SoDaCam: Software-defined Cameras via Single-Photon Imaging

Varun Sundar[†] vsundar4@cs.wisc.edu
Andrei Ardelean[‡] a.ardelean@epfl.ch
Tristan Swedish[§] tristan@ubicept.com
Claudio Bruschini[‡] Edoardo Charbon[‡] {claudio.bruschini, edoardo.charbon}@epfl.ch
Mohit Gupta[†,§] mohitg@cs.wisc.edu

[†]University of Wisconsin-Madison [‡]École polytechnique fédérale de Lausanne [§]Ubicept

Abstract

Reinterpretable cameras are defined by their post-processing capabilities that exceed traditional imaging. We present "SoDaCam" that provides reinterpretable cameras at the granularity of photons, from photon-cubes acquired by single-photon devices. Photon-cubes represent the spatio-temporal detections of photons as a sequence of binary frames, at frame-rates as high as 100 kHz. We show that simple transformations of the photon-cube, or photon-cube projections, provide the functionality of numerous imaging systems including: exposure bracketing, flutter shutter cameras, video compressive systems, event cameras, and even cameras that move during exposure. Our photon-cube projections offer the flexibility of being software-defined constructs that are only limited by what is computable, and shot-noise. We exploit this flexibility to provide new capabilities for the emulated cameras. As an added benefit, our projections provide camera-dependent compression of photon-cubes, which we demonstrate using an implementation of our projections on a novel compute architecture that is designed for single-photon imaging.

1. Introduction

Throughout the history of imaging, sensing technologies and the corresponding processing have developed hand-in-hand. In fact, sensing technologies have, to some extent, defined the scope of processing captured data. In the film era, instances of such processing included dodging and burning. The advent of digital cameras provided processing at the granularity of pixels and paved the way for modern computer vision. Light field cameras [34, 78], by sampling the plenoptic function [2], allowed post-capture processing at the granularity of light rays, enabling novel functionalities such as refocusing photos after-capture. The logical limit of post-capture processing, given the fundamental quantization of light, would be at the level of individual photons. What would imaging look like if we could perform computational processing on individual photons?

In this work, we show that photon data captured by a new class of single-photon detectors, called single-photon avalanche diodes (SPADs), makes it possible to emulate a wide range of imaging modalities such as exposure bracketing [12], video compressive systems [38, 55] and event cameras [52, 60]. A user then has the flexibility to choose one (or even multiple) of these functionalities *post-capture* (Fig. 1 *(top)*). SPAD arrays can operate as extremely high frame-rate photon detectors (∼100 kHz), producing a temporal sequence of binary frames called a photon-cube [16]. We show that computing *photon-cube projections*, which are simple linear and shift operations, can reinterpret the photon-cube to achieve novel post-capture imaging functionalities in a software-defined manner (Fig. 1 *(middle)*).

As case studies, we emulate three distinct imagers: high-speed video compressive imaging; event cameras which respond to dynamic scene content; and motion projections which emulate sensor motion, without any real camera movement. Fig. 1 *(bottom)* shows the outputs of these cameras that are derived from the same photon-cube.

Computing photon-cube projections. One way to obtain photon-cube projections is to read the entire photon-cube off the SPAD array and then perform relevant computations off-chip; we adopt this strategy for our experiments in Secs. 6.1 and 6.2. While reasonable for certain applications, reading out photon-cubes requires an exorbitant data-bandwidth, which can be up to 100 Gbps for a 1 MPixel array—well beyond the capacity of existing data peripherals. Such readout considerations will become center stage as large-format SPAD arrays are fabricated [48, 49].

An alternative is to avoid transferring the entire photon-

[*] Corresponding author: Varun Sundar. This research was supported in parts by NSF CAREER award 1943149, NSF award CNS-2107060, and the Swiss National Science Foundation grant 200021_166289. We also thank Paul Mos for providing access to SwissSPAD2 acquisition software.

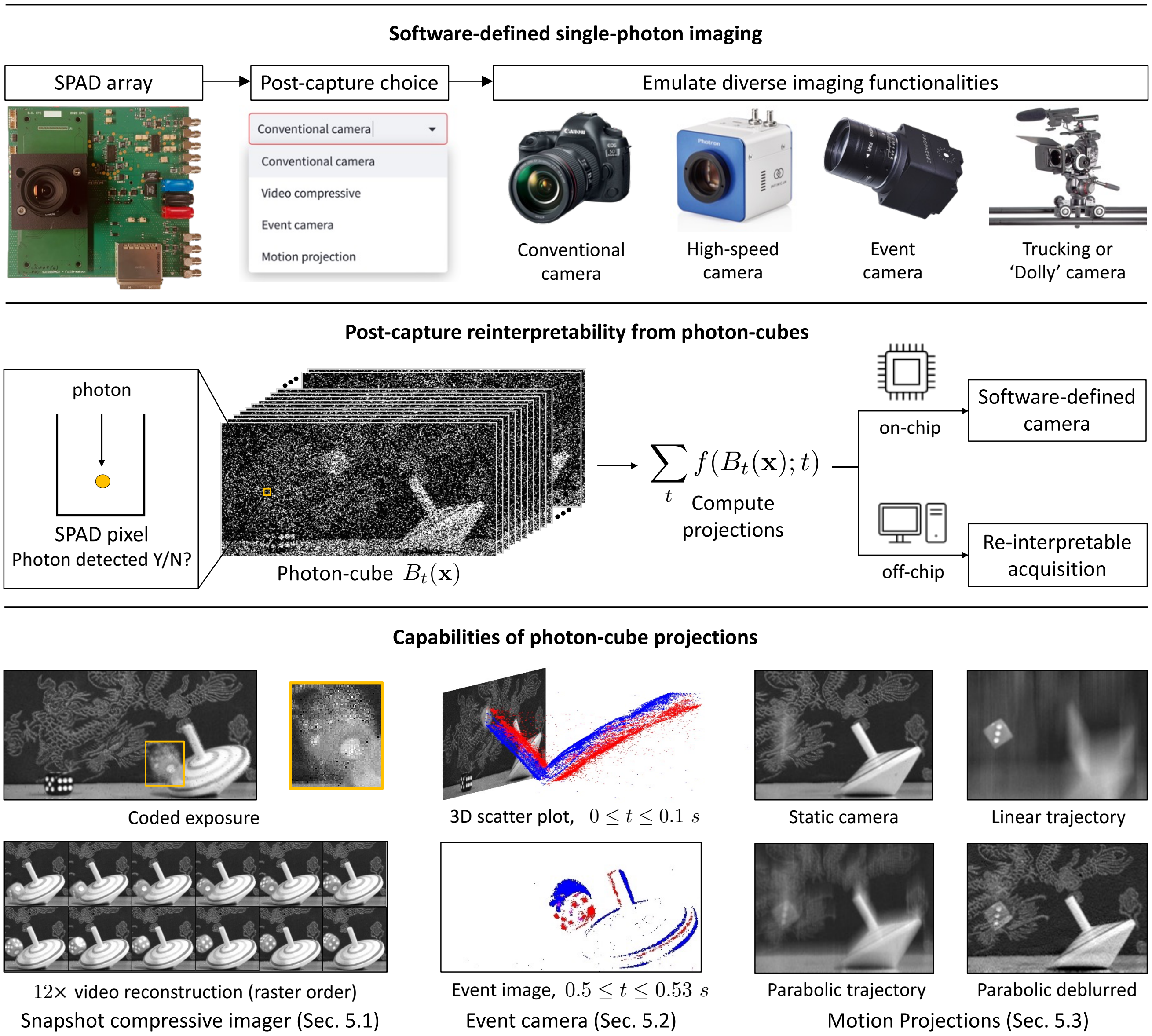

Software-defined single-photon imaging
SPAD array
Post-capture choice
Emulate diverse imaging functionalities
Conventional camera
Conventional camera
Video compressive
Event camera
Motion projection
Conventional camera
High-speed camera
Event camera
Trucking or 'Dolly' camera
Post-capture reinterpretability from photon-cubes
photon
SPAD pixel
Photon detected Y/N?
Photon-cube $B_t(\mathbf{x})$
$\sum_t f(B_t(\mathbf{x}); t)$
Compute projections
on-chip
Software-defined camera
off-chip
Re-interpretable acquisition
Capabilities of photon-cube projections
Coded exposure
3D scatter plot, $0 \leq t \leq 0.1\ s$
Static camera
Linear trajectory
12× video reconstruction (raster order)
Event image, $0.5 \leq t \leq 0.53\ s$
Parabolic trajectory
Parabolic deblurred
Snapshot compressive imager (Sec. 5.1)
Event camera (Sec. 5.2)
Motion Projections (Sec. 5.3)

