EventPS: Real-Time Photometric Stereo Using an Event Camera

Bohan Yu[1,2] Jieji Ren[3] Jin Han[4,5] Feishi Wang[1,2] Jinxiu Liang[1,2] Boxin Shi[1,2*]

[1] National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University

[2] National Engineering Research Center of Visual Technology, School of Computer Science, Peking University

[3] School of Mechanical Engineering, Shanghai Jiao Tong University

[4] Graduate School of Information Science and Technology, The University of Tokyo [5] National Institute of Informatics

{ybh1998, wangfeishi, cssherryliang, shiboxin}@pku.edu.cn

jiejiren@sjtu.edu.cn, jinhan@nii.ac.jp

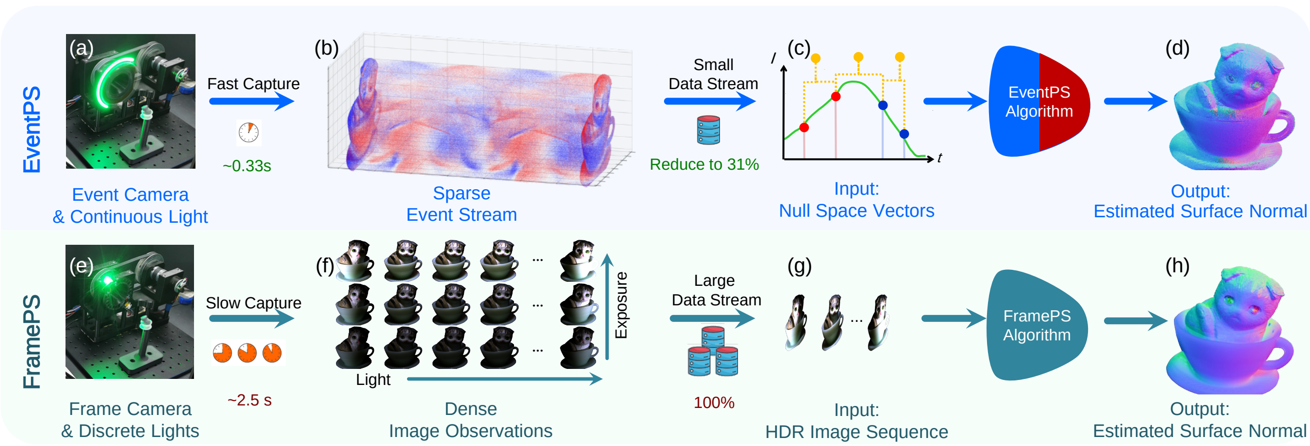

Figure 1. Comparison between the proposed EventPS and its frame-based counterpart, *i.e.*, FramePS. Bottom: FramePS estimates the surface normal (h) by analyzing images of an object illuminated from multiple directions (e). It involves capturing a series of exposure-bracketing images[2] (f, g), a process that is not only time-consuming but also demands substantial bandwidth for processing. Top: In contrast, EventPS estimates the surface normal by analyzing the events triggered by a continuously rotating light source (a). The unique attributes of event cameras, *e.g.*, low latency, high dynamic range, and low redundancy in data representation (b), enable EventPS, a rapid and highly efficient real-time solution (c, d), which significantly reduces the bandwidth usage while maintaining comparable performance to FramePS.

Abstract

Photometric stereo is a well-established technique to estimate the surface normal of an object. However, the requirement of capturing multiple high dynamic range images under different illumination conditions limits the speed and real-time applications. This paper introduces EventPS, a novel approach to real-time photometric stereo using an event camera. Capitalizing on the exceptional temporal resolution, dynamic range, and low bandwidth characteristics of event cameras, EventPS estimates surface normal only from the radiance changes, significantly enhancing data efficiency. EventPS seamlessly integrates with both optimization-based and deep-learning-based photometric stereo techniques to offer a robust solution for non-Lambertian surfaces. Extensive experiments validate the effectiveness and efficiency of EventPS compared to frame-based counterparts. Our algorithm runs at over 30 fps in real-world scenarios, unleashing the potential of EventPS in time-sensitive and high-speed downstream applications.

1. Introduction

Photometric Stereo (PS) [53], a technique that estimates the orientation of surface normals by analyzing images of an object illuminated from various directions, is distinctive by its ability to reconstruct high-resolution and precise surface details, especially under controlled lighting conditions.

*Corresponding author: Boxin Shi

[1]Code available: https://codeberg.org/ybh1998/EventPS

Due to deviations from an ideal Lambertian image formation model such as shadows, specular reflections, and various types of noise [17], it is complex and time-consuming to achieve a robust normal estimation in traditional Frame-based PS (FramePS). As shown in Fig. 1 (f), typically, this process requires capturing a series of exposure bracketing images[2] using a stationary camera under the illumination of multiple, sequentially lit distant light sources (*e.g.*, around 100 lights [40, 45]). This laborious process hinders real-time applications of PS.

Recent efforts in pushing real-time PS fall into two categories. One group of methods utilizes multi-spectral cameras to simultaneously obtain observations of objects in varying oriented multi-spectral lighting conditions [4, 8, 10, 23–25, 30, 36, 47]. Despite the single-shot data-capturing process, the ambiguity between the colors of the lights and the object poses challenges in normal estimation. Another direction involves high-speed cameras synchronized with carefully controlled light sources, which aims to expedite the image-capturing process [5, 32, 49]. However, this setup requires a high data throughput capability in cameras and experimental facilities, which becomes a barrier to their practical implementation in real-time applications, especially with limited power and cost.

