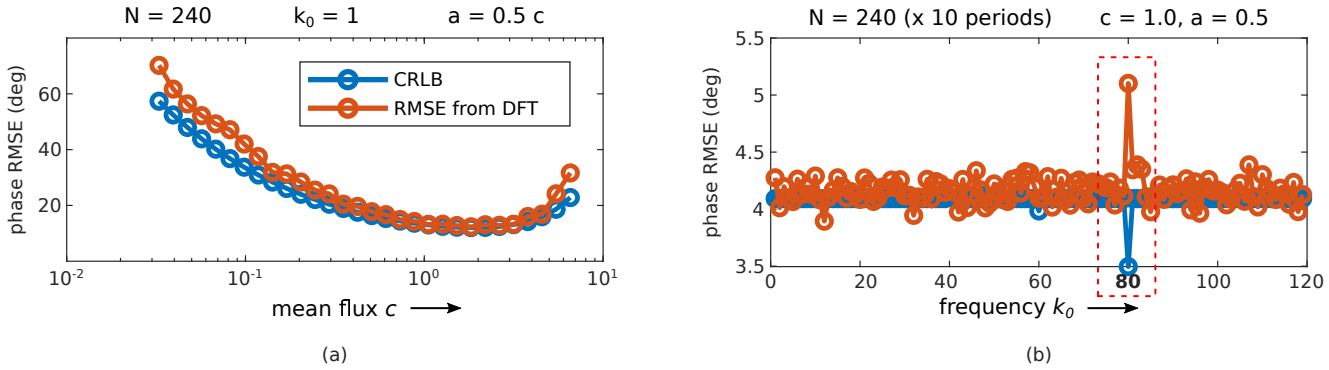

Figure 11: **Discrete Fourier Transform (DFT) phase-recovery as a good approximation to maximum-likelihood estimation**. Binary single-photon samples are simulated from a single-tone sinusoid of eq. (15) with parameters varied systematically. The numerically computed Cramér-Rao lower bound (CRLB) is compared with the root-mean-squared error (RMSE) of the directly computed DFT coefficient phase, averaged over 1,000 trials. See Sec. A for details. (a) Variation with mean flux for a fixed frequency. The error increases again after the flux crosses 1, due to saturation. In all scenarios the DFT is close to the CRLB. (b) Variation with frequency. The sinusoid is repeated $10\times$ to reduce noise, to illustrate clearly the bias at the *critical frequency* of $N/3 = 80$. The error remains close to the CRLB at all other frequencies. A similar pattern is obtained for other sampling rates and flux or contrast levels.

for $\theta_k, \theta_l \in \theta$. The CRLB itself is given as

$$\mathrm{Var}\left[\hat{\theta}\right] \succcurlyeq I^{-1} \tag{20}$$

for any unbiased estimator $\hat{\theta}$; the operator $\succcurlyeq$ denotes that $(\mathrm{Var}\left[\hat{\theta}\right] - I^{-1})$ is positive-semidefinite.

The Fisher information entries are now calculated for an ideal binary quanta sensor. It can be shown that Eq. 19 reduces to

$$I_{\theta_k\theta_l} = \sum_n \left[i(f[n;\theta]) \frac{\partial f[n;\theta]}{\partial \theta_k} \frac{\partial f[n;\theta]}{\partial \theta_l} \right] \tag{21}$$

where $f[n;\theta]$ only makes more explicit the dependence of the signal $f[n]$ on the parameters θ. $i(f[n;\theta])$ is a *pixel-wise Fisher information* defined as follows:

$$i(f[n;\theta]) := -\mathrm{E}_{B[n;\theta]}\left[\frac{\mathrm{d}^2\mathcal{L}_n}{\mathrm{d}f[n;\theta]^2}\right] = \frac{1}{e^{f[n;\theta]} - 1} \tag{22}$$

Corresponding expressions can be derived for other sensor types such as those with non-zero read noise or higher full-well capacities, but are not considered here. From Eq. 15, the partial derivative $\partial f[n;\theta]/\partial\phi$ for the case of phase is

$$\frac{\partial f[n;\theta]}{\partial \phi} = -a \sin\left(\frac{2\pi}{N} k_0 n + \phi\right) \tag{23}$$

which can be used in Eq. 21 to calculate the Fisher information entry $I_{\phi\phi}$ (similar for other parameters). As the overall expression does not have a closed-form, we calculate it numerically, inverting the FIM thus obtained to finally arrive at the CRLB.

