1. Overview of Camera Noise Model

In this section, we present an overview of different camera noise sources followed by a justification of the Poisson noise model used for photon-limited settings. Consider the sensor output of $i^{th}$ pixel of camera, denoted as $Y_i$. From [10], we model $Y_i$ as the following random variable:

$$Y_i \sim K_d \left( K_a (P(I_i) + \eta_a) + \eta_d + \eta_q \right) \quad (1)$$

Here $I_i$ represents the average number of incident photons during the exposure. $K_a$, $K_d$ represent the analog and digital gain respectively. $\eta_a$ represents noise sources before the analog gain (dark current shot noise, flicker noise etc.) and $\eta_d$ represents noise sources before the digital gain (thermal noise, fixed pattern noise). $\eta_q$ represents the quantization noise. From Eq. (1), we can see $Y_i$ is a noisy measurement of the parameter $I_i$.

Eq. (1) can be simplified in the following way:

$$\tilde{Y}_i \sim \mathcal{P}(I_i) + \eta_a + \frac{1}{K_a} (\eta_d + \eta_q) \quad (2)$$

where $\tilde{Y}_i \triangleq Y_i / (K_a K_d)$ is the normalized pixel measurement. The noise sources can be easily decoupled into signal dependent Poisson noise and signal independent noise sources. The latter is often approximated as zero mean Gaussian noise in literature [2,3,6] leading to the following Poisson-Gaussian mixture modelling

$$\tilde{Y}_i \sim \mathcal{P}(I_i) + Z_i \quad Z_i \sim N(0, \sigma^2) \quad (3)$$

We can further break down average number of incident photons $I_i$ as $I_i \triangleq \alpha x_i$ where $\alpha$ is a function of camera parameters such as exposure time, quantum efficiency, sensor area etc. and $x_i$ is the scene radiance corresponding to the $i^{th}$ pixel. This helps us decouple the scene and camera characteristics in the signal $I_i$. Therefore, we arrive at the following camera model formulation where $Y_i$ is the measurement of the true signal $x_i$:

$$\tilde{Y}_i \sim \mathcal{P}(\alpha x_i) + N(0, \sigma^2) \quad (4)$$

1.1. Poisson Noise SNR

For this subsection, we ignore the Gaussian term in Eq. (4) i.e. $\sigma = 0$ to focus on the nature of Poisson noise. An interesting property of the Poisson random variable is that its mean and variance are the same. Thus, the signal-to-noise ratio for $\tilde{Y}_i$, in absence of Gaussian noise, is given as follows:

$$\text{SNR}(\tilde{Y}_i) \triangleq \frac{\mathbb{E}[\tilde{Y}_i]}{\sqrt{\text{Var}[\tilde{Y}_i]}} = \sqrt{\alpha x_i} \quad (5)$$

This implies that our measurements get noisier with decreasing number of incident photons. If the scene is not well-illuminated (low $x_i$) or the exposure is short (small $\alpha$), the number of incident photons on the sensor is also low leading to noisier images.

1.2. Photon-Limited Scenes

Scenes where the Poisson noise dominates other sources of noise in the measurements are defined as photon-limited [4]. For the random variable $\tilde{Y}_i$ defined in (4), this occurs when the variance due to Poisson noise is greater than the variance due to Gaussian noise, i.e. $\alpha x_i \geq \sigma^2$.

However, for the purpose of this paper, not all photon-limited scenes are equally significant. To emphasize this point, we inspect the SNR for Poisson-Gaussian mixture $\tilde{Y}_i$ for different levels of $\alpha$. The SNR for $\tilde{Y}_i$ in presence of both Poisson and Gaussian noise is given as follows:

$$\text{SNR}(\tilde{Y}_i) \triangleq \frac{\mathbb{E}[\tilde{Y}_i]}{\sqrt{\text{Var}[\tilde{Y}_i]}} = \frac{\alpha x_i}{\sqrt{\alpha x_i + \sigma^2}} \quad (6)$$

Consider the case of read noise $\sigma = 1.6e^{-7}$. We assume $x_i = 1$ and inspect the random variable $\tilde{Y}_i$ for different $\alpha$ in the photon-limited regime, as shown in Table 1.

From the table we can conclude the following: Images taken in well-illuminated scenes with good exposure can be approximated as noiseless for the purpose of deblurring. However, on the other end of photon-limited regime...
### Table 1. Signal-to-noise ratio (SNR) for different photon levels in photon-limited regime.

<table>
<thead>
<tr>
<th>Photon level (\alpha)</th>
<th>SNR (in dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha = 1000)</td>
<td>29.98 dB</td>
</tr>
<tr>
<td>(\alpha = 40)</td>
<td>15.75 dB</td>
</tr>
<tr>
<td>(\alpha = 20)</td>
<td>12.48 dB</td>
</tr>
<tr>
<td>(\alpha = \sigma^2)</td>
<td>1.07 dB</td>
</tr>
</tbody>
</table>

\(\alpha = \sigma^2\), there is too much noise in the image for any meaningful recovery from a single frame. Therefore, for this paper, we explore photon-limited deconvolution for \(\alpha \in [10, 40]\) where shot noise dominates read noise but there is still a possibility of extracting the clean image.

