Supplementary Document for
“ Motion Adaptive Deblurring with Single-Photon Cameras ”

Trevor Seets, Atul Ingle, Martin Laurenzis, Andreas Velten
Correspondence to: seets@wisc.edu

S.1. Flux Changepoint Detection

S.1.1. Offline Algorithm: Cost Function Derivation

Consider a set of photon time stamp measurements $\{x_i\}_{i=1}^N$. Here each x_i is a valid measurement, and in the frame-readout capture mode, described in the main text, this is different than the t_i's. If no photon is detected in a frame we add the frame length to the next detected photon. We do this so each x_i will be i.i.d. and distributed exponentially. We again wish to find a set of change points, $\{x_{l_1}, \dots, x_{l_L}\}$. In general, the optimization problem for changepoint detection is given by Eq. (P2) in [29]:

$$(l_i^*)_{i=1}^L = \underset{l_1,..,l_L}{\arg\min} \sum_{i=1}^{L-1} c(x_{l_i \dots l_{i+1}}). + \lambda L \tag{S1}$$

The summation term represents the likelihood that each segment in between changepoints come from the same underlying distribution, while the regularization term is needed because the number of changepoints are not known *a priori*. For our case $c(\cdot)$ is the negative log likelihood for a set of exponentially distributed measurements. Let $f(x)$ be the exponential density function with rate parameter Φ, and let $\widehat{\Phi}_i$ be the maximum likelihood estimate for Φ for the set of measurements $\{x_{l_i} \dots x_{l_{i+1}}\}$. Note that the maximum likelihood estimator maximizes the log likelihood. To derive $c(\cdot)$, we begin with Eq. (C1) from [29]:

$$\begin{aligned}
c(x_{l_i \dots l_{i+1}}) &= -\max_{\Phi} \sum_{j=l_i}^{l_{i+1}} \log f(x_j|\Phi) && \text{(S2)}\\
&= -\sum_{j=l_i}^{l_{i+1}} \log f(x_j|\widehat{\Phi}_i) && \text{(S3)}\\
&= -\sum_{j=l_i}^{l_{i+1}} \log \widehat{\Phi}_i e^{-\widehat{\Phi}_i x_j} && \text{(S4)}\\
&= -\sum_{j=l_i}^{l_{i+1}} \log \widehat{\Phi}_i + \sum_{j=l_i}^{l_{i+1}} \widehat{\Phi}_i x_j && \text{(S5)}\\
&= -(l_{i+1} - l_i) \log \widehat{\Phi}_i + (l_{i+1} - l_i) && \text{(S6)}
\end{aligned}$$

where the last line comes from the fact that $\widehat{\Phi}_i = \frac{(l_{i+1} - l_i)}{\sum_{j=l_i}^{l_{i+1}} x_j}$. Plugging Eq. (S6) into Eq. (S1), the last term sums to a constant N and can be dropped from the optimization. Then we convert to the direct measurments t_i by expanding out where no photons where found to get Eq. (2).

S.1.2. QIS: Offline Cost Function

A quanta image sensor (QIS) is another sensor type capable of measuring single photons. Unlike a SPAD, the QIS senor only gives a binary output for each photon-frame corresponding to whether or not a photon was detected. Note that we can convert our experimental SPAD data to QIS data by stripping the SPAD data of the timing information. Let $n_i = 0$ if the i^{th} QIS photon-frame detects no photons and $n_i = 1$ otherwise. Let τ_b be the temporal bin width for each photon-frame. Suppose the jot is exposed to a flux of Φ, then the probability of detecting a photon during photon-frame i is:

$$p = P(n_i = 1) = 1 - e^{-q\Phi\tau_b} \tag{S7}$$

Where q is the quantum efficiency. We can model measuring multiple photon-frames with the QIS jot as a Bernoulli Trial, with probability of success given by Eq. S7. For a set of N photon-frames the maximum likelihood estimator, $\hat{\Phi}_{QIS}$, is given by [17],

$$\hat{\Phi}_{QIS} = \frac{-1}{q\tau_b} \ln(1 - \widehat{p}) \tag{S8}$$

$$\widehat{p} = \frac{\sum_{i=0}^{N} n_i}{N} \tag{S9}$$

This flux estimator should also be used in SPAD sensors under very high fluxes, where their is significant probability of detecting more than one photon in a period equal to the SPAD's time quantization. Similarly, the MLE in Eq. 1 can be used in low light conditions for a QIS sensor.

We derive the changepoint cost function in the raw data domain. Following the steps of the earlier derivation, with $f(n_i)$ being the Bernoulli distribution with parameter p:

$$c_{QIS}(x_{l_i \ldots l_{i+1}}) = -\max_{\Phi} \sum_{j=l_i}^{l_{i+1}} \log f(n_j | \Phi) \tag{S10}$$

$$= -\sum_{j=l_i}^{l_{i+1}} \log f(n_j | \widehat{\Phi}_i) \tag{S11}$$

$$= -\log(\widehat{p_i}) \sum_{j=l_i}^{l_{i+1}} n_j - \log(1 - \widehat{p_i}) \sum_{j=l_i}^{l_{i+1}} 1 - n_j \tag{S12}$$

Where $\widehat{p_i} = \frac{\sum_{j=l_i}^{l_{i+1}} n_j}{l_{i+1} - l_i}$.

