sensors

Article

Vision without the Image

Bo Chen and Pietro Perona *

Computation and Neural Systems, California Institute of Technology, 1200 E California Blvd, Pasadena, CA 91125, USA; bchen3@caltech.edu
* Correspondence: perona@caltech.edu; Tel.: +1-626-395-2084

Academic Editor: Eric R. Fossum
Received: 2 February 2016; Accepted: 4 April 2016; Published: 6 April 2016

Abstract: Novel image sensors transduce the stream of photons directly into asynchronous electrical pulses, rather than forming an image. Classical approaches to vision start from a good quality image and therefore it is tempting to consider image reconstruction as a first step to image analysis. We propose that, instead, one should focus on the task at hand (e.g., detection, tracking or control) and design algorithms that compute the relevant variables (class, position, velocity) directly from the stream of photons. We discuss three examples of such computer vision algorithms and test them on simulated data from photon-counting sensors. Such algorithms work just-in-time, *i.e.*, they complete classification, search and tracking with high accuracy as soon as the information is sufficient, which is typically before there are enough photons to form a high-quality image. We argue that this is particularly useful when the photons are few or expensive, e.g., in astronomy, biological imaging, surveillance and night vision.

Keywords: photon-counting sensors; visual recognition; low-light computer vision

1. Introduction

Current computer vision algorithms start with a high-quality image as input. While such images may be acquired almost instantly in a well-lit scene, dark environments demand a significantly longer acquisition time. This long acquisition time is undesirable in many applications that operate in low-light environments: in biological imaging, prolonged exposure could cause health risks [1] or sample bleaching [2]; in autonomous driving, the delay that is imposed by image capture could affect a vehicle's ability to stay on-course and avoid obstacles; in surveillance, long periods of imaging could delay response, as well as produce smeared images. When light is low, the number of photons per pixel is small and images become noisy. Computer vision algorithms are typically not designed to be robust *vis-a-vis* image noise, thus practitioners face an uneasy tradeoff between poor performance and long response times.

Novel sensor technology offers a new perspective on image formation: as soon as a photon is sensed it should be transmitted to the host Central Processing Unit (CPU), rather than wait until a sufficient number of photons has been collected to form good quality image. Thus an image, in the conventional sense, is never formed. Designs and prototypes of photon-counting image sensors, such as the quantum sensors [3], single-photon avalanche detectors [4], quanta image sensors [5,6], and the giga-vision camera [7], have been proposed recently. These sensors are capable of reliably detecting single photons, or a small number of photons. Instead of returning a high-quality image after a long exposure, photon-counting sensors report a stream of photon counts densely sampled in time.

Currently, the dominant use for photon-counting image sensors is image reconstruction [8]: the stream of photon counts is used to synthesize a high-quality image to be used in consumer applications or computer vision. However, the goal of vision is to compute information about the world (class, position, velocity) from the light that reaches the sensor. Thus, reconstructing the image is not a necessary first step. Rather, one should consider computing information directly from the stream of photons [9,10]. This line of thinking requires revisiting the classical image-based paradigm of computer vision, and impacts both the design of novel image sensors and the design of vision algorithms.

Computing directly from the stream of photons presents the advantage that some information may be computed immediately, without waiting for high-quality image to be formed. In other words, information is computed incrementally, enabling the downstream algorithms to trade off reaction times with accuracy. This is particularly appealing in low-light situations were photon arrival times are widely spaced. As the hardware for computation becomes faster, this style of computation will become practical in brighter scenes, especially when response times are crucial (e.g., in vehicle control).

Here we explore three vision applications: classification, search and tracking. In each application, we will propose an algorithm that makes direct use of the stream of photons, rather than an image. We find that each one of these algorithms achieves high accuracy with only a tiny fraction of the photons required for capturing high-quality images. We conclude with a discussion of what was learned.

2. Results

2.1. Simplified Imaging Model

Assume that the scene is stationary and photon arrival times follow a homogeneous Poisson process. Within an interval of length δ_t, the observed photon count X_i at pixel location i is subject to Poisson noise whose mean rate depends on maximum rate $\lambda_{max} \in \mathbb{R}^+$, the true intensity at that pixel $I_i \in [0,1]$ and a dark current rate $\epsilon_{dc} \in [0,1]$ per pixel [8]:

$$P(X_i = k) = Poisson(k; \lambda_{max} \frac{(I_i + \epsilon_{dc})}{(1+\epsilon_{dc})} \delta_t) \quad (1)$$

(a model including sensor read noise is described in Section 3.1).

The sensor produces a stream of images $\mathbf{X}_1, \mathbf{X}_2, \ldots$, where $\mathbf{X}_t \in \mathcal{N}^d$ contains the photon counts from d pixel locations from the time interval $[(t-1)\delta_t, t\delta_t]$ (Figure 1a). We use $\mathbf{X}_{1:t}$ to represent the stream of inputs $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_t\}$.

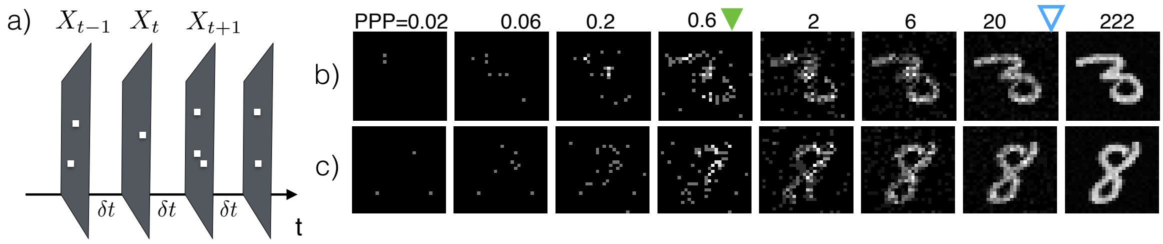

Figure 1. Synthetic low-light images. (**a**) A photon-counting sensor outputs a matrix of photon counts $\mathbf{X}_t$ with a period of δt. (**b**,**c**) Sample synthetic low-light images from the Mixed National Institute of Standards and Technology (MNIST) dataset [11] used in the classification experiments (Section 2.2.2) with increasing average photons per pixel (PPP). PPP is proportional to the exposure time t. Blue hollow arrows indicate the median PPP required for the proposed algorithm (Section 2.2.1) to achieve the same error rate (0.7%) as a model trained and tested using images under normal lighting conditions with about $2^7 \approx 10^4$ PPP. Green solid arrows indicate the median PPP required to to maintain error rates below 1%.

When the illuminance of the environment is constant, the expected number of photons collected by the sensor grows linearly with the exposure time. Hence we use the number of photons per bright pixel (PPP) as a proxy for the exposure time t. PPP $= 1$ means that the a pixel with maximum intensity has collected 1 photon. Additionally, since PPP is linked to the total amount of information content in the image regardless of the illuminance level, we will use PPP when describing the performance of vision algorithms. Figure 1b, c shows two series of inputs $\mathbf{X}_{1:t}$ with increasing PPP.