Event cameras, characterized by their high temporal resolution, high dynamic range, and low bandwidth requirements, have recently been recognized as a promising solution for real-time vision applications [6]. Unlike traditional frame-based cameras, event cameras only record logarithmic scene radiance changes. This characteristic is advantageous in many scenarios. For example, it swiftly establishes the temporal correspondences and spatial disparities for multi-view stereo [38] or 3D reconstruction under structured light [33, 34]. However, their nature of radiance changes instead of absolute values deviates from the FramePS problem. The exploration of *how to effectively utilize the unique attributes of event cameras for real-time PS* remains an open question.

In this paper, we propose a reformulation of the PS problem to observations derived solely from scene radiance changes under varying lighting conditions, which specifically tailors to advantageous characteristics of event cameras. As shown in Fig. 1 (a), an object is illuminated by a high-speed rotating light source (up to 1800 revolutions per minute, rpm) that continuously induces radiance changes and triggers event signals. Each event is associated with the lighting direction of the triggering timestamp (Fig. 1 (b)). Assuming the Lambertian reflectance model (we will release this assumption later), each pair of consecutive events is transformed into a vector orthogonal to the surface normal, named "null space vector" (Fig. 1 (c)). The surface normal for each pixel is then determined from at least two linearly independent null space vectors without ambiguity (Fig. 1 (d)). Owing to the unique attributes of event cameras, this process enables the capturing of observations with a high dynamic range under rapidly changing lighting, while maintaining economical data efficiency. This approach, termed **EventPS**, allows us to harness the inherent strengths of event cameras for achieving real-time PS.

For real scenes where events are noisy, surface normals are obtained more robustly by solving a least squares minimization problem using all null space vectors. By integrating Singular Value Decomposition (SVD) [7] with EventPS, our method notably achieves 30 frames per second (fps) in normal estimation. Additionally, acknowledging the inherent challenges in handling non-Lambertian surfaces, we propose deep learning variants [2, 13] under our EventPS formulation. We develop a custom validation platform that demonstrates the feasibility of our approach and highlights the potential of EventPS in high-speed, time-sensitive applications such as real-time 3D reconstruction. Our experiments show that EventPS matches the performance of FramePS while using only 31%[3] of the bandwidth, a testament to its effectiveness and efficiency. The key contributions of our work are summarized as follows:

- We are the first to formulate that the surface normals can be estimated from continuous radiance changes w.r.t. lighting recorded by an event camera, which achieves a significant bandwidth reduction compared to FramePS.
- We propose EventPS integrated with both optimization-based and deep-learning-based approaches to handle Lambertian and non-Lambertian surfaces.
- We build up a validation platform with a high-speed rotating light source, showcasing that the proposed EventPS estimates surface normals in real-time with 30 fps output.

2. Related Works

2.1. Photometric Stereo Methods

Since the PS was proposed in the 1980s [53], both optimization-based and deep-learning-based [40] methods have been proposed to enhance performance. Most representative optimization-based methods have been comprehensively discussed in [45], so we focus on reviewing deep-learning-based solutions in the following part.

Recent PS methods predominantly adopt deep-learning-based approaches, which are divided into two categories: all-pixel and per-pixel [56]. All-pixel methods [2, 3] combined the global information from observed images and light directions, while per-pixel methods [13, 43, 55] took the observations of each pixel under various light directions to estimate the surface normal.

[2] High Dynamic Range (HDR) images are usually required in FramePS for accurately observing the specular regions on the object surface.

[3] Average bandwidth of three algorithms. More details are described in Sec. 4.3.

To improve the performance of deep-learning-based PS methods, researchers combined the advantages from per-pixel and all-pixel methods [54], augmented the observation maps for modeling global illumination [28], and utilized inverse rendering to estimate surface normal [26, 48]. Besides, advanced learning models and techniques [21] were also introduced to handle realistic complexity, such as attention-based weight [20, 22], transformer [14], and differentiable modeling [27]. Furthermore, general lighting and feature representation [15] reshaped the deep-learning-based PS and achieved comparable performance with 3D scanners [16]. However, a significant number of images under various illuminations are still necessary. The serialized capturing process considerably limits PS application in dynamic scenarios.

The key to accelerating the imaging process of PS lies in optimizing the observation process [46] with high-speed cameras and synchronized illumination [24]. However, the cost greatly rises with the frame rate increasing Other researchers introduced multi-spectral imaging systems [25, 36, 47] to observe the object under varying directional illuminations with a single shot, which significantly enhances the efficiency of PS. However, the limitations of multi-spectral cameras [8, 23], such as the number of bands, the crosstalk and intensity inconsistency (*e.g.*, unknown illumination, surface reflectance, camera's spectral response) across different colors, introduce additional challenges to surface normal estimation [19].

2.2. Event Camera based 3D Reconstruction

Event cameras detect radiance changes in the scene, which could be induced by camera/object movement or illumination changes. We divide the related research into two categories: motion-based and active illumination-based methods. For motion-based methods, EMVS [38] and EvAC3D [51] treated individual events as rays to estimate a semi-dense 3D structure and an object mesh from an event camera with known trajectory. Besides, event-based neural radiance fields (NeRF) [1, 12, 29, 31, 37, 41] have emerged as a significant breakthrough in leveraging the event signals with high temporal resolution for constructing volumetric scene representations. Please refer to the comprehensive survey [11] for a summary of event-based SLAM methods. For active illumination-based methods, researchers applied structured light [33, 34] and maximized the spatio-temporal correlation between the projector and an event camera for depth sensing. EFPS-Net [42] interpolated the sparse event observation maps and incorporated them with the RGB images to predict the surface normal maps under ambient light. There are also methods using global illumination changes (*e.g.*, turning on the light in a darkroom [9] or applying rotating polarizer [35]) to reconstruct iso-contour or estimate surface normals.