S.1.3. Online Flux Changepoint Detection Algorithm

Offline changepoint detection is suitable for offline applications that capture a batch of photon frames and generate a deblurred image in post-processing. In some applications that require fast real-time feedback (e.g. live deblurred video) or on-chip processing with limited frame buffer memory, online changepoint detection methods can be used. We use a Bayesian online changepoint detection method [31]. This algorithm calculates the joint probability distribution of the time since the last flux changepoint. For exponentially distributed data, it uses the posterior predictive distribution which is a Lomax distribution (see Suppl. Note S.1.3). We assume that the flux changepoints appear uniformly randomly in the exposure window T. Because detecting a flux changepoint after only one photon is difficult we use a look-behind window that evaluates the probability of the photon 20–40 photon frames into the past as being a flux changepoint. Using a look-behind window greatly increases detection accuracy and introduces only minor latency (on the order of tens of microseconds). We also found that it is helpful to use a small spatial window that spreads out flux changepoints in space to increase the density of changepoints. In general, online detection will work better for slower motion as the algorithm learns from past data. We compare online and offline detection in Suppl. Note S.3.

We use a Bayesian changepoint detection algorithm shown in Algorithm 1 of [31]. Here we derive the posterior predictive distribution used in Step 3 of their algorithm. We use a $\mathsf{Gamma}(\alpha, \beta)$ prior for Φ. Let $\mathbf{x} := \{x_i\}_{i=1}^{N}$. It can be shown that $\Phi|\mathbf{x} \sim \mathsf{Gamma}(\alpha + N, \beta + \sum_{i=1}^{N} x_i)$. The predictive posterior density is given by:

$$f_{x_{N+1}|\mathbf{x}}(y|\mathbf{x}) = \int_0^\infty f_{x_{N+1}|\mathbf{\Phi}}(y|\Phi) f_{\mathbf{\Phi}|\mathbf{x}}(\Phi|\mathbf{x}) d\Phi \tag{S13}$$

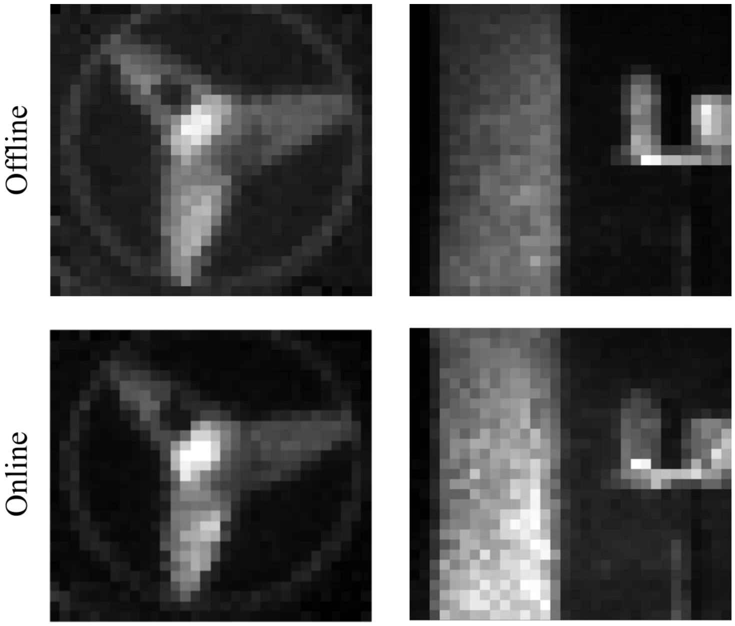
$$= \int_0^\infty \Phi e^{-\Phi y} \frac{(\beta + \sum x_i)^{\alpha+N}}{\Gamma(\alpha + N)} \Phi^{\alpha+N-1} e^{-(\beta + \sum_i x_i)\Phi} d\Phi \tag{S14}$$

$$= \frac{(\beta + \sum_i x_i)^{\alpha+N} (\alpha + N)}{(\beta + y + \sum_i x_i)^{\alpha+N-1}} \tag{S15}$$

which is a Lomax density with shape parameter $\alpha + N$ and scale parameter $\beta + \sum_i x_i$. For our data we used a $\mathsf{Lomax}(1, 100)$ in Step 3 and $H(\cdot) = 40$ in Steps 4 and 5 of Algorithm 1 in [31].