Figure 1: *(top)* **SoDaCam** can emulate a variety of cameras from the photon-cubes acquired by single-photon devices. *(middle)* Photon-cubes represent the spatio-temporal detection of photons as a sequence of binary frames. Projections of the photon-cube, when computed either on or off-chip, result in reinterpretable and software-defined cameras. We demonstrate the versatility of photon-cube projections on a **real dynamic scene**: a die falls on a table, bounces, spins in the air and later ricochets off a nearby toy top. *(bottom)* The cameras emulated by our photon-cube projections can produce: a 12× high-speed video from a single compressive snapshot, event-stream representations of two time intervals (blue and red depict positive and negative spikes respectively), an image where the die appears stationary, as well as a motion-deblurred image.

cube by computing projections near sensor. As a proof-of-concept, we implement photon-cube projections on UltraPhase [5], a recently-developed programmable SPAD imager with independent processing cores that have dedicated RAM and instruction memory. We show, in Sec. 6.3, that computing projections on-chip greatly reduces sensor readout and, as a consequence, power consumption.

Implications: Toward a photon-level software-defined camera. The photon-cube projections introduced in this paper are computational constructs that provide a realization of *software-defined cameras* or *SoDaCam*. Being software-defined, SoDaCam can emulate multiple cameras simultaneously without additional hardware complexity. SoDaCam, by going beyond baked in hardware choices, unlocks hitherto unseen capabilities—such as 2000 FPS video from 25 Hz readout (Fig. 7); event imaging in very low-light conditions (Fig. 9); and motion stacks, which are a stack of images where in each image, objects only in certain velocity ranges appear sharp (Fig. 6).

Limitations. The SPAD array [67] used in this work has a relatively low spatial resolution (512 × 256), and a low fill-factor (∼10%) owing to the lack of microlenses in the prototype used. Similarly, the near-sensor processor that we use has limited capabilities compared to off-chip processors. However, with rapid progress in the development of single-photon cameras [48, 49] and increasing interest in near-sensor processors, we anticipate that many of these shortcomings will be addressed in the upcoming years.

2. Related Work

Reinterpretable imaging has previously been explored at the granularity of light rays [3], by modulating the plenoptic function, and at the level of spatio-temporal voxels [20], by using fast per-pixel shutters. SoDaCam represents a logical culmination of reinterpretability at the level of photon detections, that facilitates multiple post-capture imaging functionalities.

Programmable imaging using a digital micromirror device was first introduced in Nayar et al. [51] to perform *pre-capture* radiometric manipulations. Modern programmable cameras are typically near-sensor processors [8, 69, 71] that can perform limited operations in analog [46, 55, 69], while more complex operations [9, 45] occur after analog-to-digital conversion (ADC). In contrast, by performing *post-capture* computations directly on photon detections, we can perform complex operations without incurring the read-noise penalty that is associated with ADC.

Passive single-photon imaging. Only recently have SPADs been utilized as passive imaging devices, with applications in high-dynamic range imaging [29, 30, 37, 50], motion-compensation [31, 58], burst photography [41, 42] and object tracking [22]. Compared to compute-intensive burst-photography methods [41], our proposed techniques involve lightweight computations that can be performed near sensor. These computations can also be performed using other single-photon imagers such as Jots [15, 17], which feature higher sensor resolution and photon-efficiency [40], albeit at lower frame-rates and higher read noise.

Reducing the readout of SPADs. Several data reduction strategies have been proposed in the context of SPADs that are used to timestamp incident photons, including: coarse histograms [13, 23, 56], compressive histograms [21], and measuring differential time-of-arrivals [70, 77]. When SPADs are operated as photon-detectors, multi-bit counting [48], or summing binary frames, can reduce readout. While compression is not our main objective, we show that photon-cube projections act as camera-specific compression schemes that dramatically reduce sensor readout.

3. Background: Single-Photon Imaging Model

A SPAD array captures incident light as a *photon-cube*: a temporal sequence of binary frames that represents the pixel-wise detection of photons across their respective exposure windows. We can model the stochastic arrival of photons as a Poisson process [72], allowing us to treat spatio-temporal values of the photon-cube as independent Bernoulli random variables with

$$\Pr\left\{B_t(\mathbf{x}) = 1\right\} = 1 - e^{-(\eta\Phi(\mathbf{x},t)+r_q)w_{\text{exp}}}, \tag{1}$$

where $B_t(\mathbf{x})$ represents the value of the photon-cube at pixel $\mathbf{x}$ and exposure index $1 \leq t \leq T$, which receives a mean incident flux of intensity $\Phi(\mathbf{x}, t)$ across its exposure of duration w_{exp}. Additionally, η is the photon detection efficiency of the SPAD, and r_q denotes the sensor's dark count rate—which is the rate of spurious counts unrelated to incident photons. While individual binary frames are extremely noisy, the temporal sum of the photon-cube

$$\mathcal{I}_{\text{sum}}(\mathbf{x}) := \sum_{t=1}^{T} B_t(\mathbf{x}), \tag{2}$$

can produce an 'image' of the scene that is sharp in static regions, but blurry in dynamic regions (Fig. 2 *(top)*). Indeed, in static regions, the sum-image can be used to derive a maximum likelihood estimator of the scene intensity [4], given by $\hat{\Phi}(\mathbf{x}) = -\ln(1 - T^{-1}\mathcal{I}_{\text{sum}}(\mathbf{x}))/\eta w_{\text{exp}} - r_q/w_{\text{exp}}$.

4. Projections of the Photon-Cube

The temporal sum described in Eq. (2) is a simple instance of projections of a photon-cube. Our key observation is that it is possible to compute a wide range of photon-cube projections, each of which emulates a unique sensing modality *post-capture*—including modalities that are difficult to achieve with conventional cameras. For example, varying the number of bit-planes that are summed over emulates exposure bracketing [12, 43], which is typically used for HDR imaging. Compared to conventional exposure bracketing, the emulated exposure stack, being software-defined, does not require spatial and temporal registration, which can often be error-prone. Fig. 2 *(top)* shows an example of an exposure stack computed from a photon-cube.

Going further, we can gradually increase the complexity of the projections. For example, consider a *coded exposure*

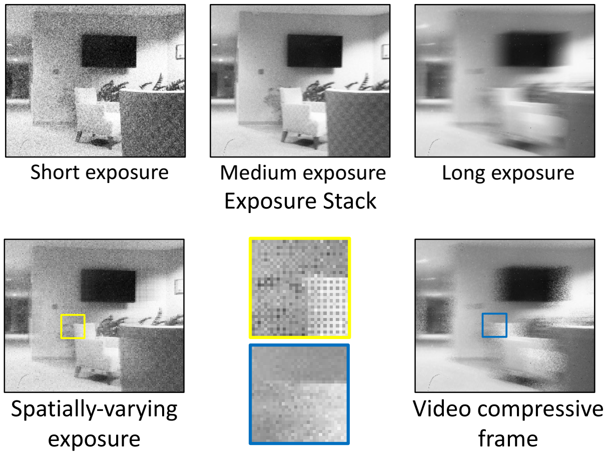


Figure 2: **Coded exposures from photon-cubes.** *(top)* An exposure stack with sum-images computed using 250, 500, and 1000 bit-planes. Short exposures are noisy while long exposures exhibit motion-blur. *(bottom)* Spatially-varying exposure that uses a quad pattern [32] (see inset), and a video compressive frame that uses 16 random binary masks to modulate the photon-cube. Zoom-in to see details.

projection that multiplexes bit-planes with a temporal code

$$\mathcal{I}_{\text{flutter}}(\mathbf{x}) := \sum_{t=1}^{T} C_t B_t(\mathbf{x}), \tag{3}$$

where C_t is the temporal code. An example of globally-coded exposures is the flutter shutter camera [53], which uses pseudo-random binary codes for motion-deblurring.