2.2. Classification

Distinguishing objects of different categories hinges upon the extraction of "features", which are structural regularities in pixel values such as edges, corners, contours, *etc*. For example, the key feature that set apart a handwritten digit "3" from a digit "8" (Figure 1b,c, last column) is the fact that a "3" has open loops and "8" has closed loops—This corresponds to different strokes on the left side of the digit. In normal lighting conditions, these features are fully visible, may be computed by, e.g., convolution with an appropriate kernel, and fed into a classifier to predict the category of the image.

In low light, classification is hard because the features are corrupted by noise. A closed contour may appear broken due to stochastically missing photons. The noise in the features in turn translates to uncertainties on the classification decision. This uncertainty diminishes as the exposure time increases. It is intuitive that a vision algorithm that is designed to compute from a minimal number of photons should keep track of said uncertainties, and dynamically determine the exposure time based on the desired accuracy.

In particular, one wishes to predict the category $Y \in \{1, 2, \ldots, C\}$ of an image based on photon counts $\mathbf{X}_{1:t}$. The predictions must minimize exposure time while being reasonably accurate, *i.e.*,

$$\min \mathbb{E}[T] \qquad\qquad s.t.\ \ \mathbb{E}[\hat{Y} \neq Y] \leq \gamma \tag{2}$$

where T is a random variable denoting the exposure time required to classify an image, $\hat{Y} \in \{1, 2, \ldots, C\}$ is the prediction of the class label, γ is the maximum tolerable misclassification error, and the expectation is taken over all images in a dataset.

2.2.1. Classification Algorithm

In order to make the most efficient use of photons, we first assume that a conditional probabilistic model $P(Y|\mathbf{X}_{1:t})$ is available for any $t \geq 0$ (we will relax this assumption later) and for all possible categories of the input image. An asymptotically optimal algorithm that solves the problem described in Equation (2) is Sequential Probability Ratio Testing (SPRT) [12] (Figure 2a,b):

$$\begin{aligned} &\text{Choose an appropriate error threshold } \ \theta \\ &c^* = \underset{c \in \{1,2,\ldots,C\}}{\arg\max}\ P(Y = c|\mathbf{X}_{1:t}) \\ &\begin{cases} \text{report } Y = c^* & \text{if } \log \frac{P(Y=c^*|\mathbf{X}_{1:t})}{P(Y \neq c^*|\mathbf{X}_{1:t})} > \theta \\ \quad\text{increase } t & \qquad\text{otherwise} \end{cases} \end{aligned} \tag{3}$$

Essentially, SPRT keeps accumulating photons by increasing exposure time until there is predominant evidence in favor of a particular category. Due to the stochasticity of the photon arrival events and the variability in an object's appearance, the algorithm observes a different stream of photon counts each time. As a result, the exposure time T, and equivalently, the required PPP, are also different each time (see Figure 3).

The accuracy of the algorithm is controlled by the threshold θ. When a decision is made, the declared class c^* satisfies that $\log \frac{P(Y=c^*|\mathbf{X}_{1:t})}{P(Y \neq c^*|\mathbf{X}_{1:t})} > \theta$, which means that class c^* has at least posterior probability $Sigm(\theta) \triangleq \frac{1}{1+e^{-\theta}}$ according to the generative model, and the error rate of SPRT is at most $1 - Sigm(\theta)$. For instance, if the maximum tolerable error rate is 10%, θ should be set so that $1 - Sigm(\theta) = 0.1$, or $\theta \approx 2.2$, while an error rate of 1% would drive θ to 4.6. Since higher thresholds lead to longer exposure times, the threshold serves as a knob to trade off speed *versus* accuracy, and should be set appropriately based on γ (Equation (2)).

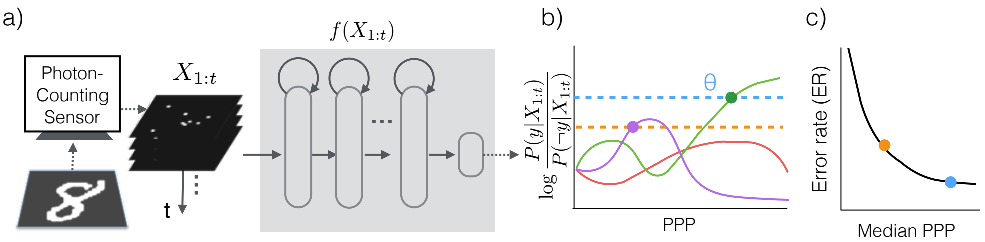

Figure 2. Low-light classification. (**a**) A sensor produces a stream of photon counts $\mathbf{X}_{1:t}$, which is fed into a recurrent neural network to compute class conditional likelihood $f(\mathbf{X}_{1:t}) \approx P(Y|\mathbf{X}_{1:t})$. (**b**) SPRT with three classes. Based on $f(\mathbf{X}_{1:t})$, SPRT compares the class with the highest log likelihood ratio to a threshold θ. Two threshold options (dashed lines) corresponds to different decisions at different times (solid dots). (**c**) The threshold allows the system to traverse the error rate (ER) *versus* PPP curve to find the optimal operating point to minimize the cost function (Equation (2)). We use PPP instead of the exposure time T to measure speed as the former is more closely related to the information content in the photon stream. We use the median PPP instead of the mean because PPP follows a heavy-tailed distribution (Figure 3c) and median is more stable than the mean.

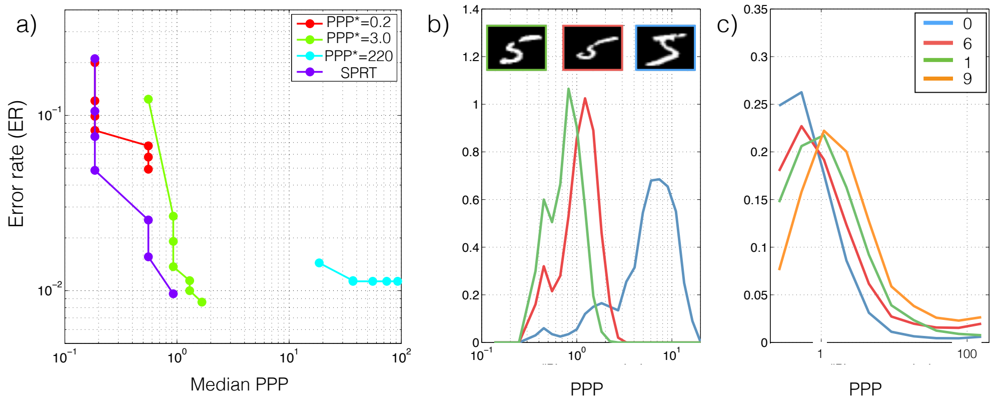

Figure 3. Low-light classification performance. (**a**) Error rate *vs.* PPP tradeoff for the SPRT algorithm (Equation (3)). $PPP^* = x$ denotes a "specialist": A model trained using images only at light level x and tested on other light levels by input normalization (see Section 2.2.2). (**b**) SPRT decision time is stochastic even for the same underlying image. The PPP distribution is plotted separately for multiple images of 5. (**c**) SPRT decision time distribution is category-dependent. Some categories, e.g., "0", are easier (faster decision) than others, say "9".