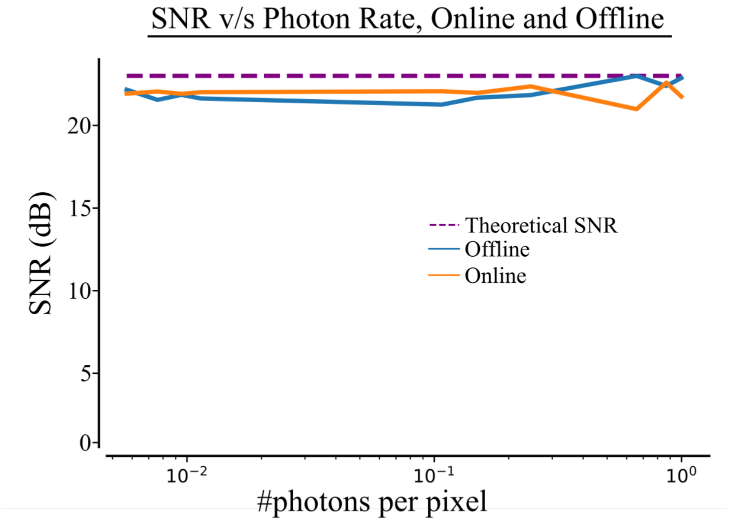
For online detection we use code modified from [33] (Commit: 7d21606859feb63eba6d9d19942938873915f8dc). Fig. 1 shows the results of using the online changepoint detection algorithm. Observe that some of the edge details are better preserved with the offline changepoint algorithm.

We run the same experiment as in Fig. 4, where an orange is rotated at different brightness levels. We show the resulting SNR for online vs offline detection in supplementary Fig. 2



(a) “Fan” (b) “Checkerboard”

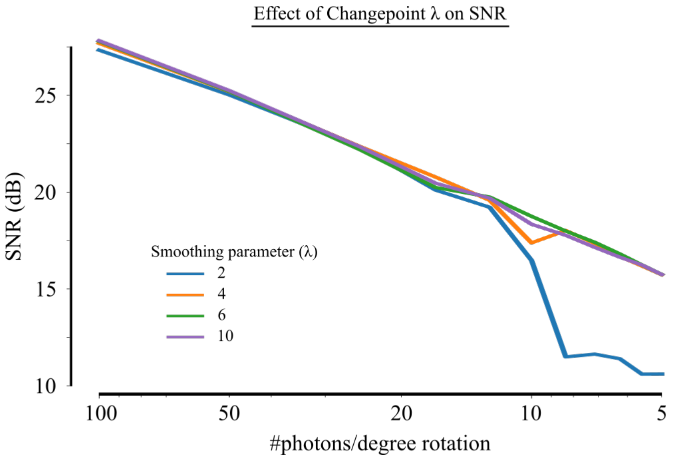
Supplementary Figure 1. **Comparing online vs. offline changepoint detection.** We processed the two experimental datasets using our online and offline changepoint detection algorithms. There is a slight loss of edge details when the online algorithm is used.



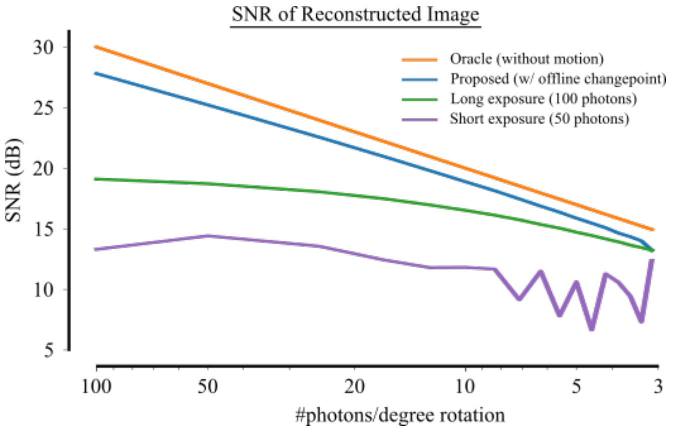
Supplementary Figure 2. **Online vs. Offline Rotating Orange SNR** We run the same experiment as in Fig. 4, with the rotating orange at many light levels. Note that the resulting SNRs of the two methods are quite similar.

S.2. SNR Analysis

We test our deblurring algorithm for different motion speeds for the case of rotational motion using the "orange" dataset in the main text. We measure the SNR by computing the discrepancy between the ground truth flux image and the deblurred result. We do this by temporally downsampling the original photon frames, so the number of photons decrease as the motion speeds up. Suppl. Fig. 3 shows how changing the regularization parameter λ in the offline flux changepoint detection algorithm effects the SNR. We find that as long as λ is high enough a good reconstruction SNR stays high. In Suppl. Fig. 4 we show that our algorithm converges to the performances of a long exposure capture (with motion blur) if the number of photons per degree of rotation falls below 3.