More general coded exposures can be obtained via spatially-varying temporal coding patterns $C_t(\mathbf{x})$:

$$\mathcal{I}_{\text{coded}}(\mathbf{x}) := \sum_{t=1}^{T} C_t(\mathbf{x}) B_t(\mathbf{x}). \tag{4}$$

Fig. 2 *(bottom)* shows spatially-varying exposures that use a quad (Bayer-like) spatial pattern and random binary masks. With photon-cubes, we can perform spatially-varying coding without bulky spatial light modulators, similar to focal-plane sensor-processors [46, 69]. Moreover, we can capture multiple coded exposures simultaneously, which is challenging to realize in existing sensors. In Sec. 5.1, we describe coding patterns for video compressive sensing.

Spatial and temporal gradients form the building blocks of several computer vision algorithms [7, 11, 24, 26, 39]. Given this, another projection of interest is temporal contrast, i.e., a derivative filter preceded by a smoothing filter:

$$\mathcal{I}_{\text{contrast}}(\mathbf{x}, t) := D_t \circ G * B_t(\mathbf{x}), \tag{5}$$

where D_t is the difference operator, G could be exponential or Gaussian smoothing, $\circ$ denotes function composition, and $*$ denotes convolution. Due to their sparse nature, gradients form the basis of bandwidth- and power-efficient event cameras [10, 14, 36, 60], which we emulate in Sec. 5.2.

So far, we have considered projections taken only along the time axis. Next, we consider a more general class of *spatio-temporal projections* that lead to novel functionalities. For instance, computing a simple projection, such as the temporal sum, along arbitrary spatio-temporal directions emulates sensor motion during exposure time [33], but *without moving the sensor*. We achieve this by shifting bit-planes and computing their sum:

$$\mathcal{I}_{\text{shift}}(\mathbf{x}) := \sum_{t=1}^{T} B_t\left(\mathbf{x} + \mathbf{r}(t)\right), \tag{6}$$

where $\mathbf{r}$ is a discretized 2D trajectory that determines sensor motion. Outside a software-defined framework, such projections are hard to realize without physical actuators. We describe the capabilities of *motion projections* in Sec. 5.3.

In summary, the proposed photon-cube projections are simple linear and shift operators that lead to a diverse set of post-capture imaging functionalities. These projections pave the way for future 'swiss-army-knife' imaging systems that achieve *multiple functionalities (e.g., event cameras, high-speed cameras, conventional cameras, HDR cameras) simultaneously with a single sensor*. Finally, these projections can be computed efficiently in an online manner, which makes on-chip implementation viable (Sec. 6.3).

At this point, we note that a key enabling factor of photon-cube projections is the extremely high temporal-sampling rate of SPADs. Indeed, the temporal sampling rate determines key aspects of sensor emulation, such as the discretization of temporal derivatives and motion trajectories. This raises a natural question: can we use conventional high-speed cameras for computing projections?

Trade-off between frame-rate and SNR. In principle, photon-cube projections can be computed using regular (CMOS or CCD based) high-speed cameras. Unfortunately, each frame captured by a high-speed camera incurs a read-noise penalty, which increases with the camera's frame-rate [6]. In fact, the read noise levels of high-speed cameras [1] can be 10–30× higher than consumer cameras [28]. Coupled with the low per-frame incident flux at high frame-rates, high levels of read noise result in extremely low SNRs. In contrast, SPADs do not incur a per-frame read noise and are limited only by the fundamental photon noise. Hence, for the post-capture software-defined functionalities proposed here, it is imperative to use SPADs.

5. Emulating Cameras from Photon-Cubes

Sec. 4 presented the concept of photon-cube projections, and its potential for achieving multiple post-capture imag-

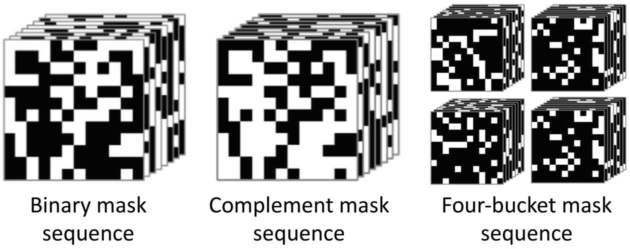


Figure 3: **Modulating masks for video compressive sensing.** *(left)* A single VCS measurement temporally compresses a sequence of frames using binary random masks. *(center)* Two-bucket cameras capture an additional measurement by using the complementary mask sequence. *(right)* We propose using multi-bucket captures by randomly choosing an active bucket for each frame. Both two-bucket and multi-bucket captures have 100% light efficiency. All masks are visualized here for 16×16 pixels.

ing functionalities. As case studies, we now demonstrate three imaging modalities: video compressive sensing, event cameras, and motion-projection cameras. These modalities have been well-studied over several years; in particular, there exist active research communities around video compressive sensing and event cameras today. We also show new variants of these imaging systems that arise from the software-defined nature of photon-cube projections.

5.1. Video Compressive Sensing

Video compressive systems *optically* multiplex light with random binary masks, such as the patterns in Fig. 3 *(left)*. As discussed in the previous section, such multiplexing can be achieved *computationally* using photon-cubes.

Two-bucket cameras. One drawback of capturing coded measurements is the light loss due to blocking of incident light. To prevent loss of light, coded two-bucket cameras [69] capture an additional measurement that is modulated by the complementary mask sequence (Fig. 3 *(center)*). Such measurements recover higher quality frames, even after accounting for the extra readout [35, 61]. Two-bucket captures can be readily derived from photon-cubes, by implementing Eq. (4) with the additional mask sequence.

Multi-bucket cameras. We can extend the idea of two-bucket captures to multi-bucket captures by accumulating bit-planes in one of k buckets that is randomly chosen at each time instant and pixel location. Since multiplexing is performed computationally, we do not face any loss in photoreceptive area that [59, 68] hampers existing multi-bucket sensors. Multi-bucket captures can reconstruct a large number of frames by better conditioning video recovery and provide extreme high-speed video imaging. Fig. 3 *(right)* shows the modulating masks for a four-bucket capture.

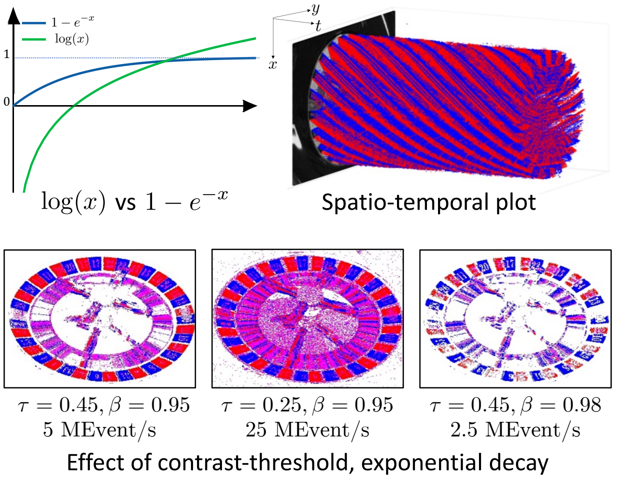


Figure 4: **Event stream from photon-cubes.** *(top left)* By exploiting the non-linear response curve of SPADs to encode brightness, we can avoid the underflow issues of a log-response. We visualize events generated from photon-cubes using a 3D scatter plot of polarities (*top right*, 14000 bit-planes), and frame accumulation of events (*bottom*, 1200 bit-planes). Blue and red denote positive and negative spikes respectively. The event images also show the effect of varying the contrast threshold τ and exponential decay β—larger values yield a less noisy but sparser event stream.