The assumption that the conditional distribution $P(Y = c|\mathbf{X}_{1:t})$ is known is rather restrictive. Fortunately, the conditional distribution may be directly learned from data. In particular we train a recurrent neural network [13] $f_c(\mathbf{X}_{1:t}) \approx P(Y = c|\mathbf{X}_{1:t})$ to approximate the conditional distribution. This network has a compact representation, and takes advantage of the sparseness of the photon-counts for efficient evaluation. Details of the network may be found in Section 3.2 and [9].

2.2.2. Experiments

We evaluate the low-light classification performance of the SPRT on the MNIST dataset [11], a standard handwritten digits dataset with 10 categories. The images are 28 × 28 in resolution and in black and white. We simulate the outputs from a photon-counting sensor according to the full noise model (Section 3.1). The images are stationary within the imaging duration. We do not assume a given conditional distribution $P(Y|\mathbf{X}_{1:t})$ but train a recurrent network approximation $f(\mathbf{X}_{1:t})$ from data (Section 3.2).

Recall that classification correctness in each trial and the required exposure time (or PPP) are random variables. We therefore characterize SPRT performance based on the tradeoff between error rates (ER, $\mathbb{E}[\hat{Y} \neq Y]$) and the median PPP in Figure 3a. The tradeoff is generated by sweeping the thresholds $\theta \in [-2.2, 9.2]$. For comparison we tested the performance of models that were trained to classify images from a single PPP. We call these models "specialists" for the corresponding PPP. The specialists are extended to classify images at different light levels by scaling the image to the specialized PPP. To get a sense of the intraclass and interclass PPP variability, we also visualize the PPP histograms for multiple runs of different images in the same class (Figure 3b), and the overall PPP histograms for a few classes (Figure 3c). Lastly, we analyze how SPRT's performance is sensitive to sensor noises in Figure 4. Details of the analysis procedure are found in Section 3.1.

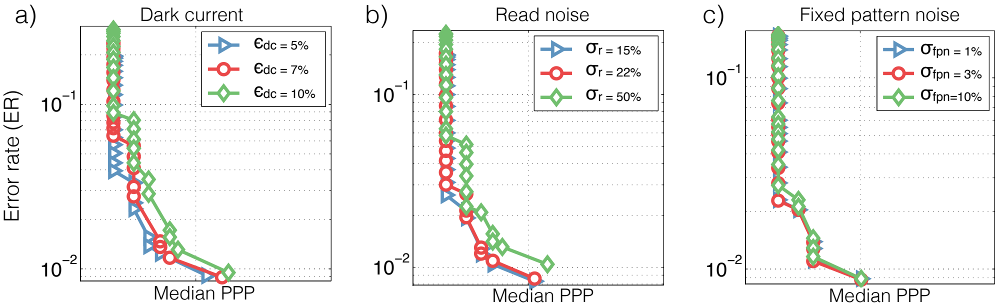

Figure 4. Sensitivity to noise in image classification. Error rate *vs.* PPP tradeoff with different levels of (**a**) dark current ϵ_{dc}, (**b**) read noise and (**c**) fixed pattern noise (see Section 3.1). The default setting uses 3% dark current, 0% read noise and 0% fixed pattern noise.

2.3. Search

Search is a generalization of classification into multiple locations. The task is to identify whether a target object (e.g., keys, a pedestrian, a cell of a particular type) is present in a scene cluttered with distractors (e.g., a messy desk, a busy street at night or a cell culture). Note that despite the multiple candidate positions for a target to appear, we consider search as a binary task, where the two hypotheses are denoted $C = 1$ (target-present) and $C = 0$ (target-absent). We assume for simplicity that at most one target may appear at a time (for multiple targets, see [14]).

The difficulty of search in low-light conditions may be attributed to the following factors. (1) There are multiple objects in the display, and each object is subject to photon count fluctuations. (2) Long range constraints, such as the prior knowledge that at most one target is present in the visual field, must be enforced. (3) Properties of the scene, such as the amount of clutter in the scene and the target and distractor appearance, may be uncertain. For example, we may know that there may be either three or twelve objects in the scene, and intuitively the search strategy for these two scenarios should be drastically different. Therefore, scene properties must be inferred for optimal performance.

We assume that a visual field consists of L non-overlapping locations, out of which M locations may contain an object. M represents the amount of clutter in the scene. The objects are simplified to be oriented bars and the only feature that separates a target from a distractor is the orientation. The orientation at location l is denoted $Y^{(l)}$. The target orientation and the distractor orientation are denoted y_T and y_D, respectively. The scene properties are collected denoted $\phi = \{M, y_T, y_D\}$. The scene properties may be unknown for many search tasks, thus ϕ is a vector of random variables. The variable of interest is $C \in \{0, 1\}$: $C = 1$ iff $\exists l \in \{1, \ldots, L\}, Y^{(l)} = y_T$, (*i.e.*, $C = 1$ iff there exists a location that contains a target).

We also assume that a low-light classifier discussed in Section 2.2 has been developed for classifying bar stimulus: the classifier computes $f_y(\mathbf{X}_{1:t}^{(l)}) \approx P(Y^{(l)} = y | \mathbf{X}_{1:t}^{(l)})$, the probability that the bar orientation at location l is y conditioned only on the local photon counts $\mathbf{X}_{1:t}^{(l)}$.

2.3.1. Search Algorithm

Similar to the low-light classification problem, an asymptotically optimal search algorithm is based on SPRT. The detailed algorithm is [12,14]:

$$\begin{aligned} &\text{Choose two error thresholds } \ \theta_0 < 0, \theta_1 > 0 \\ &\text{Compute } S(t) \triangleq \log \frac{P(C=1|\mathbf{X}_{1:t})}{P(C=0|\mathbf{X}_{1:t})} \\ &\begin{cases} \text{report } C=1 & \text{if } S(t) > \theta_1 \\ \text{report } C=0 & \text{if } S(t) < \theta_0 \\ \text{increase } t & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

where $S(t)$ is the log likelihood ratio between the two competing hypotheses, target-present ($C = 1$) and target-absent ($C = 0$). This algorithm is a binary version of the classification algorithm in Equation (3). Similar to Equation (3), the two thresholds θ_0 and θ_1 controls the amount of false reject errors (*i.e.*, declare target-absent when target-present) and false accept errors (*i.e.*, declaring target-present when target-absent).

The key for SPRT is to compute $S(t)$ from photon counts $\mathbf{X}_{1:t}$. The inference procedure may be implemented by two circuits, one infers the scene properties ϕ, and the other computes $S(t)$ (see [14]):

$$S(t) = \log \frac{1}{L} \sum_{l,\phi} R^{(l,\phi)}(\mathbf{X}_{1:t}^{(l)}) P(\phi|\mathbf{X}_{1:t}) \tag{5}$$

where

$$R^{(l,\phi)}(\mathbf{X}_{1:t}^{(l)}) \triangleq \frac{\sum_y f_y(\mathbf{X}_{1:t}^{(l)}) P(Y^{(l)} = y|\phi, C^{(l)} = 1)}{\sum_y f_y(\mathbf{X}_{1:t}^{(l)}) P(Y^{(l)} = y|\phi, C^{(l)} = 0)} \tag{6}$$

$$P(\phi|\mathbf{X}_{1:t}) \propto P(\phi) \prod_l \sum_y f_y(\mathbf{X}_{1:t}^{(l)}) \frac{P(Y^{(l)} = y|\phi, C^{(l)} = 0)}{P(Y^{(l)} = y)} \tag{7}$$

Therefore, $S(t)$ may be computed by composing the low-light classifiers $f_y(\mathbf{X}_{1:t}^{(l)})$ according to Equations (5)–(7). The probabilities used in Equations (6) and (7), such as $P(Y^{(l)} = y)$ and $P(Y^{(l)} = y|\phi,\ C = 2)$, may be estimated from past data.