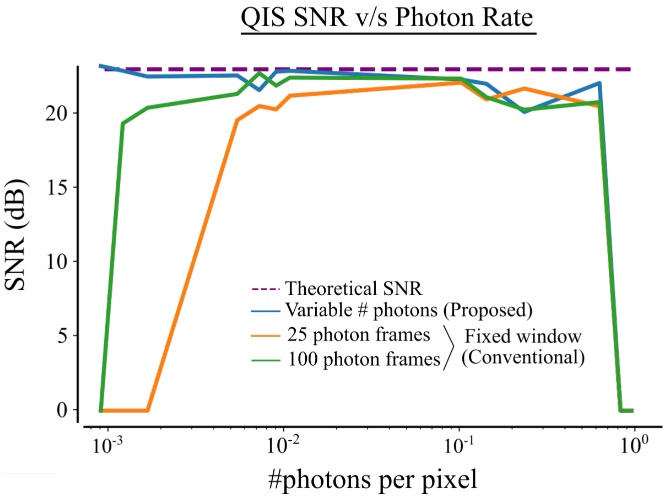


Supplementary Figure 3. **Effect of offline changepoint algorithm regularization parameter λ.** If λ is large enough we get good performance. When λ is too small, many flux changepoints are found, which will cause the CPV to be too noisy to properly align frames.



Supplementary Figure 4. Effect of number of photons captured per degree of rotation. Our algorithm is unable to find flux changepoints at speeds of 3 photon frames per degree. When no flux changepoints are found we just get a long exposure image. We also see that the SNR for a blurry image (long exposure) or a noisy image (short exposure) is worse than the proposed deblurring method until our method converges to the long exposure image.

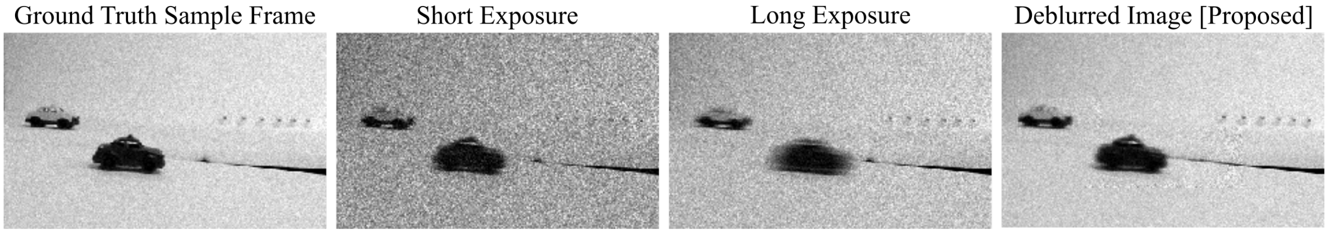
We run the same experiment as in Fig. 4, where an orange is rotated at different brightness levels. We test our offline QIS changepoint detection method by removing the timing information and only considering a binary output. Our adaptive changepoint method helps at low light levels, see Fig. 5.



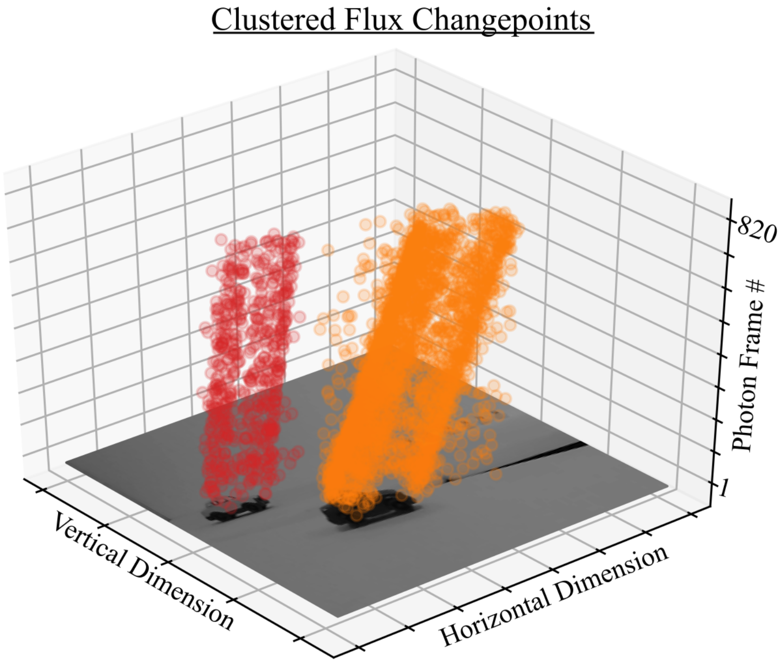
Supplementary Figure 5. **QIS, SNR vs. Brightness.** We run the same experiment as in Fig. 4, with the rotating orange at many light levels. We remove timing information from the raw data to create QIS binary data frames and run the same deblurring experiment. Again our adaptive changepoint method is helpful in low light scenarios. Note that the sharp drop on the right is due to saturation of a QIS sensor.

S.3. Additional Simulation Results

This section contains some additional simulated results. A second scene with two toy cars is displayed, we use the same parameters as the toy car scene in the main text for frame generation, changepoint detection and deblurring. For this scene the dark car moves 90 pixels and has a contrast 3.3. The bright car has a contrast of 1.2 and moves 30 pixels. Our results are shown in supplementary Fig. 6 and the clustered changepoints are shown in supplementary Fig. 7. Again, our method is able to deblur both moving cars.