5.2. Event Cameras

Next, we describe emulation of event-cameras, which capture changes in light intensity and are conceptually similar to the temporal contrast projection introduced in Eq. (5). Physical implementations of event sensors [10, 14, 36, 60] generate a photoreceptor voltage $V(\mathbf{x}, t)$ with a logarithmic response to incident flux $\Phi(\mathbf{x}, t)$, and output an event $(\mathbf{x}, t, p)$ when this voltage deviates sufficiently from a reference voltage $V_{\text{ref}}(\mathbf{x})$:

$$|V(\mathbf{x}, t) - V_{\text{ref}}(\mathbf{x})| > \tau, \tag{7}$$

where τ is called the contrast-threshold and $p = \text{sign}(V(\mathbf{x}, t) - V_{\text{ref}}(\mathbf{x}))$ encodes the polarity of the event. Once an event is generated, $V_{\text{ref}}(\mathbf{x})$ is updated to $V(\mathbf{x}, t)$. Eq. (7), for a smoothly-varying flux intensity, thresholds a function of the temporal gradient, i.e., $\partial_t \log(\Phi(\mathbf{x}, t))$.

From bit-planes to event streams. To produce events from SPAD frames, we compute an exponential moving average (EMA) of the bit-planes, as $\mu_t(\mathbf{x}) = (1 - \beta) B_t(\mathbf{x}) + \beta \mu_{t-1}(\mathbf{x})$—where $\mu_t(\mathbf{x})$ is the EMA, β is the smoothing

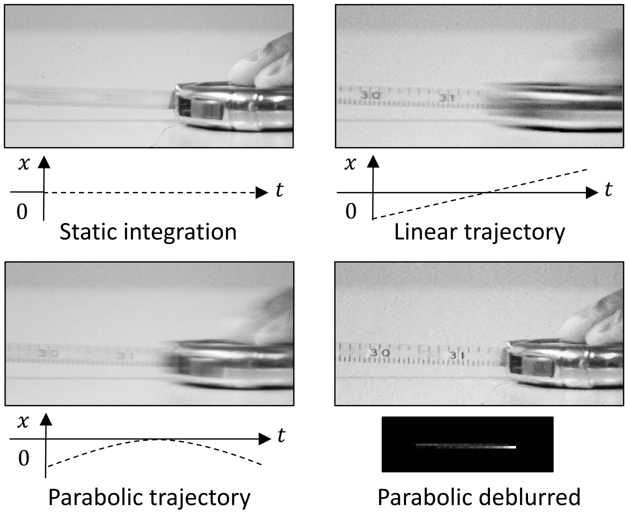


Figure 5: **Motion projections.** *(top)* Integrating along a linear trajectory in the photon-cube changes the apparent image-space velocity of scene objects. Details are seen for *(top left)* the case when static, and *(top right)* the metallic tape when the sensor translates along the x-axis. *(bottom)* A parabolic integration trajectory results in a motion-invariant image, resulting in similar blur kernels for all objects. *(bottom right)* Deblurring with the resultant shift-invariant point spread function (shown in *inset*) produces a sharp image.

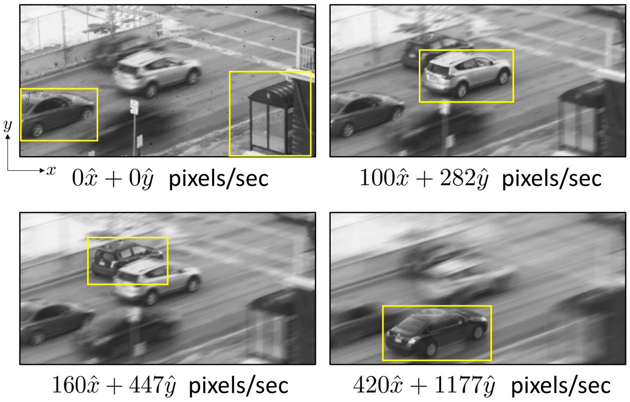

y
x
$0\hat{x} + 0\hat{y}$ pixels/sec
$100\hat{x} + 282\hat{y}$ pixels/sec
$160\hat{x} + 447\hat{y}$ pixels/sec
$420\hat{x} + 1177\hat{y}$ pixels/sec

Figure 6: **Motion stack.** Computing multiple linear projections with different trajectories can produce a stack of images where objects with matching velocity are sharp. Here, we show a traffic scene involving four cars that have four different velocities. By suitably altering the slope of the linear trajectory, we can produce images where only one of the cars appear sharp at a time. We indicate the slope of the trajectories chosen and the objects that are "in-focus".

factor, and B_t is a bit-plane. We generate an event when $\mu_t(\mathbf{x})$ deviates from $\mu_{\text{ref}}(\mathbf{x})$ by at least τ:

$$|h(\mu_t(\mathbf{x})) - h(\mu_{\text{ref}}(\mathbf{x}))| > \tau, \tag{8}$$

where h is a scalar function applied to the EMA. We can see that Eq. (8) thresholds temporal contrast, by observing the role played by the EMA and the difference operator.

Setting h to be the logarithm of the flux MLE mimics Eq. (7). However, since the log-scale is used to prevent sensor saturation, a simpler alternative is to use the non-saturating response curve of SPAD pixels ($h(x) = x$). The response curve takes the form of $1 - \exp(-\alpha\Phi(\mathbf{x}, t))$, where α is a flux-independent constant. As a major advantage, this response curve avoids the underflow issues of the log function that can occur in low-light scenarios [62].

The SPAD's frame-rate determines the time-stamp resolution of emulated events. In Fig. 4, we show the events generated from a photon-cube acquired at a frame-rate of 96.8 kHz—resulting in a time-stamp resolution of ~10 μs that is comparable to those of existing event cameras.

How do SPAD-events differ from the output of a regular event camera? The main difference is the expression of temporal contrast, given by $\partial_t h$, is now $-\partial_t \exp(-\alpha\Phi(\mathbf{x}, t))$, instead of $\partial_t \log(\Phi(\mathbf{x}, t))$. This difference poses no compatibility issues for a large class of event-vision algorithms that utilize a grid of events [54, 57, 66] or brightness changes [18]. We show examples of downstream applications using SPAD-events in Supplementary Note 2. Finally, SPAD-events can be easily augmented with spatially- and temporally-aligned intensity information—a synergistic combination that has been exploited by several recent event-vision works [18, 25, 74].

5.3. Motion Projections

Having described the emulation of cameras that capture coded exposures and temporal contrasts, we now shift our attention to cameras that emulate sensor motion during exposure, *viz.* motion cameras. We describe two useful trajectories when emulating motion cameras using Eq. (6).

Linear trajectory. The simplest sensor trajectory involves linear motion, where $\mathbf{r}(t) = (bt + c)\,\hat{\mathbf{p}}$ for some constants $b, c \in \mathbb{R}$ and unit vector $\hat{\mathbf{p}}$. As Fig. 5 *(top row)* shows, this can change the scene's frame of reference: making moving objects appear stationary and vice-versa.