3. Proposed Method

3.1. Problem Formulation

Photometric stereo. Assuming an object illuminated by an ideal distant light source, the radiance of the light source is constant and the direction is described as a normalized lighting vector function $\mathbf{L}(t)$ w.r.t. time t. For a pixel at image coordinate $\mathbf{x} = (x, y)$ with normal vector $\mathbf{n}_\mathbf{x}$ and diffuse albedo $a_\mathbf{x}$, under Lambertian assumption, the reflected radiance of this pixel $\hat{I}_\mathbf{x}(t)$ is:

$$\hat{I}_\mathbf{x}(t) = \max\left[0, a_\mathbf{x}(\mathbf{n}_\mathbf{x} \cdot \mathbf{L}(t))\right]. \quad (1)$$

Event formation model. Event cameras capture scene radiance changes on a logarithmic scale. Each pixel measures the radiance changes asynchronously. When the changes of logarithmic radiance at the pixel $\mathbf{x}$ reaches a triggering threshold C, an event $\{\mathbf{x}, p, t\}$ will be triggered, where t is the timestamp, and $p \in \{-1, +1\}$ is the polarity which represents the decrease or increase of radiance. Assume there are totally K events triggered at pixel $\mathbf{x}$ during a short period of time. These events are represented as $\mathcal{E}_\mathbf{x} = \{\mathbf{x}, p_k, t_k\}$, where $k = \{1, 2, ..., K\}$. The change of radiance value in pixel $\mathbf{x}$ from t_{k-1} to t_k becomes:

$$\log(I_\mathbf{x}(t_k) + \epsilon) = \log(I_\mathbf{x}(t_{k-1} + \eta) + \epsilon) + p_k C, \quad (2)$$

where ϵ is a small offset value to avoid taking the logarithm of zero, and η is the refractory time of the pixel [6]. By omitting the offset value and refractory time in Eq. (2) and performing exponentiation on both sides, we obtain the following equation:

$$I_\mathbf{x}(t_k) = \exp(p_k C) \cdot I_\mathbf{x}(t_{k-1}). \quad (3)$$

Substituting Eq. (1) into Eq. (3), we obtain:

$$\begin{aligned} &\max\left[0, a_\mathbf{x}(\mathbf{n}_\mathbf{x} \cdot \mathbf{L}(t_k))\right] = \\ &\exp(p_k C) \cdot \max\left[0, a_\mathbf{x}(\mathbf{n}_\mathbf{x} \cdot \mathbf{L}(t_{k-1}))\right]. \end{aligned} \quad (4)$$

Given the captured events $\mathcal{E}_\mathbf{x}$ at pixel $\mathbf{x}$ and lighting direction $\mathbf{L}(t)$, our goal is to find the following function f that estimates the surface normal $\hat{\mathbf{n}}_\mathbf{x}$ at pixel $\mathbf{x}$ as close to $\mathbf{n}_\mathbf{x}$ as possible:

$$\hat{\mathbf{n}}_\mathbf{x} = f(\mathcal{E}_\mathbf{x}, \mathbf{L}(t)). \quad (5)$$

3.2. EventPS Model

In this subsection, we start from a static object with a Lambertian surface captured by an event camera using ideal event-triggering mechanisms to explain how the EventPS model works. The proposed algorithms based on the EventPS model in the following subsections (Sec. 3.3 and Sec. 3.4) deal with all the non-ideal effects in real scenarios (generic BRDF, noisy events, and dynamic scenes).

As shown in Fig. 2, we observe that there are three properties for the event signals triggered in the PS setting that make EventPS possible:

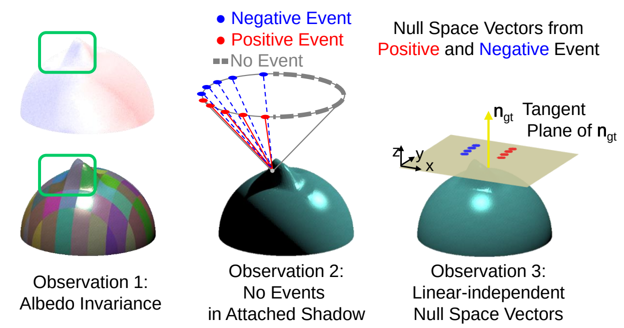

Figure 2. The key observations on event signal characteristics. (1) Albedo invariance: Surface albedo patterns at the bottom are not visible from the events on the top. (2) No events in attached shadow: For light directions on the right half circle, the current pixel is in the attached shadow and does not trigger any event. (3) Linear-independent null space vectors: The null space vectors spanning a tangent plane uniquely determines a surface normal.