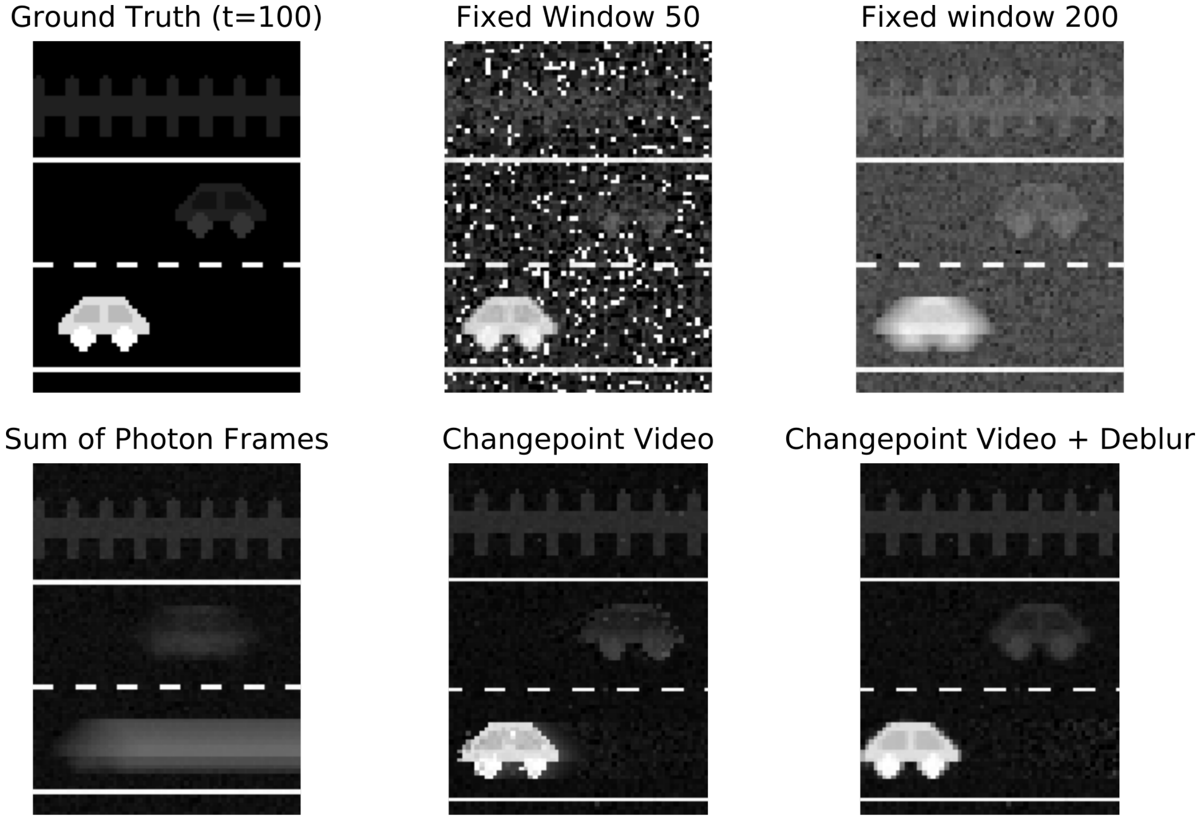


Supplementary Figure 6. **Simulated Multiple Objects.** We simulate SPAD data from a 240 fps phone video of two rolling toy cars, a fast moving dark car and slow moving bright car. From left to right, the ground truth image shows the result of generating the same number of photon frames from the first frame of the video sequence. The short and long exposure images show the results of using only the first 75 and 250 photon frames, respectively. Notice that the short exposure preserves the dark car while the bright car is quite noisy, on the other hand, the long average blurs the dark car but preserves details of the bright one better. Finally, our deblurring algorithm is able to reconstruct both the dark and bright car.