Motion-invariant parabolic projection. If motion is along $\hat{\mathbf{p}}$, parabolic integration produces a motion-invariant image [33]—all objects, irrespective of their velocity, are blurred by the same point spread function (PSF), up to a linear shift. Thus, a deblurred parabolic capture produces a sharp image of all velocity groups (Fig. 5 *(bottom row)*). The parabolic trajectory is given by $\mathbf{r}(t) = (at^2 + bt + c)\,\hat{\mathbf{p}}$. We choose a based on the maximum object velocity and b, c

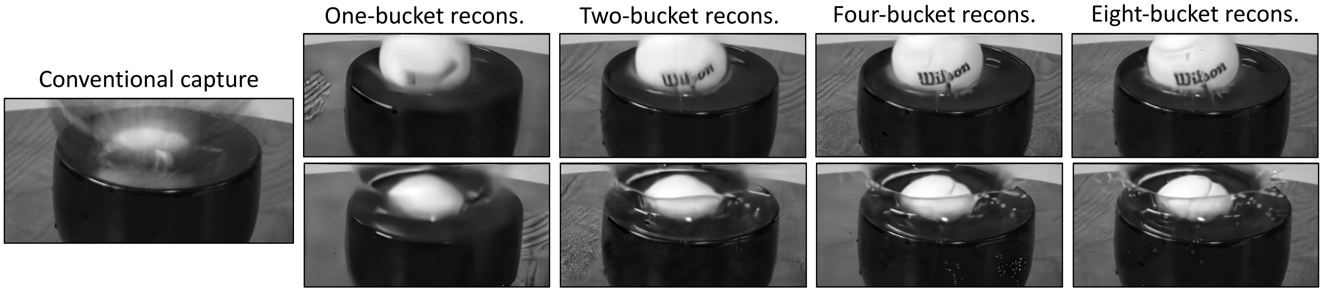


Figure 7: **High-speed videography at** 2000 **FPS** of a tennis ball dropped into a bowl of water, from a 25 Hz readout. The conventional capture provides a visualization of the scene dynamics. It is challenging to reconstruct a large number of frames from a single compressive snapshot. Multi-bucket captures recover frames with significantly greater detail, such as the crown of water surrounding the ball. **We include more sequences (e.g., a bursting balloon) in the supplementary material.**

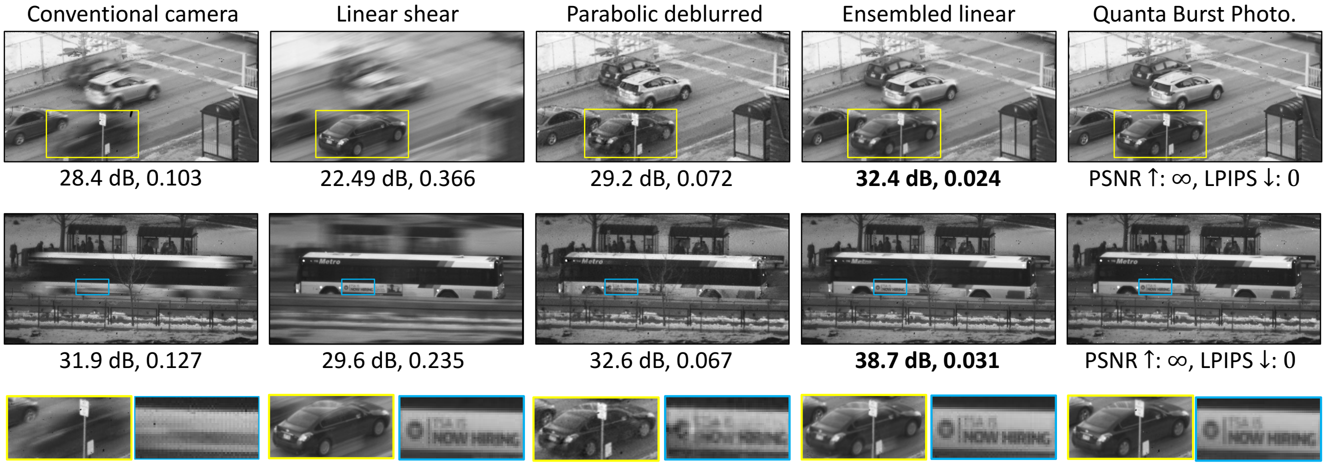


Figure 8: **Deblurring of traffic scenes using motion projections.** Linear projections can recover details of moving objects if their velocity is known. When only the motion direction is known (e.g., road's orientation), a sharp image can be obtained by either deblurring a parabolic projection or by blending multiple randomly-sampled linear projections. We quantitatively compare against the compute- and bandwidth-expensive Quanta Burst Photography [41], based on PSNR and LPIPS [76].

so the parabola's vertex lies at $T/2$. We readily obtain the PSF by applying the parabolic integration to a delta input. Upon deconvolution using the PSF, a parabolic projection provides the optimal SNR for a blur-free image from single capture when only the direction of velocity is known.

Ensembling linear projections. Finally, we leverage the flexibility of photon-cubes to compute multiple linear projections, as seen in Fig. 6. This produces a stack of images where one velocity group is motion-blur free at a time—or a 'motion stack', analogous to a focal stack. This novel construct can be used to compensate motion by blending stack images using cues such as blur orientation or optical flow.

6. Hardware and Experimental Results

We design a range of experiments to demonstrate the versatility of photon-cube projections: both when computations occur after readout (Secs. 6.1 and 6.2), and when they are performed near-sensor on-chip (Sec. 6.3). All photon-cubes were acquired using the SwissSPAD2 array [67], operated using one of two sub-arrays, each having 512 × 256 pixels, and at a frame-rate of 96.8 kHz. For the on-chip experiments, we use the UltraPhase compute architecture to interface with photon-cubes acquired by the SwissSPAD2.

6.1. SoDaCam Capabilities

High-speed compressive imaging. We reconstruct 80 frames from compressive snapshots that are emulated at 25 Hz, resulting in a 2000 FPS video. We decode compressive snapshots using a plug-and-play (PnP) approach, PnP-FastDVDNet [73]. As Fig. 7 shows, it is challenging to recover a large number of frames from a single compressive measurement. Using the proposed multi-bucket scheme significantly improves the quality of video reconstruction.

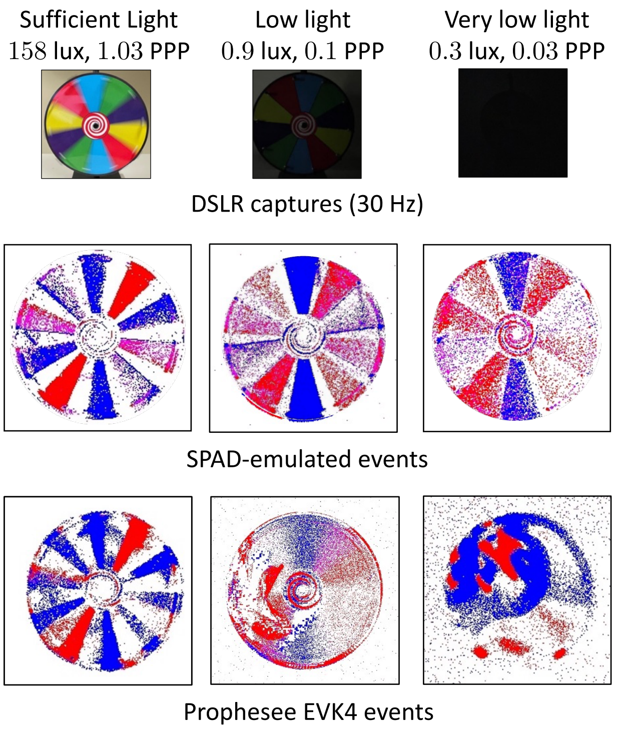


Figure 9: **Comparison to a state-of-the-art event camera.** SPAD-events can capture temporal gradients even when the light-level is reduced by $500\times$, by benefiting from their single-photon sensitivity and bounded brightness response curve. In contrast, low-light induces blur and deteriorates the Prophesee's event stream. As a measure of the light-level, we report the PPP (photons per pixel) averaged across bit-planes and a light meter's reading at the sensor location.

While multi-bucket captures require more bandwidth, this can be partially amortized by coding only dynamic regions, which we show in Supplementary Note 1.

Motion projections on a traffic scene. Fig. 8 shows two traffic scenes captured using a 50 mm focal length lens and at 30 Hz emulation. When object velocity is known, a linear projection can make moving objects appear stationary. If only the velocity direction is known (e.g., road's orientation in Fig. 8), a parabolic projection provides a sharp reconstruction of all objects. We deblur parabolic captures using PnP-DnCNN [75]. We offer an improvement by randomly sampling 8 linear projections along the velocity direction and blending them using the optical flow predicted by RAFT [65] between two short-exposures.