Simulation experiment The CRLB is compared to the expected error from directly reading out DFT phase. We generate single-tone sinusoids with $N = 240$ samples, systematically varying the mean flux level c and the signal frequency k_0 (going up to the Nyquist rate of $N/2 = 120$). The phase ϕ is randomly set initially. 1,000 trials are performed for each configuration and the mean squared error of phase (relative to the initial value) is calculated. Fig. 11 shows the results.

The error with the DFT phase is close to the CRLB in almost all conditions except when the signal frequency is exactly

$N/3 = 80$, which is when induced harmonics from the sensor's non-linearity (discussed in the main paper in Appendix A) alias onto the fundamental frequency, resulting in bias [7].

The results are affected systematically by the value of the initial phase ϕ (similar to classical settings [10]), but the impact is relatively minor – it changes the amount of bias at the critical frequencies which have been just identified, and the CRLB on phase variances at other frequencies. Importantly, all non-critical frequencies remain unbiased in single-tone signals for any value of ϕ.

B. Velocity-Tuned Filter Design

We use space-time separable filters; that is, $h_{\mathbf{k}} := h^{\text{spatial}}_{(k_r,k_\theta)} * h^{\text{temporal}}_{k_t}$ where $\mathbf{k} := (k_x, k_y, k_t)$ represents the filter's tuned 3D frequency. The spatial filters are log-Gabor [4, 8], designed in the frequency domain to be *polar-separable*. They largely follow the description given in the following link: https://peterkovesi.com/matlabfns/PhaseCongruency/Docs/convexpl.html, but the details are given below for completeness.

If $(k_r, k_\theta) := \left(\sqrt{k_x^2 + k_y^2}, \texttt{atan2}(k_y, k_x)\right)$ represent the polar coordinates for (k_x, k_y), the transfer function is

$$H^{\text{spatial}}_{(k_r,k_\theta)}(k'_r, k'_\theta) := H^{\text{radial}}_{k_r}(k'_r) \cdot H^{\text{ang.}}_{k_\theta}(k'_\theta), \text{ where} \tag{24}$$

$$H^{\text{radial}}_{k_r}(k'_r) := \exp\left(-\frac{\left(\log {}^{k'_r}/_{k_r}\right)^2}{2\left(\log(\sigma_0)\right)^2}\right), \text{ and} \tag{25}$$

$$H^{\text{ang.}}_{k_\theta}(k'_\theta) := \cos^2\left(\frac{\pi}{2} \cdot \min\left(1, \frac{\texttt{angdiff}(k'_\theta, k_\theta)}{\Delta\theta_0}\right)\right) \tag{26}$$

σ_0 and $\Delta\theta_0$ are constants, set to 0.55 and the spacing between the tuned orientations (60° for six orientations, 90° for four, etc.) respectively. This setting for σ_0 ensures a radial bandwidth of approximately two octaves, and that for $\Delta\theta_0$ ensures that the angular weight decays to zero exactly at the next tuning orientation. We typically create these filters at three scales, meaning $k_r \in \{f_0, {}^{f_0}/_{\mu}, {}^{f_0}/_{\mu^2}\}$ for an $f_0 := {}^1/_{\lambda_{\min}}$ which is set depending on the scene content and the expected SNR level. In general $\lambda_{\min} \in [3, 6]$ pixels, and μ is set to 2.1.