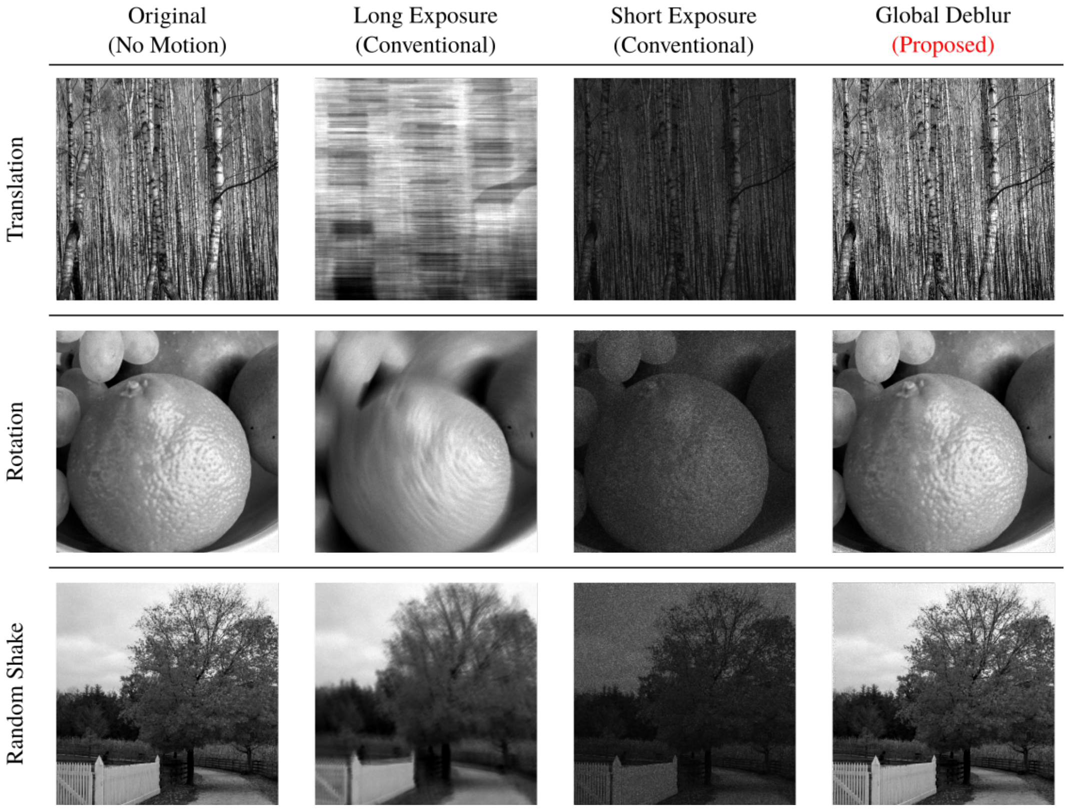
Supplementary Figure 7. **Clustered Flux Changepoints** Displayed are the 2 flux changepoint clusters found for the scene in Supplementary Fig. 6. We only display half of the flux changepoints in each cluster for visualization purposes.

We simulate a simple pixel art scene to demonstrate the advantage of deblurring on a changepoint video rather than burst frames. Notice in supplementary Fig. 8 that the changepoint video frame is able to capture a wide range of motion speeds and contrasts that a single fixed frame cannot capture.



Supplementary Figure 8. **Pixel Art Multiple Objects.** We simulate a scene where a bright car moves quickly to the right and a dim car moves slowly to the left. Notice that in the short averaging window, the dim car is lost in the noise while in the long averaging window the bright car is blurred. The sum of all photon frames maintains the background quite well. The changepoint video frame adapts to motion in each pixel and captures both the bright car, the dim car, and the background. Notice that the changepoint video frame loses some of the structure of the dim car due to noisy changepoints. We combine the adaptive changepoint video with a deblurring algorithm to deblur both cars.

S.3.1. Global Motion Results



Supplementary Figure 9. **Simulated motion deblurring results for different types of global motion.** From left the right the columns show, a ground truth image that shows the result if the same number of photons are sampled from the original image but with no motion. A long exposure where all photon frames are summed, the result is a blurred image. A short exposure image shows the combination of the first 20 photons of the timestamp data, notice the edges are sharp but noise dominates. The result of the global motion deblur algorithm is shown in the last column. (Original images from FreeImages.com)

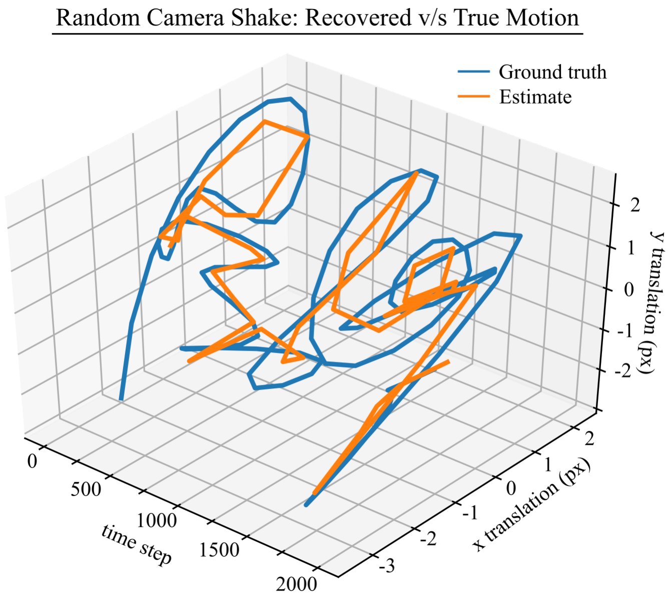
We start with a ground truth high resolution image, successively apply rigid transformations (using known rotations and translations), and generate photon frames using exponentially distributed arrival times. We reassign these high spatial resolution timestamps a lower resolution 2D array to simulate a low resolution SPAD pixel array.

We model a photon frame readout from a SPAD array with 8000 bins and bin width of $256\,\mathrm{ps}$. Images are scaled so that the true photon flux values ranges between 10^4 and 10^8 photons per second. We then iteratively transform the flux image according to known motion parameters, and downsample spatially to a resolution of 425×425 before generating photon timestamps.