Observation 1: Albedo invariance. Event signals are irrelevant to surface albedo $a_{\mathbf{x}}$. Since there are $a_{\mathbf{x}}$ on both sides of Eq. (4), we remove the $a_{\mathbf{x}}$. It means the surface albedo does not affect the event triggering given the same changes in lighting directions. Thus, Eq. (4) can be simplify it as:

$$\max\left[0, \mathbf{n_x} \cdot \mathbf{L}(t_k)\right] = \exp(p_k C) \cdot \max\left[0, \mathbf{n_x} \cdot \mathbf{L}(t_{k-1})\right]. \tag{6}$$

Observation 2: No events in attached shadow. From Eq. (2), we infer that the derivative of $I_{\mathbf{x}}(t_k)$ must be non-zero at t_k. Otherwise, there will be no events triggered. This property indicates that the event signal does not contain redundant information for pixels in the attached shadow region and $\hat{I}$ should be greater than 0 at any event timestamp t_k. Therefore, we remove the $\max$ operator from both sides of Eq. (6) and obtain:

$$\begin{aligned} \mathbf{n_x} \cdot \mathbf{L}(t_k) = \exp(p_k C)(\mathbf{n_x} \cdot \mathbf{L}(t_{k-1})), \\ \textit{i.e.}, \;\; \mathbf{n_x} \cdot (\mathbf{L}(t_k) - \exp(p_k C) \cdot \mathbf{L}(t_{k-1})) = 0. \end{aligned} \tag{7}$$

For each pixel, we convert each pair of successive event signals into a vector that lies in the tangent plane of the object surface at this pixel, which is perpendicular to the surface normal. We call these vectors *null space vectors*, which are represented as $\mathbf{z}_k$, where $k = \{1, 2, ..., K-1\}$:

$$\mathbf{z}_k = \mathbf{L}(t_{k+1}) - \exp(p_{k+1} C)\mathbf{L}(t_k). \tag{8}$$

Combining Eq. (7) and Eq. (8), we verify that null space vectors are perpendicular to the surface normal, *i.e.*, $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_{K-1}\} \perp \mathbf{n_x}$.

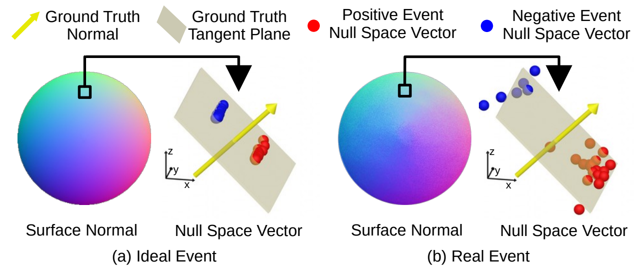

Figure 3. Visualization of null space vectors and estimated normal maps for (a) a Lambertian sphere with ideal event triggering model and (b) a non-Lambertian sphere with real events.

Observation 3: Linear-independent null space vectors. To determine the surface normal of each pixel, at least 2 null space vectors that are linearly independent are required. If all null space vectors are linearly correlated, there would be infinite surface normal vectors perpendicular to all null space vectors. When applying convex curves at each round as the scanning pattern, any 3 points on this curve are not on the same line, which means all the null space vectors should not be linearly dependent:

$$\mathbf{z}_i \neq \gamma \mathbf{z}_j, \quad \forall i \neq j \text{ and } \gamma \neq 0. \tag{9}$$

Therefore, for each pixel, as long as we have obtained 2 null space vectors, the tangent plane is determined, and then we can calculate the unique surface normal at that pixel.

We use two examples in Fig. 3 to verify the validity of the EventPS formulation. In case (a), we show a point on the sphere with Lambertian surface and the ideal event triggering model. We visualize the positive and negative null space vectors computed from Eq. (8). As visualized in Fig. 3 (a), all of the null space vectors are perfectly lying on the tangent surface (gray transparent plane), which determines the unique normal direction (yellow arrow). In case (b), we show the scenario with non-Lambertian surface captured by a real event camera (more details about the experiment setup will be introduced in Sec. 4.1). As visualized in Fig. 3 (b), even with offsets caused by non-ideal reflectance model and noise events, the null space vectors are still around the tangent plane.

To demonstrate that surface normal can be clearly described by the profile of event signals, we show an example in Fig. 4. We plot the radiance changes and event signals triggered along the rotation of light direction using 4 points in different directions. When the light source is rotating with the azimuth angle $\phi_{\mathbf{L}}$ sweeping from $0°$ to $360°$, the radiance of blue, orange, and green points decreases. The red point has a $90°$ delay due to the difference in surface normal azimuth angle. As the elevation angle increases (blue-orange-green points), the change of radiance becomes smoother and the number of events triggered monotonically decreases. The unique events triggering pattern (*i.e.*, times-

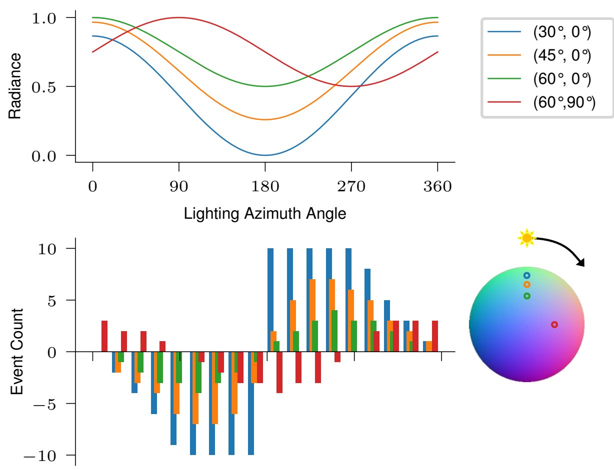

Figure 4. Given 4 points with coordinates (elevation angle θ, azimuth angle ϕ) of blue: $(30°, 0°)$, orange: $(45°, 0°)$, green: $(60°, 0°)$, and red: $(60°, 90°)$ on a sphere, and a light source rotating in a clockwise circle, the radiance changes (top) and events triggered (bottom) of the 4 points w.r.t. light direction changing are plotted. The bottom part shows event number determines the normal elevation angle (comparing blue, orange, and green points), while the zero-crossing point determines the normal azimuth angle (comparing green and red points).

tamp and number) at each point clearly reflects the radiance changes. Therefore, we can directly get the normal vector at each point solely from event signals without any ambiguity.