The temporal filters are designed separately for each scale, with the center frequencies k_t obtained for a pre-specified set of velocities $\{0, v_1, v_2, \ldots\}$ through the velocity-tuning relation of Fig. 3 in the main paper (repeated below):

$$k_t = -v_{\text{normal}}\sqrt{k_x^2 + k_y^2} \qquad \text{(velocity-tuning)} \tag{27}$$

The transfer function is again log-Gabor with transfer function like Eq. 25:

$$H^{\text{temporal}}_{k_t}(k'_t) := \exp\left(-\frac{\left(\log {}^{k'_t}/_{k_t}\right)^2}{2\left(\log(\sigma^t_0)\right)^2}\right) \tag{28}$$

We set $\sigma^t_0 = 0.55$ here as well, and generally tune the filters to three distinct velocities $\{0, v_1, v_2\}$ (including zero), with v_1 and v_2 set depending on the scene.

C. Motion Estimation from Phase-Constancy Equations

Sec. 5.2 described how the phase-constancy relation yields an equation for each filter (Eq. 7). Fleet and Jepson [5] provide a method to compute the phase gradient (or local frequency) directly from the responses $R_{\mathbf{k}}$ without needing unwrapping. More relevant to the current situation, they also show that the local frequency term on the left-hand side is expected to be close to the filter tuning $\mathbf{k}$: this is because the response $R_{\mathbf{k}}$ turns out to be essentially a modulation of a base or carrier signal, the sinusoid at the frequency $\mathbf{k}$. They use this fact as one of their two criteria to determine if an equation from a filter is reliable (the other is based on the magnitude of the response being over a threshold). They then solve an *unweighted* least-squares problem from the selected reliable responses in a $5 \times 5 \times 5$ neighborhood. Our algorithm is similar to theirs, but makes the following changes:

- first, we only use the responses at the pixel itself (no neighborhood);
- and second, we replace the local frequency estimate with the filter tuning $\mathbf{k}$ – this is a coarse approximation which removes the need to compute phase gradients entirely. Both enable a faster implementation.
- Finally, their unweighted least-squares formulation is equivalent to setting a 0–1 weight on the equations in a weighted one. In our work we define continuous-valued weights based on the responses' z-scores using the function of Fig. 3 from the main paper. This allows more responses to contribute (needed since we do not use neighboring pixel responses), without adding much computation to the solver.

D. Edge Detector Evaluation with Simulated Single-Photon Data

We compare the Temporal Phase Congruency algorithm (TPC) described in the main paper (Sec. 5.1), which works on the whole video sequence at once, to frame-by-frame processing approaches applied both with and without single-image denoising, and after simply averaging all the frames. We find that for extremely low flux levels ($\leq$ 1 photon/pixel), our method is significantly better than processing a single frame, even after reducing noise by single-image denoising or by averaging all the frames. Frame-by-frame methods take the lead for flux levels $\geq$ 3 photons/pixel, as the frame quality improves sufficiently for the machine learning-based baseline detector to start detecting edges.

Dataset We simulate single-photon sensing under different light level (exposure) settings, in a similar way to the experiment described in Fig. 7 from the main paper, with the moving synthetic circle. The images for the simulation are taken from the XVFI dataset [11] which contains high-speed clips of 33 frames each, at 4K resolution. We used the "Test" split which contains 15 clips in total. Three of these (#13 – #15) are excluded because they consist of heavily textured scenes containing foliage and/or water. It is difficult to extract meaningful edges from them, even with very high-quality images.

The images are down-sampled by a factor of $1/8$ to make the motion closer to the 1 pixel/frame regime we expect quanta samples to be acquired in. The simulation is performed under the following settings (4 different precision levels $\times$ 4 flux settings for each case $=$ 16 imaging regimes):

- 9-bit sampling, at flux levels $\in \{20, 80, 300, 800\}$ photons per pixel (ppp);
- 6-bit sampling, flux levels $\in \{7, 20, 60, 100\}$ ppp;
- 3-bit sampling, flux levels $\in \{1, 3, 7, 10\}$ ppp; and
- 1-bit sampling, flux levels $\in \{0.1, 0.5, 1.25, 2\}$ ppp.