2.3.2. Experiments

We choose a simple setup (Figure 5a) to illustrate how the performance of the search algorithm is affected by scene properties: the amount of clutter M, the target/distractor appearances y_T and y_D, as well the degree of uncertainty associated with them. The setup contains $L = 14$ locations, each occupying a 7×7 area from which the sensor collects photons. The area contains a 3×7-pixel bar with intensity 1 and background pixels with intensity 0. The max emission rate is $\lambda_{max} = 3$ photons/s, and the dark current is 50% (causing the background to emit 1 photon/s). Examples of the lowlight search setup are shown in Figure 6d.

We conduct two experiments, one manipulates the scene complexity M and the other target/distractor appearances. In the first experiment M is either chosen uniformly from $\{3, 6, 12\}$, or fixed at one of the three values (Figure 6a,b): (1) Despite the high dark current noise, a decision may be made quickly with less than 2 photons per pixel. (2) The amount of light required to achieve a given classification error increases as M. (3) Not knowing the complexity further increases the required photon count. (4) Target-absent conditions requires more photons than target-present conditions. In the second experiment the target-distractor appearance difference $\delta y = |y_T - y_D|$ is either chosen uniformly from $\{20^\circ, 30^\circ, 90^\circ\}$ or fixed at one of the three. Figure 6c suggests that target dissimilarity heavily influences the ER-PPP tradeoff, while uncertainty in the target and distractor appearances does not.

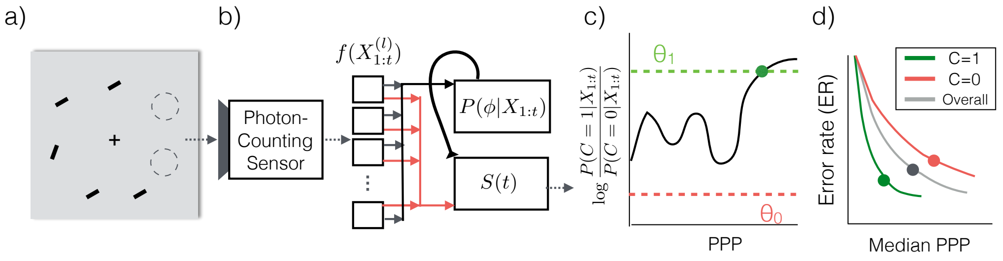

Figure 5. low-light Search. (**a**) Search stimuli with $L = 7$ display locations and 5 objects (oriented bars). The dashed circles are not part of the display but used to indicate empty locations. (**b**) The search algorithm. Local classification results $f(\mathbf{X}^{(l)}_{1:t})$ go through two circuits, one estimates the scene properties $P(\phi|\mathbf{X}_{1:t})$ (Equation (7)) and sends feedback to the other circuit that computes the log likelihood ratio $S(t)$ (Equation (5)). (**c**) SPRT compares $S(t)$ against a pair of thresholds θ_1 and θ_0 to decide whether to declare target-presence or absence, or wait for more evidence (Equation (4)). (**d**) SPRT produces ER *vs.* PPP tradeoffs (sketch) for different conditions.

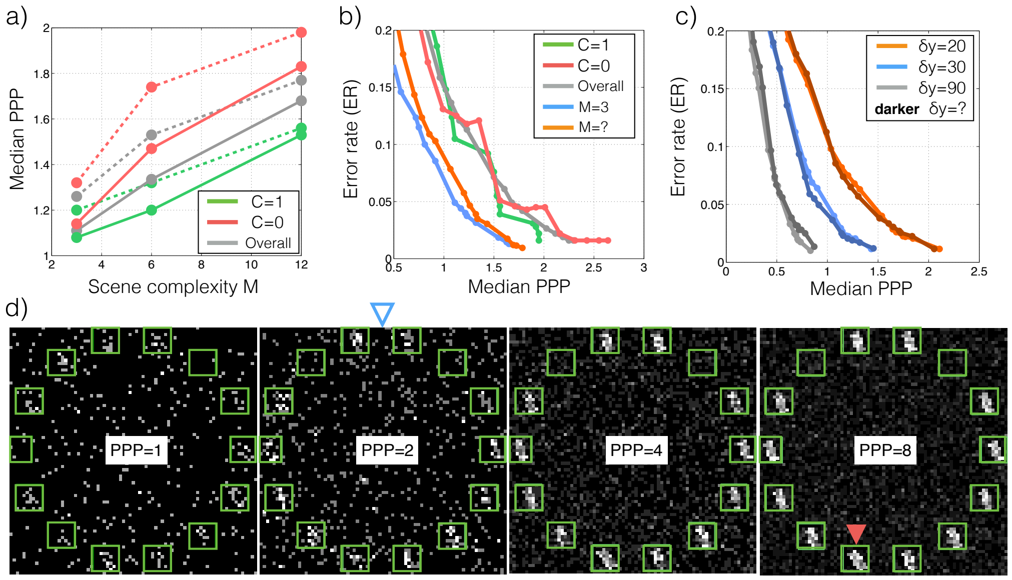

Figure 6. Characteristics of the search algorithm. (**a**) Median PPP required to achieve 5% error rate as a function of scene complexity M. Solid/dashed lines represent search problems where the complexity is known/unknown in advance. (**b**) ER *vs.* PPP tradeoff for various conditions. The first three legends corresond to target-present (C = 1), target-absent (C = 0) and their average for $M = 12$. The last two corresponds to a simpler image containing three bars with the complexity known ($M = 3$) and unknown ($M =$?) in advance. (**c**) ER *vs.* PPP tradeoff for different appearance differences $\delta y = |y_T - y_D|$ between the target and the distractor. Darker lines denote the cases where the difference is unknown before hand. (**d**) Examples of lowlight search stimuli as a function of PPP. $L = 14$, $M = 12$, $\delta y = 20°$, target-present (C = 1, location indicated by red arrow, same for every PPP). The photon-counting sensors receive inputs from within the green windows. With known complexity the algorithm achieves a 1% error rate with a median PPP of 2 (blue hollow arrow).

2.4. Tracking

Finally, we demonstrate the potential of photon-counting sensors in tracking under low-light conditions. The goal of tracking is to recover time-varying attributes (*i.e.*, position, velocity, pose, *etc.*) of one or multiple moving objects. It is challenging because, unlike classification and search, objects in tracking applications are non-stationary by definition. In low-light environments, as the object transitions from one state to another, it leaves only a transient footprint, in the form of stochastically-sprinkled photons, which is typically insufficient to fully identify the state. Instead, a tracker must postulate the object's dynamics and integrate evidence over time accordingly. The evidence in turn refines the estimates of the dynamics. Due to the self-reinforcing nature of this procedure, the tracker must perform optimal inference to ensure convergence to the true dynamics.