For the horizontal translation blur, we moved the image 1 pixel to the right blur, we rotate the image 0.1 degree for every 10 generated photons for a total of 1000 photons. To emulate random camera shake, we create random motion trajectories by drawing two i.i.d. discrete uniform random variables between -3 and 3 and use that as the number of pixels to translate along the horizontal and vertical directions. We generate 20 photons per translation for a total of 2000 photons. We use the BOTTOMUP algorithm [29] with $\lambda = 5$ for the changepoint detection step. In practice we found that the results were not very sensitive to the choice of λ and values between 2 and 12 produced similar results.

We generate photon events from an exponential distribution. We transform the flux image, then down-sample to simulate objects with more detail than pixel resolution. We then generate 10-20 photons from the down-sampled flux image. Continuing this we get a 3-d tensor of photons representing global motion of the original image.

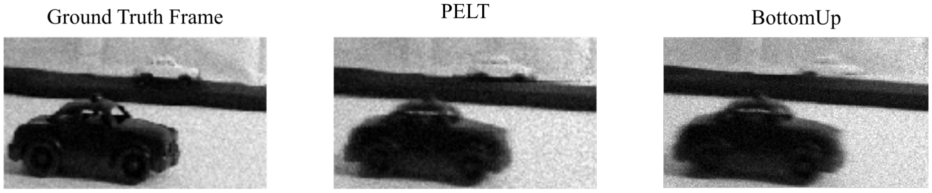
Supplementary Fig. 9 shows simulated deblurring results for three different motion trajectories. The top row shows a case of horizontal translation: conventional long/short exposures must trade off motion blur and shot noise. Our deblurring method reproduces sharp details, such as vertical lines of the tree stems. The second row shows a case of rotation: note that different pixels of the scene now undergo different amount of motion per unit time. Our method reconstructs fine details of the texture of the orange peel. The bottom row shows random camera shake with unstructured motion. Our technique is able to correct for this global motion by approximating the overall motion trajectory as a sequence of small translations and rotations. Supplementary Fig. 10 shows the comparison between the true motion trajectory and the trajectory estimated as part of our deblurring algorithm.



Supplementary Figure 10. **Comparison of estimated and true motion trajectories.** This plot shows the true and estimated motion trajectories for the random shake case in Fig. 9. The recovered motion tracks the ground truth motion quite well.

S.4. Comparison of BottomUp and PELT

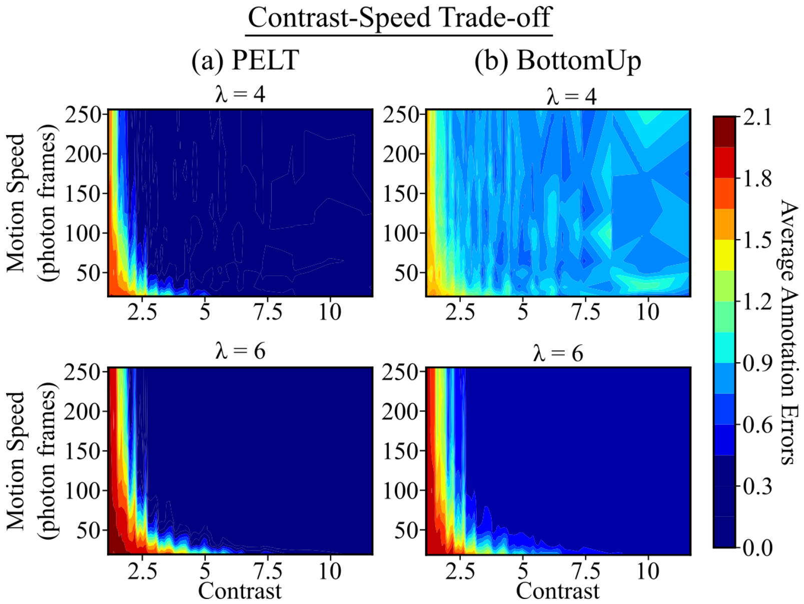
In this section we run some of the same simulation experiments from the main text but with the BottomUp algorithm instead of the PELT algorithm. In Suppl. Fig. 11 we compare the ability of both algorithms on the toy car scene, BottomUp seems to produce slightly more noisy and blurry results. In Suppl. Fig. 12 we re-run the contrast vs. speed simulations and find that BottomUp does comparably well with slightly more false positives.


Ground Truth Frame

PELT

BottomUp

Supplementary Figure 11. **Toy Car Simulated Scene.** Here are the results using the toy car simulated scene. The parameters used are the same as in the main text. Note that the BottomUp algorithm is able to detect and deblur both cars; however, it seems to produce a slightly noisier and blurrier result. For this scene 210 by 300 with 690 photon frames per pixel, on our unoptimized system with 30 parallel cores, it takes the PELT algorithm two minutes to run while the BottomUp takes approximately one minute.