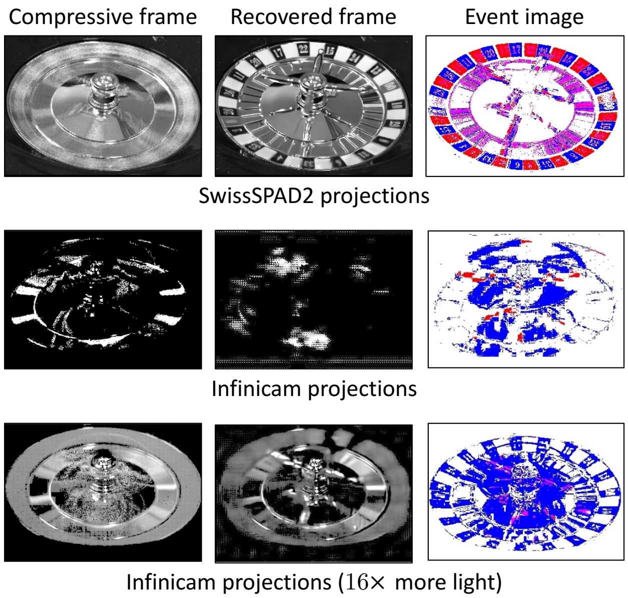


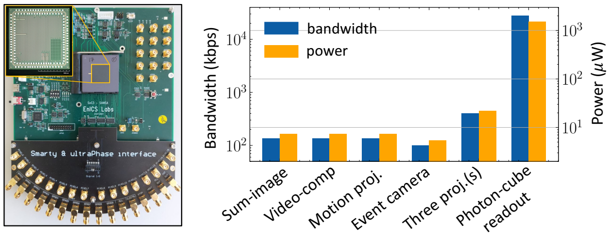
Figure 10: **Comparison against conventional high-speed acquisition at** 4000 **Hz.** *(top)* SPAD projections recover a $16\times$ compressive video and an event image of a spinning roulette wheel. *(middle)* Read-noise corrupts the incident flux in the Infinicam high-speed camera, removing details in frames which are compressed on-the-fly. *(bottom)* Although using a larger aperture to admit more light recovers some detail, noise and compression artifacts still persist.

Low-light event imaging. Fig. 9 compares event-image visualizations of SPAD and that of a state-of-the-art commercial event sensor (Prophesee EVK4), across various light levels, with an accumulation period of 33 ms. For a fair comparison, we bin the Prophesee's events in blocks of 2×2 pixels and use a smaller aperture to account for the lower fill-factor of the SPAD. We tuned event-generation parameters (contrast threshold, integrator decay rate) of both cameras at each light level. Low light induces blur and deteriorates the Prophesee's event stream. In contrast, SPAD-events continue to capture temporal gradients, due to the SPAD's low-light capabilities and its brightness-encoding response curve. We include an ablative study of brightness-encoding functions in Supplementary Note 2.

Our observations are in concurrence with recent works that examine the low-light performance of event cameras [19, 27], and show that SPAD-events can provide neuromorphic vision in these challenging-SNR scenarios.

6.2. Comparison to High-Speed Cameras

Recall, as previously discussed in Sec. 4, that read-noise limits the per-frame SNR of high-speed cameras. To demonstrate this limitation, we compute projections using the 4 kHz acquisition of the Photron Infinicam, a conventional high-speed camera, at a resolution of 1246×240 pixels. We operate the SwissSPAD2 and the Infinicam at ambient light conditions using the same lens specifications.



UltraPhase board

Power and bandwidth measurement

Figure 11: **Power and bandwidth requirements** when computing photon-cube projections on UltraPhase [5] *(left)*, a recent compute architecture designed for single-photon imaging, at 40 Hz readout. *(right)* Our projections act as a compression scheme for photon-cubes, resulting in dramatically reduced sensor-readout and power consumption.

As Fig. 10 shows, read noise corrupts the incident signal in Infinicam and makes it impossible to derive any useful projections. The read noise could be averaged out to some extent if the Infinicam did not perform compression-on-the-fly, but compression is central to the camera's working and enables readout over USB. Using a larger aperture to admit more light improves the quality of computed projections, but the video reconstruction and event image remain considerably worse than corresponding outputs of the SPAD.

6.3. Bandwidth and Power Implications

While Sec. 6.1 has demonstrated the capabilities of photon-cube projections, we now show that our projections can also be obtained in a bandwidth-efficient manner via near-sensor computations. We implement photon-cube projections on UltraPhase (Fig. 11 *(left)*), a novel compute architecture designed for single-photon imaging. UltraPhase consists of 3×6 processing cores, each of which interfaces with 4×4 pixels, and can be 3D stacked beneath a SPAD array. We include visualizations and programming details of a few example projections in Supplementary Note 5.

We measure the readout and power consumption of UltraPhase when computing projections on 2500 bit-planes of the falling die sequence (Fig. 1). The projections include: VCS with 16 random binary masks, an event camera, a linear projection and a combination of the three. We output projections at 12-bit depth and calculate metrics based on the clock cycles required for both compute and readout. As seen in Fig. 11 *(right)*, computing projections on-chip dramatically reduces sensor-readout and power consumption as compared to reading out the photon-cube. Finally, similar to existing event cameras, SPAD-events have a resource footprint that reflects the underlying scene dynamics.

In summary, our on-chip experiments show that performing computations near-sensor can increase the viability of single-photon imaging in resource constrained settings.

7. Discussion and Future Outlook

SoDaCam provides a realization of reinterpretable software-defined cameras [2, 3, 20, 34, 51] at the fine temporal resolution of SPAD-acquired photon-cubes. The proposed computations, or photon-cube projections, can match and in some cases, surpass the capabilities of existing imaging systems. The software-defined nature of photon-cube projections provides functionalities that may be difficult to achieve in conventional sensors. These projections can reduce the readout and power-consumption of SPAD arrays and potentially spur widespread adoption of single-photon imaging in the consumer domain. Finally, future chip-to-chip communication standards may also make it feasible to compute projections on a camera image signal processor.

Adding color to SoDaCam. One way to add color is by overlaying color filter arrays (CFAs) and perform demosaicing on the computed photon-cube projection: depending on the projection, demosaicing could be relatively simple or more complex. As a reference, Bayer CFAs have been considered in the context of both video compressive sensing [73] and event cameras [64]. Incorporating CFAs with motion projections requires careful considerations, e.g., avoiding integrating across pixel locations of differing color.

Future outlook on SPAD characteristics. A key SPAD characteristic that determines several properties of emulated cameras is the frame rate. While no fundamental limitations prevent SPADs from being operated at the frame-rates utilized in this work (~100 kHz), sensor readout and power constraints can preclude high speeds, especially in high-resolution SPAD arrays. Photon-cube projections can enable future large-format SPADs to preserve high-speed information with modest resource requirements.

A platform for comparing cameras. Comparing imaging modalities can be quite challenging since hardware realizations of sensors can differ in numerous aspects, such as their quantum efficiency, fill factor, pixel pitch and array resolution. By emulating their imaging models, SoDaCam can serve as a platform for hardware-agnostic comparisons; for instance, determining operating conditions where one imaging modality is advantageous over another.

A Cambrian explosion of new cameras. Besides comparing cameras, by virtue of being software-defined, SoDaCam can also make it significantly easier to prototype and deploy new unconventional imaging models, and even facilitate sensor-in-the-loop optimization [44, 47, 63] by tailoring photon-cube projections for downstream computer-vision tasks. This is an exciting future line of research.