Another challenge that sets low-light tracking apart from regular tracking problems is that the observation likelihood model is not only non-Gaussian, but also often unavailable, as it is commonly the case for realistic images. This renders most Kalman filter algorithms [15] ineffective.

2.4.1. Tracking Algorithm

The tracking algorithm we have designed is a hybrid between the Extended Kalman Filter [15] and the Auxiliary Particle Filter [16]. Let $\mathbf{Z}_t$ denote the state of the object, F the forward dynamics that govern the state transition: $\mathbf{Z}_{t+1} = F(\mathbf{Z}_t)$, which are known and differentiable, and $P_t(Z)$ the posterior distribution over the states at time t: $P_t(Z) \overset{\triangle}{=} P(\mathbf{Z}_t = Z|\mathbf{X}_{1:t})$.

We make two assumptions: (1) P_t may be approximated by a multivariate Gaussian distribution; (2) A low-light regressor $f(Z|\mathbf{X}_t) \approx P(\mathbf{Z}_t = Z|\mathbf{X}_t)$ is available to compute a likelihood score of $\mathbf{Z}_t$ given only the snapshot $\mathbf{X}_t$ at time t. $f(Z|\mathbf{X}_t)$ does not have to be normalized. We justify assumption (1) and describe algorithms for realizing assumption (2) in Section 3.3. As we will see in Equation (17), the Poisson noise model (Equation (1)) ensures that $f(Z|\mathbf{X}_t)$ exists and takes a simple form.

Given a prior probability distribution P_0, our goal is to compute the posterior distribution P_t for all t. The tracking algorithm starts with $t = 0$ and repeat the following procedure (Figure 7b).

$$\begin{aligned}
&\text{1. Compute the predictive distribution } P(\mathbf{Z}_{t+1}|\mathbf{Z}_t) \text{ from } P_t\\
&\text{2. Draw } K \text{ samples } Z'_s \text{ from } P(\mathbf{Z}_{t+1}|\mathbf{Z}_t)\\
&\text{3. Observe } \mathbf{X}_{t+1} \text{ and compute } W_s = f(Z'_s|\mathbf{X}_{t+1}) \qquad (8)\\
&\text{4. Approximate } P_{t+1}(Z') \text{ as using samples } Z'_s \text{ weighted by } W_s\\
&\text{5. Increase } t = t+1
\end{aligned}$$

Under the Gaussian assumption for P_t, both steps 1 and 4 may be computed in close-form (Section 3.3). This is in sharp contrast to regular particle filters, which do not assume any parametric form for P_t and accomplish steps 1 and 4 using samples. Empirically we found that the Gaussian assumption is reasonable and often leads to efficient solutions with less variability.

2.4.2. Experiments

We choose the 1D inverted pendulum problem (Figure 7a) that is standard in control theory. A pendulum is mounted via a a massless pole on a cart. The cart can move horizontally in 1D on a frictionless floor. The pendulum can rotate full circle on a fixed 2D plane perpendicular to the floor. The pendulum is released at time $t = 0$ at an unknown angle $\alpha_0 \in [0, 360^\circ)$ from the vertical line, while the cart is at an unknown horizontal offset $\beta_0 \in \mathbb{R}$. The task is to identify how the angle α_t and the offset β_t change through time from the stream of photon counts $\mathbf{X}_{1:t}$. The state of the pendulum system is $\mathbf{Z}_t = \{\alpha_t, \dot{\alpha}_t, \beta_t, \dot{\beta}_t\}$. The system's forward dynamics is well-known [17].

In our simulations, only the pole of the pendulum is white and everything else is dark. The highest photon emission rate of the scene is λ_{max} and the dark current rate is ϵ_{dc}. We systematically vary λ_{max} and ϵ_{dc} and observe the amount of estimation error in the angle α_t and cart position β_t. See Section 3.4 for the simulation procedure.

We see that (1) estimation errors decrease over time (Figure 8a,b), (2) smaller ϵ_{dc} leads to faster reduction in estimation error on average (Figure 8a,b), and (3) the tracker's *convergence time*, *i.e.*, the time it takes to achieve a certain level of estimation accuracy, decreases with illuminance (Figure 8c). The time required to satisfy high accuracy requirements (e.g., <1° for α estimation) does NOT follow a simple inverse proportional relationship with illuminance. Instead, the convergence time plateaus, potentially due to the noise in the sampling procedure (Equation (8), step 2).

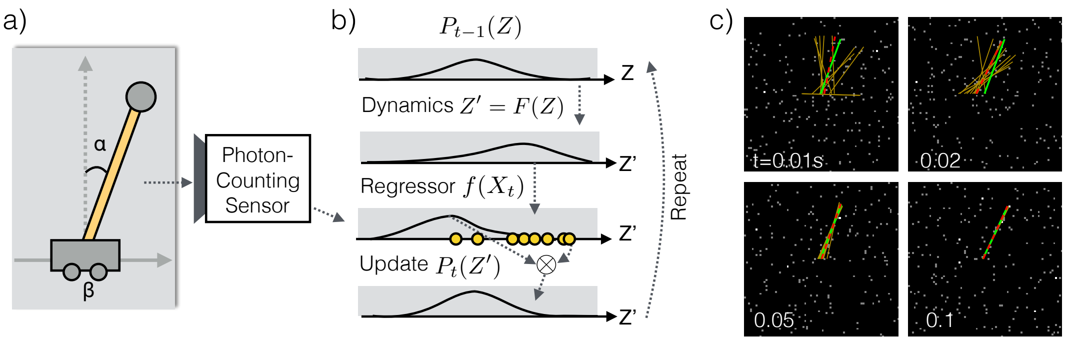

Figure 7. Low-light tracking. (**a**) An illustration of an inverted pendulum with attributes of interest α (pendulum angle) and β (cart location). The pole (yellow) is bright and everything else is dark. (**b**) Tracking algorithm (Equation (8)) iteratively updates the posterior $P_t(Z)$ using new evidence $\mathbf{X}_t$ and a low-light regressor $f(\mathbf{X}_t)$. (**c**) Snapshots of a sample run at exposure times $t = 0.01, 0.02, 0.05$ and 0.1 s. The brightest pixels emit photons at $\lambda_{max} = 10$ photons/s and the dark current $\epsilon_{dc} = 50\%$. The true position of the pendulum pole is shown in green, its estimate in red dashed, and samples from the tracker's posterior in yellow.