Contrast-Speed Trade-off

(a) PELT

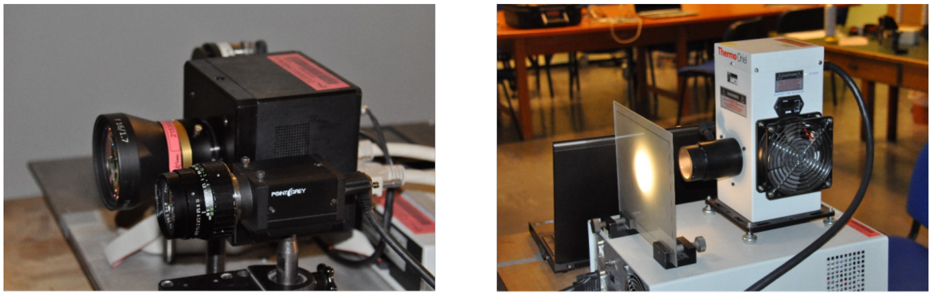
(b) BottomUp

Supplementary Figure 12. **BottomUp vs. PELT - Contrast vs. Speed.** We repeat the contrast vs. speed simulation as done in the main text but with the BottomUp algorithm. Note that the BottomUp algorithm does comparably well for $\lambda = 6$. For $\lambda = 4$, the BottumUp algorithm is able to detect more difficult objects at the expense of false positives during easier scenarios.

S.5. Experiment Setup

For the experimental investigation we used a 32×32 InGaAs SPAD array from Princeton Lightwave (PL GM-APD 32 x 32 Geiger-Mode Flash 3-D LiDAR Camera) and an RGB camera (VIS/BW point gray Grashopper 3, GS3-U3-23S6M-C) capturing the same field-of-view. The SPAD camera samples photon events with a depth of 10^{13} bins and $250\,\mathrm{ps/bin}$. Further, during the experiments we used a frame readout rate of $50\,\mathrm{kHz}$. The InGaAs sensor is sensitive in near infrared (NIR) to shortwave infrared (SWIR) wavelengths ranging from $900\,\mathrm{nm}$ to $1.6\,\mu\mathrm{m}$.

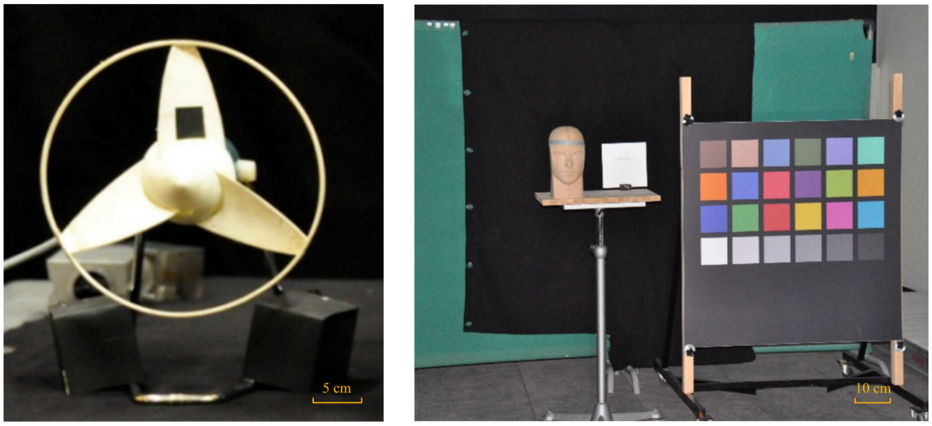
During the measurements we investigated two different type of scene setups: a "fan" and a "checkerboard" scene, as depicted in Suppl. Fig. 13. In the first scene, the fan consists of three blades mounted on a central cone and is enclosed by a circular frame with a diameter of $18\,\mathrm{cm}$. One blade was marked with a black piece of paper. The fan scene was used to investigate rotational motion. The second scene consists of an artificial head, a white plate and a colored checkerboard. Colors appear at a different gray levels in the SWIR wavelength images. This second scene was used to investigate random motion due to a horizontally shaking camera.



(a) Camera Setup

(b) Ambient Illumination Source

Supplementary Figure 13. **Hardware setup** (a) Our hardware setup consists of a Princeton Lightwave SPAD array (PL GM-APD 32×32 Geiger-Mode Flash 3-D LiDAR Camera) and an RGB camera (VIS/BW point gray Grashopper 3, GS3-U3-23S6M-C) capturing the same field-of-view. (b) Ambient illumination is provided by a diffuse light source (broadband arc lamp ThermoOriel Model 66881).



(a) "Fan" scene

(b) "Checkerboard" scene

Supplementary Figure 14. **Experimental Scenes** (a) The "fan" scene consists of a small fan with a black square patch on one of the fan blades. (b) The "checkerboard" scene consists of a large color checkerboard and a mannequin head.

S.6. Description of Video Results

Please refer to included `.txt` and `.mp4` files for supplementary video results.

Supplement References

[31] Ryan P. Adams and David J. C. MacKay. Bayesian online changepoint detection. *arXiv: Machine Learning*, 2007.

[32] Atul Ingle, Andreas Velten, and Mohit Gupta. High flux passive imaging with single-photon sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6760–6769, 2019.

[33] Johannes Kulick, nariox, Dan Marthaler, Minesh A. Jethva, and Sean Kruzel. bayesian changepoint detection. `https://github.com/hildensia/bayesian_changepoint_detection`, 2020.

[34] Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.