Next, we will introduce the optimization-based and deep-learning-based EventPS solutions to estimate the surface normal from the noisy null space vectors robustly.

3.3. EventPS by Optimization

For each pixel, we combine all the null space vectors into a $3 \times (K-1)$ matrix $\mathbf{Z}_\mathbf{x}$. Theoretically, at least 3 events are required to get a rank-2 matrix $\mathbf{Z}_\mathbf{x}$ for surface normal estimation. Given sufficient events (*i.e.*, $K > 3$), we define the optimization target to estimate the surface normals $\hat{\mathbf{n}}_\mathbf{x}$ as minimizing the following mean square error (MSE):

$$\underset{\hat{\mathbf{n}}_\mathbf{x}}{\operatorname{argmin}} \|\mathbf{Z}_\mathbf{x}^\top \hat{\mathbf{n}}_\mathbf{x}\|_2. \tag{10}$$

This optimization problem is solved by SVD. We calculate the eigenvector corresponding to the smallest eigenvalue of the matrix $\mathbf{Z}_\mathbf{x}\mathbf{Z}_\mathbf{x}^\top$, then we obtain the surface normal $\hat{\mathbf{n}}_\mathbf{x}$. We name this method **E**vent **P**hotometric **S**tereo **OP**timization (**EventPS-OP**).

It has been verified on a benchmark [44] that adding a threshold to filter out the brightest region (most likely in specular highlight) and the darkest region (most likely in attached/cast shadow) effectively improves the PS accuracy

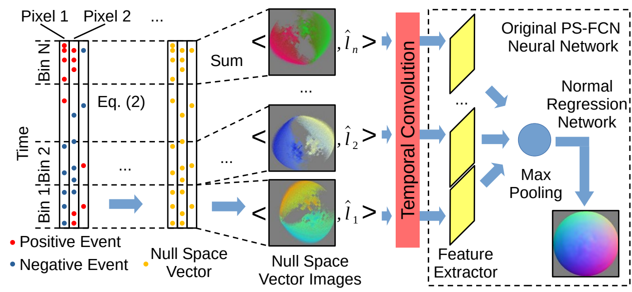

Figure 5. EventPS-FCN structure. The events triggered within each time bin are summed up and converted to null space vectors. Then the null space vector maps are fed into the PS-FCN [2] in replacement of the images.

solved by least squares [45]. In EventPS, due to the lack of absolute radiance information, we can hardly add such a threshold to the event signals. However, events are triggered at a high frequency when intensity variations with high contrast are observed. In PS settings, this usually happens when a point is crossing shadow boundaries (including attacked shadows and cast shadows) or specular highlights. By setting a threshold on event triggering frequency, we can achieve a similar goal as adding a threshold to the least squares method in the frequency domain. The filtered null space vector $\hat{\mathbf{Z}}$ is:

$$\hat{\mathbf{Z}} = \{\mathbf{z}_k \mid k > 1 \text{ and } t_k > t_{k-1} + \delta\}, \tag{11}$$

where δ is the time threshold and $\delta \geq \eta$. With a larger δ more null space vectors are removed by this filter, resulting in a stricter filtering on the EventPS-OP algorithm.

3.4. EventPS by Deep Learning

In FramePS, deep-learning-based methods [2, 13] demonstrate higher robustness against shadows, specular reflection, and inter-reflection thanks to the prior learned from the large-scale synthetic training dataset. To improve the robustness and generalization of EventPS, we adapt two frame-based deep learning methods, *i.e.*, PS-FCN [2] and CNN-PS [13][4] to the modality of event signals.

The original PS-FCN [2] applies convolution layers to each individual image under specific lighting and merges multiple image features by max pooling. As illustrated in Fig. 5, we adapt PS-FCN [2] to event modality (named as **EventPS-FCN**) by constructing null space vector images as the input to maintain the intra-pixel relationship. We first divide the scanning time period of interest (typically a whole circle) into N bins. The events are converted to null space vectors using Eq. (8). The null space vector images are formed by summing up all the null space vectors

[4] According to the survey paper [56], these approaches represent two typical categories of deep-learning-based PS formulated in "all-pixel" [2] and "per-pixel" [13] manner, respectively.

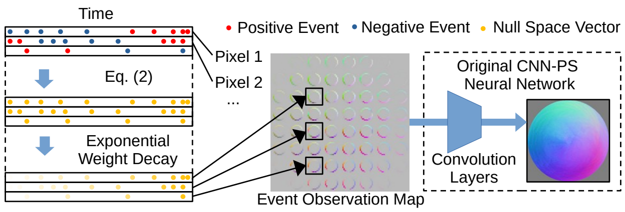

Figure 6. EventPS-CNN structure. The null space vectors are calculated from the events of each pixel, which are accumulated as event observation maps and fed into the CNN-PS [13] architecture in replacement of the original frame observation maps. The observation maps are down-sampled by 32×32 times for visualization.

in each pixel within each time bin, which share the light direction changes. We follow the original PS-FCN [2] design by adding a light direction $\hat{\mathbf{L}}_i$ channel to each null space vector image for feature extraction. Since event features are much sparser than image features and the differences between the adjacent time bins are not distinct, we add two temporal convolution layers to extract temporal features from events of adjacent bins. Then features from all bins are max-pooled together to estimate surface normal.