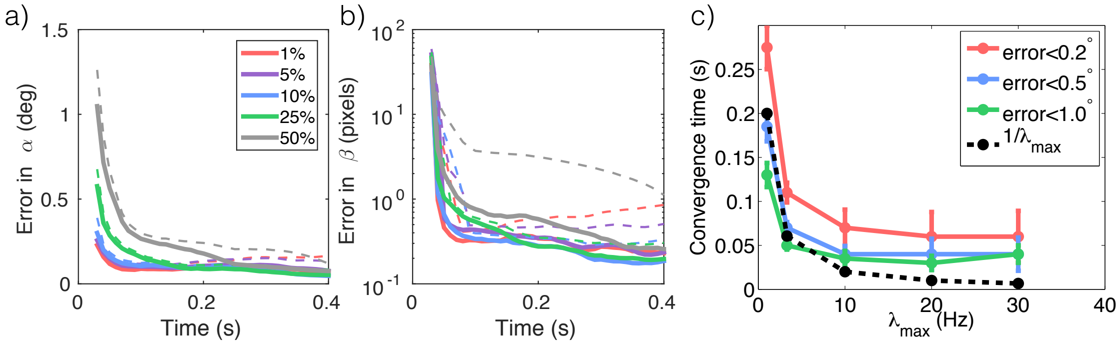

Figure 8. Tracking performance. The average estimation error in (**a**) the pendulum angle α_t and (**b**) in cart position β_t over time as a function of the amount of dark current ϵ_{dc} (color-coded). The photon emission rate is set at $\lambda_{max} = 10$ Hz. Dashed lines shows 1std above the mean. (**c**) The tracker's convergence times for its angle estimates to be within $0.2°, 0.5°$ and $1°$, respectively, of the truth as a function of illuminance. ϵ_{dc} is set at 10%. As a reference "$1/\lambda_{max}$" is inversely proportional to the illuminance and scaled to have roughly the same starting position as the "error $< 0.5°$" curve.

3. Materials and Methods

3.1. Imaging Model Including Noise Sources

Within an interval δt, the sensor readout x is corrupted by a series of noise sources.

1. The amount of photons N incident on the pixel is subject to Poisson noise (shot noise). The noise level is determined by the true intensity I and the dark current $\epsilon_{dc} \sim \mathcal{N}(0, \sigma_\epsilon)$.
2. The photon counts are corrupted by an additive Gaussian read noise $\epsilon_r \sim \mathcal{N}(0, \sigma_r)$ and a multiplicative fixed pattern noise $\epsilon_{fpn} \sim \mathcal{N}(0, \sigma_{fpn})$.

$$N \sim Poisson(\cdot|I + \epsilon_{dc}) \tag{9}$$

$$x = max(0, (N + \epsilon_r)(1 + \epsilon_{fpn})) \tag{10}$$

Sensors designed for low light applications (e.g., [5]) have promised low read noise and low fixed pattern noise. Therefore, we focus on modeling the shot noise and dark current, and assume that algorithms have access to N when the algorithms are trained. We then test the algorithm using realistic values of read noise and fixed pattern noise to study robustness against noise (see Figure 4).

3.2. Low-Light Classifier

A low-light classifier $f(\mathbf{X}_{1:t})$ that approximates the conditional distribution $P(Y|\mathbf{X}_{1:t})$ for $Y = \{1, 2, \ldots, C\}$ is developed in [9]. The classifier is a recurrent neural network consisting of multiple layers $h^{(1)}(t), \ldots, h^{(L)}(t)$ where the activation at layer l is:

$$h_j^{(l)}(t) = \sum_{\tau=1}^{t} W_j^{(l)} \mathbf{X}_\tau + b_i t \tag{11}$$

$$= h_j^{(l)}(t-1) + W_j^{(l)} \mathbf{X}_t + b_i \tag{12}$$

where $W_j^{(l)} \in \mathbb{R}^d$ and $b_j^{(l)} \in \mathbb{R}$ are the weights and the biases of the j-th unit. Equation (12) suggests that $h_j^{(l)}(t)$ may be computed incrementally from its old value $h_j^{(l)}(t-1)$.

The hidden units at layer l are organized into non-overlapping groups and pooled. A pooling unit $h_k^{(l)}$ oversees the hidden units at block G_k, and its activation is computed by:

$$m_k^{(l)}(t) = \max(0, \max_{j \in G_k}(h_j^{(l)}(t)) \tag{13}$$

Let $j^*(t) = \arg\max_{j \in G_k}(h_j^{(l)}(t))$ denote the index of the max unit at time $t-1$ ($j^*(t) = 0$ denote the event that the max value is 0). If within time interval δt only a small set $G'_k \subseteq G_k$ of hidden units within group G_k are updated, $m_k^{(l)}(t)$ may also be computed incrementally:

$$m_k^{(l)}(t) = \max_{j \in G'_k \cup j^*(t-1)} (h_j^{(l)}(t)) \tag{14}$$

Both Equations (12) and (14) are critical for an efficient implementation of the classifier. For example, if only a tenth of the units are updated in each layer, the computation time for $f(\mathbf{X}_{1:t})$ may be reduced by a factor of 10.

Finally, the output of the classifier is given by:

$$f_c(\mathbf{X}_{1:t}) = \frac{\exp(m_c^{(L)}(t))}{\sum_{c'} \exp(m_{c'}^{(L)}(t))}, \quad c \in \{1, 2, \ldots, C\} \tag{15}$$

Since $f(\mathbf{X}_{1:t})$ approximates the conditional likelihood $P(Y = c|\mathbf{X}_{1:t})$, the parameters $\{W_j^{(l)}, b_j^{(l)}\}_{j,l}$ of $f(\cdot)$ may be learned by maximum likelihood from a dataset of photon counts. However, it is expensive to keep track of a high number of photon count streams. Fortunately, Equation (11) also suggests that the network's prediction at time t only depends on the cumulative photon counts $\mathbf{S}_t \overset{\triangle}{=} \sum_{\tau=1}^{t} \mathbf{X}_\tau$. Therefore, one only needs a dataset of $\{\mathbf{S}_t, t, Y\}$ tuples to perform maximum likelihood learning.

In detail, we simulated a lowlight MNIST dataset $\{\mathbf{S}^{(i)}, PPP^{(i)}, Y^{(i)}\}_i$ where the PPPs are sampled uniformly from $\{0.22, 2.2, 22, 220\}$. Note that we are using PPP instead of the exposure time t for reasons discussed in Section 2.1. At PPP= 220 each image pixel contains around 5 bits of information (log signal-to-noise-ratio ≈ 5). Our implementation uses the MatConvNet package [18] and its default hyper-parameters for training. We train a model with the same connectivity as the LeNet [11] denoted: 784-20-50-500-10. The model contains 784 input units, followed by two convolutional hidden layers with 20 and 50 filters, respectively, of size 5 × 5. Inputs to a convolutional layer is convolved with the filters, and then pooled over 2 × 2 non-overlapping windows. After the convolutional layers are a fully connected hidden layer with 500 units and a fully connected softmax layer with 10 output categories. We minimize the negative log likelihood with a $L2$ weight decay:

$$-\sum_i \log f_{Y^{(i)}}(\mathbf{S}^{(i)}) + \eta \sum_l ||W^{(l)}||_2^2 \tag{16}$$

where $\eta = 0.0005$ is the strength of the weight decay. We use stochastic gradient descent with mini-batches of 100 and train for 60 epochs with learning rate $= 0.001$ and momentum $= 0.9$.