## 2. Initialization Algorithm

In Algorithm 1, we describe the kernel initialization method for the proposed method. This scheme is a minor variation of the kernel estimation method from [1] and used to estimate a rectilinear kernel with parameters \(\{\rho, \theta\}\) from the blur-only image \(G(y)\). We would like to reiterate to the reader that the scheme in Algorithm 1 is not the kernel estimation process in its entirety and only represents the initialization process. The kernel estimated at the end of this algorithm is further refined in Stage I and II of the main iterative scheme.

**Algorithm 1 Initialization for Kernel Estimation**

**Steps:**
1. **Input:** Blur-only Image \(G(y)\), Number of Key Points \(K\)
2. Estimate gradient images \(D_x, D_y\) from \(G(y)\)
3. for \(\theta = 1, 2, \ldots, 180\) do
   4. \(D_\theta \leftarrow D_x \cos(\theta) - D_y \sin(\theta)\)
   5. \(f_\theta \leftarrow \max(|D_\theta|)\)
4. end for
5. \(f_\theta, \hat{\theta} \leftarrow \min f_\theta\)
6. \(\rho \leftarrow \sqrt{\frac{C_1}{K} - \sigma_b^2}\)
7. \((x_0, y_0) \leftarrow (0, 0)\)
8. for \(k = 1, 2, \ldots, K - 1\) do
   9. \((x_k, y_k) \leftarrow \frac{k\theta}{K\pi} \cos(\hat{\theta}), \frac{k\theta}{K\pi} \sin(\hat{\theta})\)
10. end for
11. \(z_0 \leftarrow ([x_0, y_0, x_1, y_1, \ldots, x_{K-1}, y_{K-1}]^T)\)
12. return \(z_0\)

## 3. Qualitative Comparison

Extended qualitative results comparing end-to-end trained methods to our approach are provided in the supplementary document. Figure 1 and 2 provide qualitative examples from the RealBlur dataset which contains realistic blur. Figure 3 contains examples from the PLDD dataset [9] contains real-shot noise corrupted and blurred image sensor data along with the ground-truth kernel, as measured using a point source. Finally, Figure 4 provides reconstruction results on synthetically blurred images using motion kernels from Levin dataset [5].

## 4. KTN Architecture and Training

The KTN architecture can be summarized as follows: the vectorized control points of dimension \(2 \times (K - 1)\) are passed through 3 fully-connected layers followed by reshaping into an image. The reshaped image is then passed through the decoding half of a UNet to give the kernel output. The final output, when used in the iterative scheme, is clipped to zero and then normalized to one. Architecture details of KTN are provide in Figure 5.

## 5. Implementation Details

**Boundary Conditions:** While blurring the image synthetically, the boundary conditions are important to take into account. Circular boundary conditions allow the blur operator to be written in terms of FFTs and making it computationally inexpensive, it is not a realistic assumption for natural blur. A more appropriate boundary condition to assume is symmetric boundary condition.

This has major implications for our inverse problem scheme. Since PhD-Net assumes circular boundary conditions, we need to pad the image symmetrically, pass through PhD-Net and crop out the relevant portion to deblur the image without any artifacts. Second, when calculating the re-blurring loss, \(h_x \odot F(y, h_x)\) needs to be calculated using symmetric boundary conditions.

**Step Size and Backtracking:** For Stage I, we set the initial step size as \(\delta = 10^2\) and for Stage II, we set \(\delta = 2.0\). For every iteration, we check whether the current choice of step-size decreases the cost-function or not. If it doesn’t, then the step-size is reduced by half for rest of the iterative scheme until the next time the cost function increases instead of decreasing. Note that \(\delta\) is set very large in the first stage compared to the second. This is because the gradients are backpropagated through two networks i.e., \(F(\cdot)\) and \(T(\cdot)\) instead of one, leading to the vanishing gradient problem and hence justifying the larger step size.

**Computational Time:** For each stage of the iterative scheme, we limit the number of iterations to 150. The experiments in the main document are performed on a Nvidia TitanX GPU, and take approximately 0.35 seconds per iteration.

## References


Figure 1. Qualitative examples on the Real-Blur Dataset
Figure 2. More Qualitative examples on the Real-Blur Dataset
Figure 3. More comparisons on Photon-Limited Deblurring Dataset [8].
Figure 4. **Qualitative Examples on Synthetic Blur**: "Non-Blind" is provided for reference and serves as an upper bound on the deconvolution performance. It is obtained through PhD-Net with noisy-blurred image and ground truth kernel as inputs. The kernels in inset of "Ours" and "Non-Blind" represent the estimated and true blur kernel respectively.
Figure 5. **Kernel Trajectory Network Architecture** 3 fully connected layers followed by a U-Net decoder