The original CNN-PS [13] treats each pixel individually by extracting a 32×32 observation map from each pixel and applying convolution layers on such an observation map. Similarly, the conversion from event signals into null space vectors using the proposed EventPS formulation is also performed on a per-pixel basis. As illustrated in Fig. 6, we modify the definition of observation map to adapt the original CNN-PS [13] to the event modality (named as **EventPS-CNN**). In our event observation maps, we increase the number of channels from 1 (gray-scale image) to 3 (x, y, z axis of the null space vector). Each pixel represents a null space vector at the corresponding lighting direction. In this way, all the null space vectors at each pixel are gathered in this event observation maps and fed to the original CNN-PS [13] model. Compared to the time bins in EventPS-FCN, the observation map contains more information for each pixel. As a result, more details about each individual null space vector are preserved in EventPS-CNN.

4. Experiment

4.1. Implementation Details

Algorithms implementation. To demonstrate the real-time performance of our method, we implement the event pre-processing part (for EventPS-OP, EventPS-FCN, and EventPS-CNN) and SVD part (for EventPS-OP only) with GPU acceleration written in Rust and OpenCL. We implement an asynchronous pipeline for EventPS-OP to keep updated with the latest incoming events for lower latency, and synchronous pipelines for EventPS-FCN and EventPS-CNN to wait and process all the events for better quality. The EventPS-FCN neural network is fine-tuned and evaluated with the checkpoint from the original PS-FCN [2] using PyTorch. For EventPS-CNN, we implement a PyTorch version similar to the original CNN-PS [13] and train it from scratch. More details can be found in the released source code (upon acceptance of this paper).

Validation platform. To verify the performance of the algorithms on real-world objects, we design a high-speed illumination and capturing validation platform. There is a green LED light source powered by an in-suit Lithium-ion battery. The LED is mounted on a rotating axis and driven by a synchronous belt-wheel system with a DC motor at up to 1800 rpm, resulting in the high-speed "circle" scanning pattern. A Hall effect angular sensor is installed to detect the LED position, which is sent to the event camera for synchronization. We use a Prophesee EVK4 HD camera (with an IMX636 sensor) to capture event signals during rotation. The two "contrast sensitivity threshold biases" are set to -20, and the "dead time bias" is set to -20, resulting in about 580 µs refractory time.

4.2. Datasets

Synthetic dataset. To train the deep-learning-based algorithms for systemic and controllable comparison, we build a pipeline to render a synthetic dataset and generate simulated event streams. We choose all the objects from the Blobby dataset [18] and 15 objects from the Sculpture dataset [52]. For each object, we add random transformation and random BRDF textures similar to previous deep-learning-based PS methods [2, 13]. We choose three types of scanning patterns for lighting in the synthetic dataset: "circle" for mechanical feasibility, "hypotrochoid" to avoid blind area, and "DiLiGenT" for compatibility of the following semi-real dataset. Then we pick a scanning pattern with random parameters and use a ray-tracing renderer to render 600 dense images under rotating lighting for 6 rounds. These images are converted to event streams with an event simulator ESIM [39].

Semi-real dataset. Popular real datasets for FramePS [40, 45, 50] only contain images captured under several discrete lighting directions. We select the images at the out-most border light directions from DiLiGenT dataset [45] and convert them to event streams with event simulator [39] to generate this semi-real dataset named **DiLiGenT-Ev**.

Real dataset. To validate the performance of the proposed EventPS methods, we fabricate 5 objects and capture a real dataset with ground truth normal maps. The real dataset covers simple geometry (BALL), spatially-varying albedo (BALLCVPR), and shapes with moderate details (BUNNY) and complex details (HORSE, TIGER). Each ob-

Table 1. Full comparison results of EventPS and FramePS methods on DiLiGenT-Ev dataset. The second row is the number (#) of events per round for each data. The middle three rows show the MAE of our EventPS. The last three rows show the percentage of data rate that EventPS requires to achieve the same MAE compared to the FramePS counterparts.

		BALL	BUDDHA	CAT	COW	GOBLET	HARVEST	POT1	POT2	READING	Average
	# Events	260 k	176 k	203 k	251 k	112 k	179 k	141 k	189 k	201 k	192 k
MAE	EventPS-OP	10.99	18.73	12.74	26.51	18.43	36.06	13.78	15.75	24.61	19.73
	EventPS-FCN	7.49	18.13	11.42	20.61	18.07	26.05	12.83	16.59	15.16	16.26
	EventPS-CNN	10.44	16.79	11.88	20.60	16.44	25.26	12.93	15.54	18.19	16.45
Data Rate	EventPS-OP	38%	31%	23%	13%	17%	47%	21%	20%	23%	25.86%
	EventPS-FCN	45%	61%	35%	52%	29%	37%	37%	33%	29%	39.82%
	EventPS-CNN	45%	31%	26%	37%	11%	27%	21%	13%	29%	26.61%

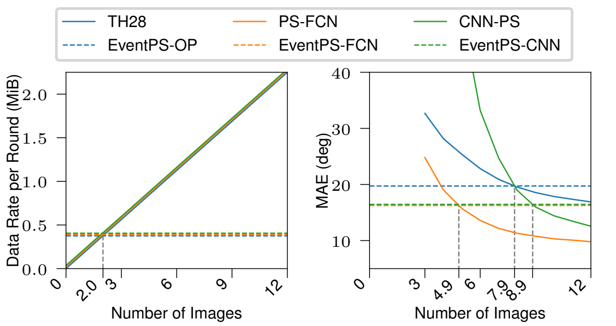

Figure 7. Comparison of data rate and MAE between FramePS and EventPS. On the left, the data rate for FramePS increases linearly as the number of images increases. In contrast, EventPS has a low and constant data rate paramount to about 2 frame images. On the right, the MAE for FramePS decreases with more images. EventPS achieves comparable MAE as about 7.9 images (for EventPS-OP), 8.9 images (for EventPS-CNN), and 4.9 images (for EventPS-FCN).

ject is captured using our validation platform (rotating at 240 rpm in a darkroom for better quality[5]).