3.3. Tracking Algorithm

3.3.1. Low-Light Regressor

The photon count $\mathbf{X}_t$ at time t depends only on the angle α_t and cart position β_t and not their time derivatives, so our low-light regressor can only predict α_t and β_t: $f(\mathbf{Z}_t|\mathbf{X}_t) = f(\alpha_t, \beta_t|\mathbf{X}_t)$. Since we simulate the scene using a generative model, we can compute the exact form of the low-light regressor. Let $P(\mathbf{I}_t|\alpha_t, \beta_t)$ be the generative model where $\mathbf{I}_t$ is the intensity value of the image, the photon emission rate $\lambda_i(\alpha_t, \beta_t)$ for every pixel may be computed using Equation (1). As a result the log likelihood of observing $\mathbf{X}_t$ is:

$$\log P(\mathbf{X}_t|\alpha_t, \beta_t) = \log \prod_i Poisson(X_{ti}; \lambda_i(\alpha_t, \beta_t)) = Const. + \mathbf{X}_t^T \log(\lambda(\alpha_t, \beta_t)) - 1^T \lambda(\alpha_t, \beta_t) \tag{17}$$

Note that this likelihood model is linear in the intensity image λ given the parameters α_t and β_t, and *not* linear in terms of the parameters themselves. In addition, the likelihood is Poisson, not Gaussian. Hence Kalman filters are not applicable here.

Using Bayes rule we have:

$$f(\mathbf{Z}_t|\mathbf{X}_t) = P(\alpha_t, \beta_t|\mathbf{X}_t) \propto P(\mathbf{X}_t|\alpha_t, \beta_t) P(\alpha_t, \beta_t) \tag{18}$$

When a generative model is not available, the regressor may be trained discriminatively on a dataset using maximum likelihood, similar to Section 3.2.

3.3.2. Approximating the Predictive Distribution (Step 1 of Equation (8))

Let $P_t \approx \mathcal{N}(\mu_t, \Sigma_t)$ and F be the dynamics of the inverted pendulum. The predictive distribution may be approximated as a gaussian by linearizing F:

$$\mathbf{Z}_{t+1}|\mathbf{Z}_t \approx \mathcal{N}(F(\mu_t), (\nabla F|_{\mu_t})\Sigma(\nabla F|_{\mu_t})^T) \tag{19}$$

where $\nabla F|_{\mu_t}$ is the Jacobian of the dynamics F evaluated at the prior mean μ_t.

3.3.3. Approximating the Posterior Distribution (Step 4 of Equation (8))

Given weighted samples $\{Z_s'\}_{s=1}^K$ and normalized weights $\{W_s\}_{s=1}^K$ (*i.e.*, $\sum_s W_s = 1$), the posterior P_{t+1} may be approximated by a Gaussian with: $P_{t+1} \approx \mathcal{N}(\mu_{t+1}, \Sigma_{t+1})$, where

$$\mu_{t+1} = \sum_s W_s Z'_s \tag{20}$$

$$\Sigma_{t+1} = \frac{\sum_s W_s (Z'_s - \mu_{t+1})(Z'_s - \mu_{t+1})^T}{1 - \sum_s W_s^2} \tag{21}$$

3.4. *Tracking Experiment*

The pole of the pendulum has an intensity of 1 while everything else (background, cart, *etc*) has intensity 0. The pole is 3 × 30 pixels and the entire scene is 80 × 80 pixels. The pendulum is half the mass of the cart. The maximum photon emission rate is λ_{max}, and dark current is ϵ_{dc}. The pendulum is released at the origin ($\beta_0 = 0$) with α_0 randomly chosen from $\{10°, 20°, 30°, 40°\}$. Each initial state is simulated 25 times and then aggregated, yielding a total of 100 trials for each condition (*i.e.*, for each pair of λ_{max} and ϵ_{dc}). The distribution on α_0 and β_0 used for simulation is not available to the tracking algorithm, which instead assumes uniform distributions for both.

3.4.1. Internal Noise Due to Sampling

For Figure 7a,b we used $K = 1000$ samples from the predictive distribution (Equation (8), step 2). Repeating the experiment using $K = 300$ samples shows the same trend with larger error bars. One counterintuitive finding is that as the exposure time increases, the estimation error first decreases and then diverges (both higher mean and deviation are visible, Figure 7a,b). The degree of divergence is aggravated by increasing signal-to-noise ratio (or reducing dark current ϵ_{dc}). This may be explained by the internal noise in the tracking algorithm. The algorithm relies on samples from the predictive distribution $P(\mathbf{Z}_{t+1}|\mathbf{Z}_t)$ (Equation (8), step 1) to coincide with states that have a high observation likelihood $f(Z|\mathbf{X}_t)$ (Equation (8), step 3). This coincidence is less likely to happen when the observation likelihood becomes precise, or as the posterior becomes sharply-peaked. Similarly, for Figure 7c we used $K = 1000$, and repeated the experiment with $K = 300$. They obtain exactly the same trends except that smaller K corresponds to a higher plateau for the convergence time (Figure 7c), indicating that sampling noise may be limiting the speed for accurate tracking.

4. Discussion and Conclusions

The advent of photon-counting sensors motivates us to reconsider the prevalent paradigm in computer vision: Rather than first capturing an image and then analyzing it, we should design algorithms that incrementally compute information from the stream of photons that hits the sensor, without any attempt to reconstruct the image. This style of thinking is particularly attractive in low light conditions, where the exposure time required for capturing a high-quality image is prohibitively lengthy.

Photon-counting sensors deliver small increments of the image at short delays and high frequencies. We show that this incremental input could in principle be applied to solve a variety of vision problems with a short exposure time. Algorithms that are inspired by the asymptotically optimal SPRT appear particularly well suited for minimizing photon counts while satisfying a desired accuracy bound.

Our first finding is that useful information may be computed in a short amount of time, well ahead of the integration time that is required for forming (or reconstructing) a high quality image. In Figure 1b,c, we see that a low-light classification algorithm can achieve 1% classification error of handwritten digits before one photon per pixel has been collected. The low-light classifier may be viewed as reconstructing the features (instead of the image), and carrying the uncertainty of the features all the way to classification. This uncertainty is essential in a sequential decision making setting to determine when to stop collecting more photons. In comparison, conventional approaches simply reconstruct the image, and pass it to a classifier trained on high-quality images. The conventional approach suffers from two issues. (1) Since the conventional approach discards the uncertainty information, it is not clear how to determine the required exposure time; and (2) statistics of the reconstruction may be different from that of high-quality images, hence the classifier's performance may not be guaranteed.

Second, algorithms for classification and search from streams of photons are photon-efficient: they stop as soon as a confident decision is made. This efficiency is critical for domains such as astrophysics where each photon is precious [19], and cell imaging applications where the dies that are employed to visualize cell structures are phototoxic [20,21]. As an example of the photon efficiency, Figure 3a shows that at PPP = 1 the low-light classifier based on SPRT can already achieve a better performance than a classifier using PPP = 220. Additionally, contrary to the conventional paradigm that obtains images with a fixed duration, low-light classifiers and search algorithms uses different exposure times depending on the specific photon arrival sequence (Figure 3b) and on the overall classification difficulty of the example (Figure 3b,c).

Third, algorithms become faster when more light is present. For classification and search where the input image is stationary, time is synonymous with the amount of photons. Higher illuminance therefore translates to faster decisions. This simple relationship is useful in that a low-light system trained for classification or search at one illuminance level may be easily applied at another illuminance level. The transition only requires knowing the illuminance level of the new scene, which may be estimated either via an explicit illuminance sensor or from the total photon count across the image [22]. In addition, the ER *vs.* median PPP tradeoff (Figures 3a and 6a–c) is an illuminance-independent characteristic of the algorithm and the task.