References

[1] Phantom-v2640. https://www.phantomhighspeed.com/products/cameras/ultrahigh4mpx/v2640. Accessed: 2023-01-28. 4

[2] E. H. Adelson, J. R. Bergen, et al. The plenoptic function and the elements of early vision. *Computational models of visual processing*, 1(2):3–20, 1991. 1, 9

[3] A. Agrawal, A. Veeraraghavan, and R. Raskar. Reinterpretable imager: Towards variable post-capture space, angle and time resolution in photography. In *Computer Graphics Forum*, volume 29, pages 763–772. Wiley Online Library, 2010. 3, 9

[4] I. M. Antolovic, S. Burri, C. Bruschini, R. Hoebe, and E. Charbon. Nonuniformity analysis of a 65-kpixel CMOS SPAD imager. *IEEE Transactions on Electron Devices*, 63 (1):57–64, 2016. doi: 10.1109/TED.2015.2458295. 3

[5] A. Ardelean. *Computational Imaging SPAD Cameras*. PhD thesis, École polytechnique fédérale de Lausanne, 2023. 2, 9

[6] A. Boukhayma, A. Peizerat, and C. Enz. A sub-0.5 electron read noise VGA image sensor in a standard CMOS process. *IEEE Journal of Solid-State Circuits*, 2016. 4

[7] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. doi: 10.1109/TPAMI.1986.4767851. 4

[8] S. J. Carey, A. Lopich, D. R. Barr, B. Wang, and P. Dudek. A 100,000 fps vision sensor with embedded 535GOPS/W 256×256 SIMD processor array. In *2013 Symposium on VLSI Circuits*, pages C182–C183, 2013. 3

[9] J. Chen, S. J. Carey, and P. Dudek. Feature extraction using a portable vision system. In *IEEE/RSJ Int. Conf. Intell. Robots Syst., Workshop Vis.-based Agile Auton. Navigation UAVs*, volume 2, 2017. 3

[10] S. Chen and M. Guo. Live demonstration: Celex-v: A 1m pixel multi-mode event-based sensor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 4, 5

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 4

[12] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, page 369–378. ACM Press/Addison-Wesley Publishing Co., 1997. doi: 10.1145/258734.258884. URL https://doi.org/10.1145/258734.258884. 1, 3

[13] F. M. Della Rocca, H. Mai, S. W. Hutchings, T. Al Abbas, K. Buckbee, A. Tsiamis, P. Lomax, I. Gyongy, N. A. Dutton, and R. K. Henderson. A 128× 128 SPAD motion-triggered time-of-flight image sensor with in-pixel histogram and column-parallel vision processor. *IEEE Journal of Solid-State Circuits*, 55(7):1762–1775, 2020. 3

[14] T. Finateu, A. Niwa, D. Matolin, K. Tsuchimoto, A. Mascheroni, E. Reynaud, P. Mostafalu, F. Brady, L. Chotard, F. LeGoff, H. Takahashi, H. Wakabayashi, Y. Oike, and C. Posch. 5.10 a 1280×720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86µm pixels, 1.066GEPS readout, programmable event-rate controller and compressive data-formatting pipeline. In *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, pages 112–114, 2020. doi: 10.1109/ISSCC19947.2020.9063149. 4, 5

[15] E. R. Fossum. What to do with sub-diffraction-limit (SDL) pixels?—a proposal for a gigapixel digital film sensor (DFS). In *IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors*, pages 214–217, 2005. 3

[16] E. R. Fossum. The quanta image sensor (QIS): Concepts and challenges. In *Imaging and Applied Optics*, page JTuE1. Optica Publishing Group, 2011. doi: 10.1364/COSI.2011.JTuE1. URL http://opg.optica.org/abstract.cfm?URI=COSI-2011-JTuE1. 1

[17] E. R. Fossum, J. Ma, S. Masoodian, L. Anzagira, and R. Zizza. The quanta image sensor: Every photon counts. *Sensors*, 16(8):1260, 2016. 3

[18] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza. Eklt: Asynchronous photometric feature tracking using events and frames. *International Journal of Computer Vision*, 128(3): 601–618, 2020. 6

[19] R. Graca and T. Delbruck. Unraveling the paradox of intensity-dependent DVS pixel noise. *arXiv preprint arXiv:2109.08640*, 2021. 8

[20] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. G. Narasimhan. Flexible voxels for motion-aware videography. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, pages 100–114. Springer, 2010. 3, 9

[21] F. Gutierrez-Barragan, A. Ingle, T. Seets, M. Gupta, and A. Velten. Compressive single-photon 3D cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17854–17864, June 2022. 3

[22] I. Gyongy, N. A. Dutton, and R. K. Henderson. Single-photon tracking for high-speed vision. *Sensors*, 18(2):323, 2018. 3

[23] I. Gyongy, S. W. Hutchings, A. Halimi, M. Tyler, S. Chan, F. Zhu, S. McLaughlin, R. K. Henderson, and J. Leach. High-speed 3D sensing via hybrid-mode imaging and guided upsampling. *Optica*, 7(10):1253–1260, 2020. 3

[24] C. Harris, M. Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 4

[25] J. Hidalgo-Carrió, G. Gallego, and D. Scaramuzza. Event-aided direct sparse odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5781–5790, June 2022. 6

[26] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 4

[27] Y. Hu, S.-C. Liu, and T. Delbruck. v2e: From video frames to realistic DVS events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1312–1321, June 2021. 8

[28] J. Igual. Photographic noise performance measures based on

raw files analysis of consumer cameras. *Electronics*, 8(11): 1284, 2019. 4

[29] A. Ingle, A. Velten, and M. Gupta. High Flux Passive Imaging With Single-Photon Sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[30] A. Ingle, T. Seets, M. Buttafava, S. Gupta, A. Tosi, M. Gupta, and A. Velten. Passive inter-photon imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 3

[31] K. Iwabuchi, Y. Kameda, and T. Hamamoto. Image quality improvements based on motion-based deblurring for single-photon imaging. *IEEE Access*, 9:30080–30094, 2021. doi: 10.1109/ACCESS.2021.3059293. 3

[32] Y. Jiang, I. Choi, J. Jiang, and J. Gu. HDR video reconstruction with tri-exposure quad-bayer sensors. *arXiv preprint arXiv:2103.10982*, 2021. 4

[33] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman. Motion-invariant photography. *ACM Transactions on Graphics (TOG)*, 27(3):1–9, 2008. 4, 6

[34] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 1, 9

[35] Y. Li, M. Qi, R. Gulve, M. Wei, R. Genov, K. N. Kutulakos, and W. Heidrich. End-to-end video compressive sensing using anderson-accelerated unrolled networks. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12, 2020. doi: 10.1109/ICCP48838.2020.9105237. 5

[36] P. Lichtsteiner. 64x64 event-driven logarithmic temporal derivative silicon retina. In *Program 2003 IEEE Workshop on CCD and AIS*, 2003. 4, 5

[37] Y. Liu, F. Gutierrez-Barragan, A. Ingle, M. Gupta, and A. Velten. Single-photon camera guided extreme dynamic range imaging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1575–1585, January 2022. 3

[38] P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady. Coded aperture compressive temporal imaging. *Opt. Express*, 21(9):10526–10545, May 2013. doi: 10.1364/OE.21.010526. URL https://opg.optica.org/oe/abstract.cfm?URI=oe-21-9-10526. 1

[39] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981. 4

[40] J. Ma, S. Masoodian, D. A. Starkey, and E. R. Fossum. Photon-number-resolving megapixel image sensor at room temperature without avalanche gain. *Optica*, 4(12): 1474–1481, Dec 2017. doi: 10.1364/OPTICA.4.001474. URL http://www.osapublishing.org/optica/abstract.cfm?URI=optica-4-12-1474. 3

[41] S. Ma, S. Gupta, A. C. Ulku, C. Bruschini, E. Charbon, and M. Gupta. Quanta burst photography. *ACM Transactions on Graphics*, 39(4):1–16, July 2020. ISSN 0730-0301, 1557-7368. 3, 7

[42] S. Ma, P. Mos, E. Charbon, and M. Gupta. Burst vision using single-photon cameras. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5375–5385, January 2023. 3

[43] S. Mann and R. Picard. Beingundigital'with digital cameras. *MIT Media Lab Perceptual*, 1:2, 1994. 3

[44] J. N. Martel, L. K. Mueller, S. J. Carey, P. Dudek, and G. Wetzstein. Neural sensors: Learning pixel exposures for HDR imaging and video compressive sensing with programmable sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(7):1642–1653, 2020. 9

[45] J. N. P. Martel, L. K. Müller, S. J. Carey, and P. Dudek. Parallel HDR tone mapping and auto-focus on a cellular processor array vision chip. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1430–1433, 2016. doi: 10.1109/ISCAS.2016.7527519. 3