4.3. Comparison with FramePS

We conduct a quantitative comparison of the proposed EventPS with the FramePS counterparts on the DiLiGenT-Ev dataset. To compute the data rate required by the event input and frame input, we assume that the event streams employ 16-bit Prophesee EVT 3.0[6] format, and frame images are captured as 8-bit gray-scale images with 3 exposure bracketing. For the three FramePS algorithms *i.e.* TH28 [45] (least square method with [20%, 80%] thresholding, counterpart of EventPS-OP), CNN-PS [13] (counterpart of EventPS-CNN), and PS-FCN [2] (counterpart of EventPS-FCN), we randomly select images from 96 light directions in DiLiGenT dataset [45]. For three EventPS algorithms, different numbers of events are generated for each scene. The Mean Angular Error (MAE) and data rate comparison are shown in Tab. 1. On average, EventPS reduces the required data rate to around 25.9% (for EventPS-OP), 39.8% (for EventPS-FCN), and 26.6% (for EventPS-CNN).

As shown in Fig. 7, FramePS shows a linear increase in data rate as the number of input images increases, accompanied by a decrease in normal MAE. In contrast, the proposed EventPS has a constant data rate and MAE. For each algorithm, the cross point of data rate is on the left, while the cross point of MAE is on the right. This indicates that EventPS achieves smaller MAE with better data efficiency. For qualitative evaluation, we show three object examples in Fig. 8, which indicates that the error distributions of the proposed EventPS evenly across the object.

[5]The impact of rotation speed on normal estimation quality can be found in the supplementary material.

[6]https://docs.prophesee.ai/stable/data/encoding_formats/evt3.html

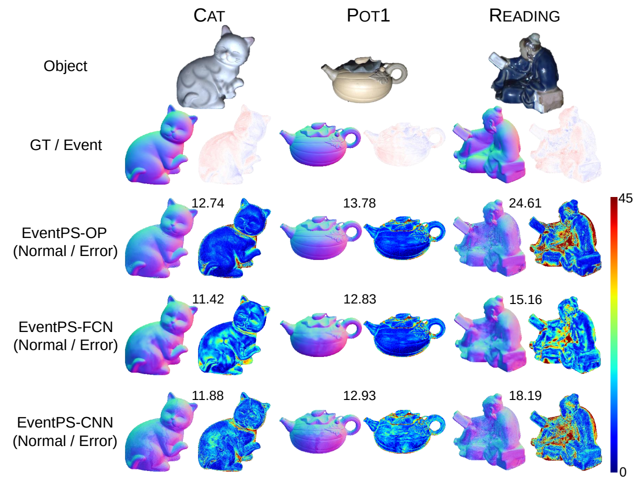

Figure 8. Results on DiLiGenT-Ev dataset. The first row shows the preview of our objects. The second row displays ground truth surface normals and simulated events. The last three rows plot the estimated surface normals (with MAE on the top right corner) and the corresponding angular error maps.

4.4. Evaluation on Real Camera

Results on static objects. We evaluate the performance of EventPS on real data. The results are shown in Tab. 2. On average, our EventPS achieve MAE of 18.8 (for EventPS-OP), 14.7 (for EventPS-FCN), and 17.6 (for EventPS-CNN), which demonstrates the effectiveness of utilizing only event signals for PS. We show 3 object examples and normal estimation results in Fig. 9. The left example shows a ball with spatially varying albedo. We can hardly see the "CVPR" words in the captured event signals and the estimated normal map, demonstrating the "albedo invariance" property of EventPS. The MAEs are higher in the boundaries of the normal estimation results, which is due to the near-light effects (only around 12 cm light-object distance) and coarsely aligned lighting.

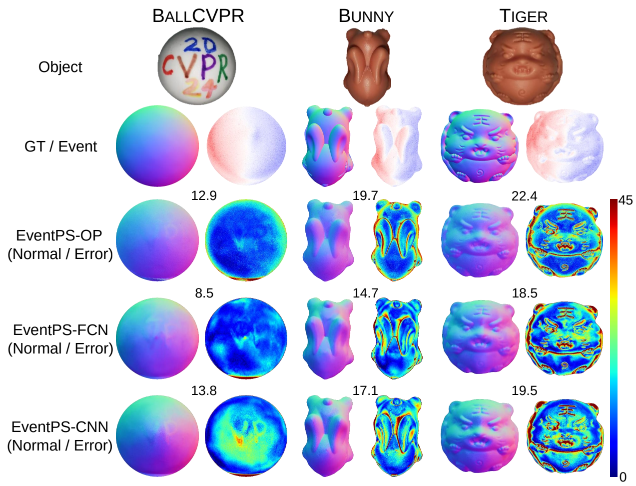

Figure 9. Results on real dataset. The first row shows the preview of our objects. The second row displays ground truth surface normals and captured events. The last three rows plot the estimated surface normals and the corresponding angular error maps.

Table 2. Results of EventPS on real dataset.