Each simulated image is denoised with BM3D [2]. The frame-by-frame edge detectors are run on both the raw and the denoised images separately, and on the full average of all the frames.

Reference edge maps for evaluation are generated using the Richer Convolutional Features (RCF [9]) detector, since human-annotated edges are not available for this dataset.

Methods tested Apart from the Temporal Phase Congruency (TPC) algorithm described in Sec. 5.1 of the main paper, RCF is applied on the single frames as a baseline method. Since the reference edge map is also generated using RCF, the evaluation is likely biased towards this algorithm, and therefore we also employ the Structured Edges (SE [3]) detector as an independent high-quality frame-based method which works in similar settings. Thus there are seven sets of results:

- Temporal Phase Congruency applied to the entire video sequence (TPC);
- RCF directly applied to single-photon frames (*RCF-direct*);
- RCF applied after single-image denoising with BM3D (*RCF-denoised*);
- RCF applied after averaging all the frames (*RCF-avgall*);
- SE applied directly (*SE-direct*);
- SE applied after denoising (*SE-denoised*); and
- SE applied after averaging all the frames (*SE-avgall*).

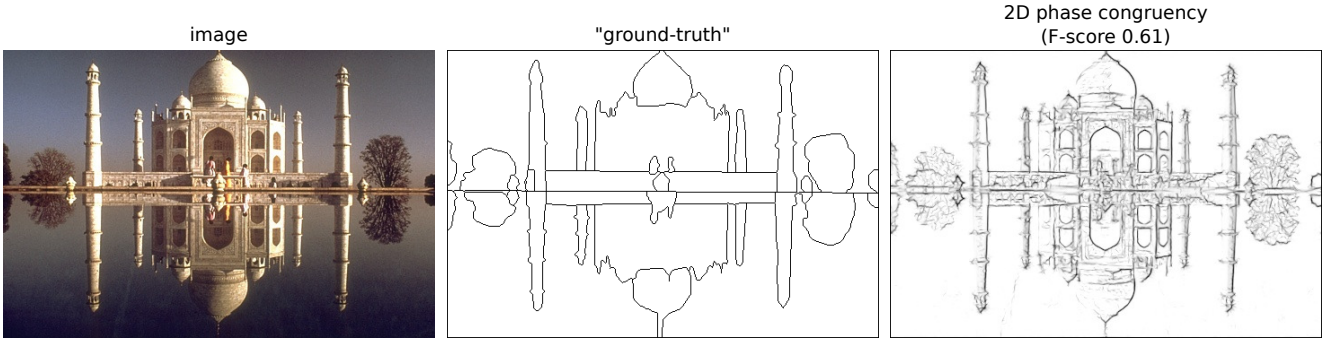

Figure 12: **Limitations of quantitative evaluation of edge detectors**. Left, middle: example image from BSDS dataset [1], and human-annotated "ground-truth" edge map. Right: edge map from a 2D phase congruency-based edge detector [8], which takes a single image as input unlike TPC. While fine details are recovered by the phase congruency algorithm, the over-simplified edges in the so-called ground-truth result in a relatively poor score.

Evaluation metric Edge detectors are typically evaluated on their *precision* and *recall* relative to a reference ("ground-truth") edge map, which are combined in the F-measure. In general, a detector outputs a continuous-valued edge score which is thresholded to obtain the edge map, and the precision and recall are then calculated. In our case, we use the script accompanying the BSDS boundary detection dataset [1], which searches for an *optimal dataset scale* (ODS) threshold; this is a single common threshold value for all images over the dataset that yields the best performance. The F-score at this threshold is termed the *ODS score*, and is reported as the final measure of performance.