[46] J. N. P. Martel, L. K. Müller, S. J. Carey, and P. Dudek. High-speed depth from focus on a programmable vision chip using a focus tunable lens. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, 2017. doi: 10.1109/ISCAS.2017.8050548. 3, 4

[47] C. A. Metzler, H. Ikoma, Y. Peng, and G. Wetzstein. Deep optics for single-shot high-dynamic-range imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 9

[48] K. Morimoto, A. Ardelean, M.-L. Wu, A. C. Ulku, I. M. Antolovic, C. Bruschini, and E. Charbon. Megapixel time-gated SPAD image sensor for 2D and 3D imaging applications. *Optica*, 7(4):346–354, Apr. 2020. 1, 3

[49] K. Morimoto, J. Iwata, M. Shinohara, H. Sekine, A. Abdelghafar, H. Tsuchiya, Y. Kuroda, K. Tojima, W. Endo, Y. Maehashi, Y. Ota, T. Sasago, S. Maekawa, S. Hikosaka, T. Kanou, A. Kato, T. Tezuka, S. Yoshizaki, T. Ogawa, K. Uehira, A. Ehara, F. Inui, Y. Matsuno, K. Sakurai, and T. Ichikawa. 3.2 megapixel 3D-stacked charge focusing SPAD for low-light imaging and depth sensing. In *2021 IEEE International Electron Devices Meeting (IEDM)*, pages 20.2.1–20.2.4, 2021. doi: 10.1109/IEDM19574.2021.9720605. 1, 3

[50] S. Namiki, S. Sato, Y. Kameda, and T. Hamamoto. Imaging method using multi-threshold pattern for photon detection of quanta image sensor. In *International Workshop on Advanced Imaging Technology (IWAIT) 2022*, volume 12177, page 1217702. SPIE, 2022. 3

[51] S. K. Nayar, V. Branzoi, and T. E. Boult. Programmable imaging: Towards a flexible camera. *International Journal of Computer Vision*, 70:7–22, 2006. 3, 9

[52] C. Posch, D. Matolin, and R. Wohlgenannt. A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2011. doi: 10.1109/JSSC.2010.2085952. 1

[53] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. In *Acm Siggraph 2006 Papers*, pages 795–804. 2006. 4

[54] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6

[55] D. Reddy, A. Veeraraghavan, and R. Chellappa. P2C2: Programmable pixel compressive camera for high speed imaging. In *CVPR 2011*, pages 329–336, 2011. doi: 10.1109/CVPR.2011.5995542. 1, 3

[56] X. Ren, P. W. Connolly, A. Halimi, Y. Altmann, S. McLaughlin, I. Gyongy, R. K. Henderson, and G. S. Buller. High-resolution depth profiling using a range-gated CMOS SPAD quanta image sensor. *Optics express*, 26(5):5541–5557, 2018. 3

[57] C. Scheerlinck, H. Rebecq, D. Gehrig, N. Barnes, R. Mahony, and D. Scaramuzza. Fast image reconstruction with an event camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020. 6

[58] T. Seets, A. Ingle, M. Laurenzis, and A. Velten. Motion adaptive deblurring with single-photon cameras. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1945–1954, January 2021. 3

[59] M.-W. Seo, Y. Shirakawa, Y. Masuda, Y. Kawata, K. Kagawa, K. Yasutomi, and S. Kawahito. 4.3 a programmable sub-nanosecond time-gated 4-tap lock-in pixel cmos image sensor for real-time fluorescence lifetime imaging microscopy. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 70–71, 2017. doi: 10.1109/ISSCC.2017.7870265. 5

[60] T. Serrano-Gotarredona and B. Linares-Barranco. A 128 × 128 1.5% contrast sensitivity 0.9% FPN 3 μs latency 4 mW asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers. *IEEE Journal of Solid-State Circuits*, 48(3):827–838, 2013. 1, 4, 5

[61] P. Shedligeri, A. S, and K. Mitra. A unified framework for compressive video recovery from coded exposure techniques. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1600–1609, January 2021. 5

[62] C. Shi, N. Song, W. Li, Y. Li, B. Wei, H. Liu, and J. Jin. A review of event-based indoor positioning and navigation. 2022. 6

[63] V. Sitzmann, S. Diamond, Y. Peng, X. Dun, S. Boyd, W. Heidrich, F. Heide, and G. Wetzstein. End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging. *ACM Transactions on Graphics (TOG)*, 37(4):114, 2018. 9

[64] G. Taverni, D. Paul Moeys, C. Li, C. Cavaco, V. Motsnyi, D. San Segundo Bello, and T. Delbruck. Front and back illuminated dynamic and active pixel vision sensors comparison. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(5):677–681, 2018. doi: 10.1109/TCSII.2018.2824899. 9

[65] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, 2020. 8

[66] S. Tulyakov, D. Gehrig, S. Georgoulis, J. Erbach, M. Gehrig, Y. Li, and D. Scaramuzza. Time lens: Event-based video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16155–16164, June 2021. 6

[67] A. C. Ulku, C. Bruschini, I. M. Antolovic, Y. Kuo, R. Ankri, S. Weiss, X. Michalet, and E. Charbon. A 512 × 512 SPAD Image Sensor With Integrated Gating for Widefield FLIM. *IEEE Journal of Selected Topics in Quantum Electronics*, 25 (1):1–12, Jan. 2019. ISSN 1077-260X, 1558-4542. doi: 10.1109/JSTQE.2018.2867439. 3, 7

[68] G. Wan, X. Li, G. Agranov, M. Levoy, and M. Horowitz. CMOS image sensors with multi-bucket pixels for computational photography. *IEEE Journal of Solid-State Circuits*, 47 (4):1031–1042, 2012. doi: 10.1109/JSSC.2012.2185189. 5

[69] M. Wei, N. Sarhangnejad, Z. Xia, N. Gusev, N. Katic, R. Genov, and K. N. Kutulakos. Coded two-bucket cameras for computer vision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 3, 4, 5

[70] M. White, S. Ghajari, T. Zhang, A. Dave, A. Veeraraghavan, and A. Molnar. A differential SPAD array architecture in 0.18 μm CMOS for HDR imaging. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 292–296, 2022. doi: 10.1109/ISCAS48785.2022.9937558. 3

[71] T. Yamazaki, H. Katayama, S. Uehara, A. Nose, M. Kobayashi, S. Shida, M. Odahara, K. Takamiya, Y. Hisamatsu, S. Matsumoto, L. Miyashita, Y. Watanabe, T. Izawa, Y. Muramatsu, and M. Ishikawa. 4.9 a 1ms high-speed vision chip with 3D-stacked 140GOPS column-parallel PEs for spatio-temporal image processing. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 82–83, 2017. doi: 10.1109/ISSCC.2017.7870271. 3

[72] F. Yang, Y. M. Lu, L. Sbaiz, and M. Vetterli. Bits from photons: Oversampled image acquisition using binary poisson statistics. *IEEE Transactions on Image Processing*, 21(4): 1421–1436, 2012. doi: 10.1109/TIP.2011.2179306. 3

[73] X. Yuan, Y. Liu, J. Suo, F. Durand, and Q. Dai. Plug-and-play algorithms for video snapshot compressive imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7093–7111, 2022. doi: 10.1109/TPAMI.2021.3099035. 7, 9

[74] J. Zhang, X. Yang, Y. Fu, X. Wei, B. Yin, and B. Dong. Object tracking by jointly exploiting frame and event domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13043–13052, 2021. 6

[75] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte. Plug-and-play image restoration with deep denoiser prior. *arXiv preprint*, 2020. 8

[76] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 7

[77] T. Zhang, M. J. White, A. Dave, S. Ghajari, A. Raghuram, A. C. Molnar, and A. Veeraraghavan. First arrival differential LiDAR. In *2022 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12, 2022. doi: 10.1109/ICCP54855.2022.9887683. 3

[78] W. Zhang and L. Lin. Light field flow estimation based on occlusion detection. *Journal of Computer and Communications*, 5(3):1–9, 2017. 1