Last, the relationship between illuminance and speed is not always simple in tracking. The dynamics governing the object movement/state transition has its own time scale. A regressor $f(\mathbf{X}_t)$ thus has only a finite duration for integrating information before the object moves too far. A tracker relies on accurate prediction of the regressor to postulate the object's next position. An inaccurate prediction due to short exposure time may cause tracking failure. In addition, internal noise in the tracking algorithm (Section 3.4) may cause the speed to plateau after a certain illuminance level. As a result, the relationship between illuminance and convergence time is not a simple inversely proportional relationship, as shown in Figure 8c.

Although we have provided proof-of-concept illustrations of low-light vision applications with photon-counting sensors, many challenges still remain. (1) We are not aware of any hardware specialized at processing streams of photon counts at high speeds. Nonetheless, current Field-programmable Gate Array (FPGA) implementations have achieved over 2000 Hz throughput for classifying images of a similar resolution as those in Section 2.2 [23]. In addition, the low-light classifiers implemented as a recurrent neural network (Section 2.2) can be updated incrementally, *i.e.*, $f(\mathbf{X}_{1:t})$ can be computed from the internal states of $f(\mathbf{X}_{1:t-1})$ and $\mathbf{X}_t$ with sparse updates. The sparseness may be key to expedite computation. (2) We do not yet have datasets collected directly from photon-counting sensors to verify the robustness of the proposed methodology, as many such sensors are still in the making [5–7]. (3) Our noise model (Section 3.1) may be a crude approximation to handle moving objects. For example, we do not model motion induced blur or input disturbances due to camera self-motion. Nonetheless, motion induced blur may not be an issue if the sensor is collecting a single photon at a time, such as in low light and/or high-frequency imaging scenarios. In these scenarios even though the amount of photons is so low that full image reconstruction is difficult, our algorithm can still make correct and full use of the where and when of photon arrivals. This is precisely the advantage of image-free vision.

In conclusion, we propose to integrate computer vision with photon-counting sensors to address the challenges facing low-light vision applications. We should no longer wait for a high-quality image to be formed before executing the algorithm. Novel algorithms and hardware solutions should be developed to operate on streams of photon-counts. These solutions should also sidestep image reconstruction and focus directly on the task at hand.

Acknowledgments: This work was funded by Office of Naval Research (ONR) N00014-10-1-0933 and Gordon and Betty Moore Foundation (Palo Alto , CA, USA).

Author Contributions: B. Chen and P. Perona conceived and designed the experiments. B. Chen performed the experiments and analyzed the data. B. Chen and P. Perona wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CPU	Central Processing Unit
ER	Error rates
FPGA	Field-programmable Gate Arrays
MNIST	The Mixed National Institute of Standards and Technology dataset
PPP	The number of photons per pixel in a low-light image, averaged across all locations and the imaging duration
SPRT	Sequential probability ratio test

References

1. Hall, E.; Brenner, D. Cancer risks from diagnostic radiology. *Cancer* **2014**, *81*, doi:10.1259/bjr/01948454.
2. Stephens, D.J.; Allan, V.J. Light microscopy techniques for live cell imaging. *Science* **2003**, *300*, 82–86.
3. Brida, G.; Genovese, M.; Berchera, I.R. Experimental realization of sub-shot-noise quantum imaging. *Nat. Photonics* **2010**, *4*, 227–230.
4. Zappa, F.; Tisa, S.; Tosi, A.; Cova, S. Principles and features of single-photon avalanche diode arrays. *Sens. Actuators A Phys.* **2007**, *140*, 103–112.
5. Fossum, E. The quanta image sensor (QIS): Concepts and challenges. In Proceedings of the Computational Optical Sensing and Imaging 2011, Toronto, ON, Canada, 10–14 July 2011; doi:10.1364/COSI.2011.JTuE1.
6. Fossum, E.R. Multi-Bit Quanta Image Sensors. In Proceedings of the International Image Sensor Workshop, Vaals, The Netherlands, 8–12 June 2015; pp. 292–295.
7. Sbaiz, L.; Yang, F.; Charbon, E.; Süsstrunk, S.; Vetterli, M. The gigavision camera. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 1093–1096.
8. Morris, P.A.; Aspden, R.S.; Bell, J.E.; Boyd, R.W.; Padgett, M.J. Imaging with a small number of photons. *Nat. Commun.* **2015**, *6*, do:10.1038/ncomms6913.
9. Chen, B.; Perona, P. Scotopic Visual Recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Santiago, Chile, 7–13 December 2015; pp. 8–11.
10. Abu-Naser, A.; Galatsanos, N.P.; Wernick, M.N. Methods to detect objects in photon-limited images. *JOSA A* **2006**, *23*, 272–278.
11. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *IEEE Proc.* **1998**, *86*, 2278–2324.
12. Wald, A. Sequential tests of statistical hypotheses. *Ann. Math. Stat.* **1945**, *16*, 117–186.
13. Elman, J.L. Distributed representations, simple recurrent networks, and grammatical structure. *Mach. Learn.* **1991**, *7*, 195–225.
14. Chen, B.; Perona, P. Speed versus accuracy in visual search: Optimal performance and neural architecture. *J. Vis.* **2015**, *15*, 9–9.
15. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Fluids Eng.* **1960**, *82*, 35–45.
16. Pitt, M.K.; Shephard, N. Filtering via simulation: Auxiliary particle filters. *J. Amer. Stat. Assoc.* **1999**, *94*, 590–599.
17. Liberzon, D. *Switching in Systems and Control*; Springer Science & Business Media: New York, NY, USA, 2012.
18. Vedaldi, A.; Lenc, K. MatConvNet—Convolutional Neural Networks for MATLAB. Available online: http://arxiv.org/abs/1412.4564 (accessed on 4 April 2016).
19. Martin, D.C.; Chang, D.; Matuszewski, M.; Morrissey, P.; Rahman, S.; Moore, A.; Steidel, C.C. Intergalactic medium emission observations with the Cosmic Web Imager. I. The circum-QSO medium of QSO 1549+19, and evidence for a filamentary gas inflow. *Astrophys. J.* **2014**, *786*, 106–106.

20. Hoebe, R.; Van Oven, C.; Gadella, T.W.; Dhonukshe, P.; Van Noorden, C.; Manders, E. Controlled light-exposure microscopy reduces photobleaching and phototoxicity in fluorescence live-cell imaging. *Nat. Biotechnol.* **2007**, *25*, 249–253.
21. Ji, N.; Magee, J.C.; Betzig, E. High-speed, low-photodamage nonlinear imaging using passive pulse splitters. *Nat. Methods* **2008**, *5*, 197–202.
22. Cheng, D.; Price, B.; Cohen, S.; Brown, M.S. Effective Learning-Based Illuminant Estimation Using Simple Features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1000–1008.
23. Ovtcharov, K.; Ruwase, O.; Kim, J.Y.; Fowers, J.; Strauss, K.; Chung, E.S. Accelerating Deep Convolutional Neural Networks Using Specialized Hardware, Microsoft Research Whitepaper. Available online: http://research.microsoft.com/apps/pubs/?id=240715 (accessed on 4 April 2016).