Results The scores of each tested detector under all simulation settings are plotted in Fig. 13 (see next page). With higher-quality images (6-bit and 9-bit), the frame-based approaches typically perform well, scoring higher than TPC. But at lower flux settings (≤ 1 photon/pixel with 1-bit and 3-bit samples), TPC outperforms frame-based methods by a large margin.

Fig. 14 studies the variation in performance of each detector as the light level changes. Considering the frame-based methods RCF and SE applied directly to the frames, we see that detector performance ordinarily degrades smoothly as the light reduces, though the rate of fall is quite steep. Single-image denoising arrests this decline significantly, but only down to flux levels of ~ 3 photons/pixel. Improvements are also marginal for very high flux levels (≥ 300 photons/pixel or so). Simply averaging the frames yields modest improvements in performance over both direct processing and single-image denoising, but only at extremely low flux levels (~ 0.1 photons/pixel). It suffers from motion blur in general, which hampers performance enough to generally outweigh the benefit of noise reduction.

In contrast to frame-based methods, the rate of degradation of TPC's performance is much more graceful, and its performance is nearly its optimal level even with flux ~ 1 photon/pixel.

Limitations A relevant historical detail here is that the BSDS dataset (which both RCF and SE are trained on) was designed for segmentation, and is thus biased to contain only the most salient edges corresponding to object boundaries (RCF inherits this bias through training). See Fig. 12 for an example. A method not based on learning (like TPC) typically yields denser edge maps and therefore scores relatively poorly even with clean images. This bias needs to be kept in mind when evaluating an algorithm. Extending TPC through learning-based methods may help score better on this type of evaluation.

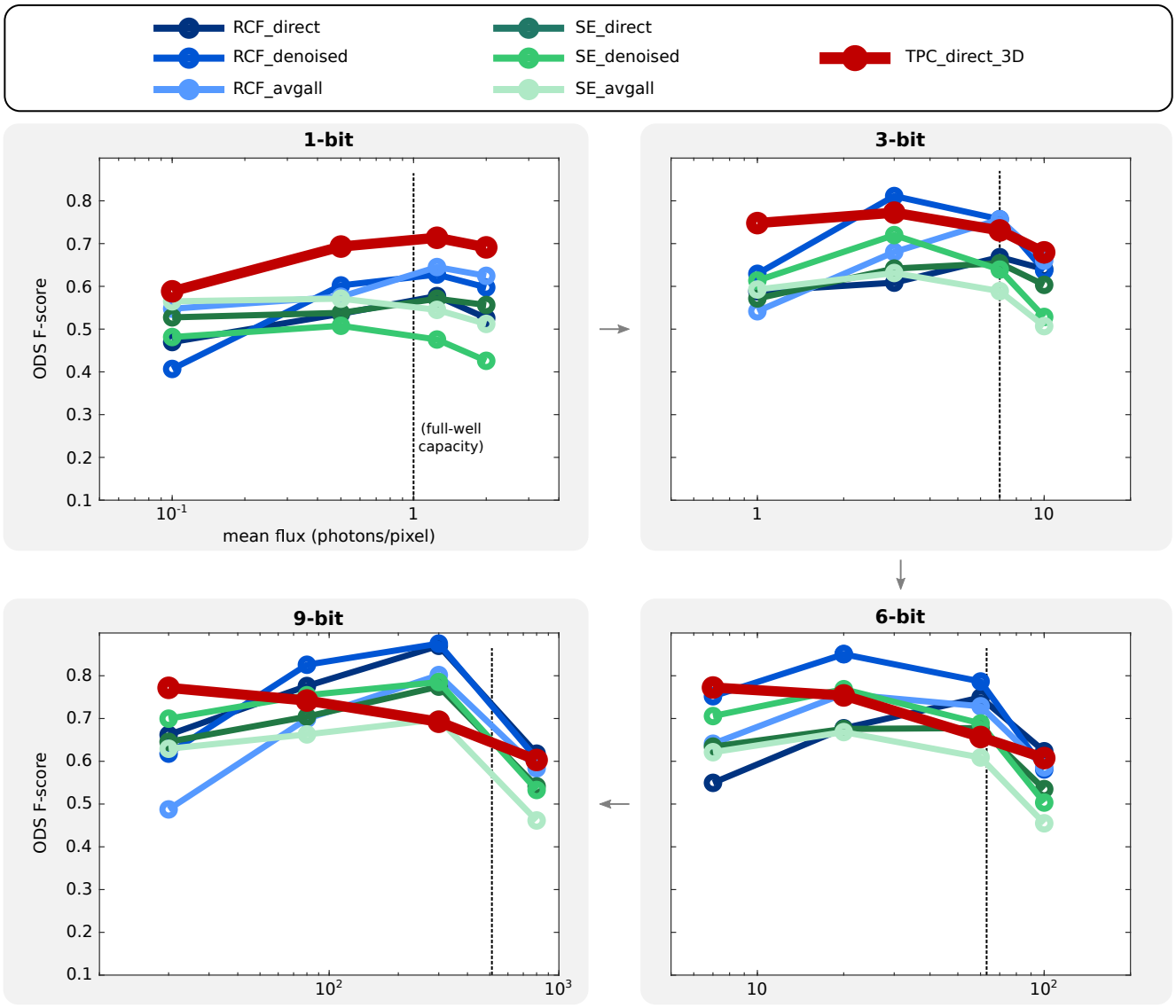

Figure 13: **XVFI simulation results: comparing edge detectors at a fixed setting**. Edge detectors are evaluated on single-photon data simulated from the XVFI dataset [11] — a higher score is better. See Sec. D for details. Each plot shows performance with samples of the specified precision. With more light and high-precision data (light levels $\geq$ 3 photons/pixel and $\geq$ 3-bit samples), applying RCF or SE after single-image denoising generally performs best. The video-based Temporal Phase Congruency detector (TPC_direct_3D) outperforms all frame-based methods by a large margin at the 1-bit level with flux $\sim$ 1 photon/pixel. Simply averaging all the frames is relatively effective at extremely low flux levels ($\sim$ 0.1 photons/pixel) as all techniques struggle in these settings, but generally performs poorer than single-image denoising or even direct processing.

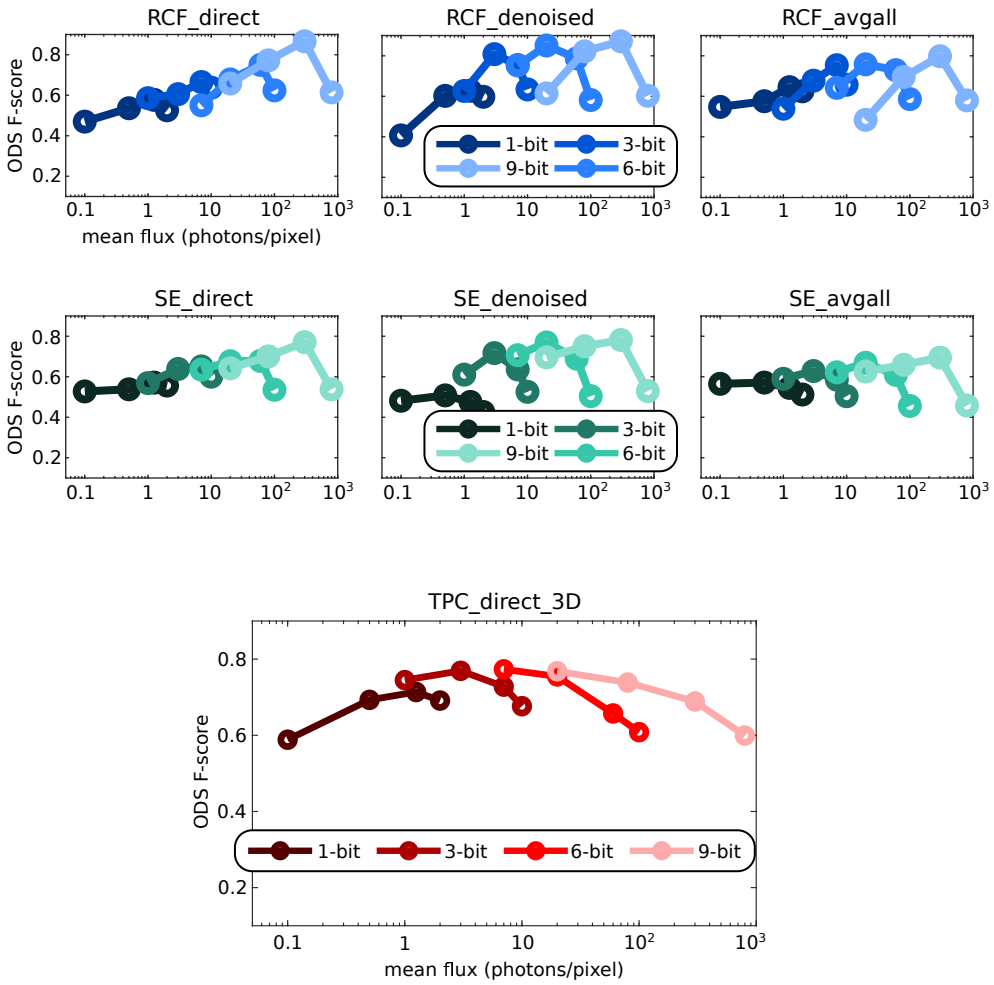

Figure 14: **XVFI simulation results: edge detector performance variation across imaging conditions**. For each detector setting, performance is plotted versus light level for each sample precision setting (see Sec. D for details on the setup). For direct detection with both RCF and SE, performance degrades smoothly but steeply as the light reduces. Large gains are seen from single-image denoising for flux levels > 1 photon/pixel, but not below that. Averaging all the frames yields modest improvements in extremely low flux (~ 0.1 photon/pixel) but quickly saturates due to losing edges to motion blur. Bottom: Performance of Temporal Phase Congruency (TPC) under the same conditions. While performance does degrade at low light levels similar to the frame-based detectors, the drop-off is much smaller.

References

[1] P Arbeláez, M Maire, C Fowlkes, and J Malik. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, May 2011. 5

[2] Lucio Azzari and Alessandro Foi. Variance Stabilization for Noisy+Estimate Combination in Iterative Poisson Denoising. *IEEE Signal Processing Letters*, 23(8):1086–1090, Aug. 2016. 4

[3] Piotr Dollar and C. Lawrence Zitnick. Fast Edge Detection Using Structured Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570, Aug. 2015. 4

[4] David J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379, Dec. 1987. 3

[5] David J. Fleet and Allan D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, Aug. 1990. 3

[6] Christopher Gianelli, Luzhou Xu, Jian Li, and Petre Stoica. One-Bit compressive sampling with time-varying thresholds for multiple sinusoids. In *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Curacao, Dec. 2017. IEEE. 1

[7] A. Host-Madsen and P. Handel. Effects of sampling and quantization on single-tone frequency estimation. *IEEE Transactions on Signal Processing*, 48(3):650–662, Mar. 2000. 1, 3

[8] Peter Kovesi. Image Features from Phase Congruency. *Videre: Journal of Computer Vision Research*, 1(3), 1999. 3, 5

[9] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer Convolutional Features for Edge Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3000–3009, 2017. 4

[10] D. Rife and R. Boorstyn. Single tone parameter estimation from discrete-time observations. *IEEE Transactions on Information Theory*, 20(5):591–598, Sept. 1974. 3

[11] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. XVFI: eXtreme Video Frame Interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14489–14498, 2021. 4, 6

[12] S. Tretter. Estimating the frequency of a noisy sinusoid by linear regression (Corresp.). *IEEE Transactions on Information Theory*, 31(6):832–835, Nov. 1985. 1