	BallCVPR	Ball	Bunny	Horse	Tiger	Average
EventPS-OP	12.9	14.2	19.7	24.8	22.4	18.8
EventPS-FCN	8.5	10.6	14.7	21.2	18.5	14.7
EventPS-CNN	13.8	12.2	17.1	25.3	19.5	17.6

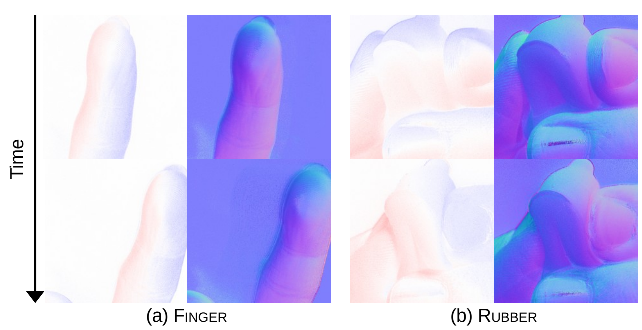

Figure 10. Results on dynamic objects. (a) A human finger movement. (b) The hand-pinching process of a soft rubber toy[7].

Results on dynamic objects. To adapt the EventPS model to the dynamic objects in real-world scenarios, we add exponentially decreasing weights on all the null space vectors to prioritize the latest events. In Fig. 10, we show real-time PS on (a) fingers and (b) rubber toys using our validation platform (rotating at 1800 rpm full speed for lowest latency). We can see the fine-grained details like fingerprint and rubber deformation in real-time[7], which demonstrates the superiority of EventPS in recovering fine-grained details. The processing speeds of EventPS algorithms are over 1000 fps (for EventPS-OP), about 2 fps (for EventPS-FCN), and about 0.1 fps (for EventPS-CNN).

[7]Please refer to the video in supplementary material for full animation.

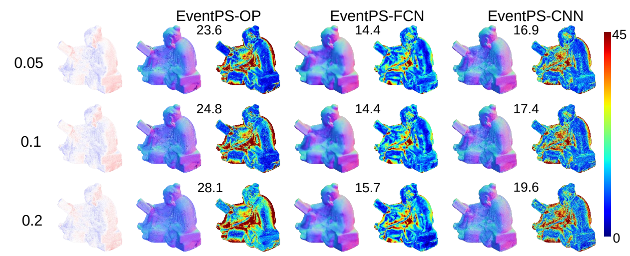

Figure 11. Results on DiLiGenT-Ev dataset with different level of noises. The mean event triggering threshold is 0.15, and the standard deviations are 0.05, 0.1, and 0.2.

5. Conclusion and Discussion

In this paper, we propose EventPS, a novel real-time PS approach using a single event camera. Our method demonstrates the remarkable advantages of speed and data efficiency, which shows great potential to extend the capability for real-time sensing in the dynamic scenes and rapid measurement of the object surface normal.

Robustness to event noise. In both optimization and deep-learning-based methods, there are designs concerning noise robustness: We collect events from a sliding window and aggravate them with SVD (for EventPS-OP in Eq. (10)) or sum them up as neural network input (for EventPS-FCN in Fig. 5 and EventPS-CNN in Fig. 6). In this way, the noise in each pixel is reduced. During the training stage of the two deep-learning methods. By adding event triggering noise with the variable noise levels, we conduct hyperparameter analysis experiment about noise level in Fig. 11 to demonstrate the robustness of our method. All three EventPS algorithms are robust as the noise level increases.

Limitation. Firstly, the scanning patterns of lighting have their limitations: the "circle" pattern leaves a blind area for high elevation angle surface normal, and the "hypotrochoid" pattern is difficult to implement mechanically. Secondly, as the scanning speed of lighting increases, the quality of event signals gradually degrades due to frequency response [6]. Achieving diverse scanning patterns, implementing non-mechanical illumination devices, and improving event signal quality under high-speed illumination is worth exploring as further work.

Acknowledgement

This work is supported by National Natural Science Foundation of China (Grant No. 62136001, 62088102, 62302019).

References

[1] Anish Bhattacharya, Ratnesh Madaan, Fernando Cladera, Sai Vemprala, Rogerio Bonatti, Kostas Daniilidis, Ashish Kapoor, Vijay Kumar, Nikolai Matni, and Jayesh K Gupta. EvDNeRF: Reconstructing event data with dynamic neural radiance fields. In *Proc. of IEEE Winter Conf. App. Comput. Vis.*, 2023. 3

[2] Guanying Chen, Kai Han, and Kwan-Yee K. Wong. PS-FCN: A flexible learning framework for photometric stereo. In *Proc. of Eur. Conf. Comput. Vis.*, 2018. 2, 5, 6, 7

[3] Guanying Chen, Kai Han, Boxin Shi, Yasuyuki Matsushita, and Kwan-Yee K Wong. Self-calibrating deep photometric stereo networks. In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[4] Per H. Christensen and Linda G. Shapiro. Three-dimensional shape from color photometric stereo. *Int. J. Comput. Vis.*, 1994. 2

[5] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proc. of Conf. on Comput. Graph. and Interactive Tech.*, 2000. 2

[6] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 2, 3, 8

[7] Gene H. Golub and Christian H. Reinsch. Singular value decomposition and least squares solutions. In *Milestones in Matrix Computation - Selected Works of Gene H. Golub, with Commentaries*. 2007. 2

[8] Heng Guo, Fumio Okura, Boxin Shi, Takuya Funatomi, Yasuhiro Mukaigawa, and Yasuyuki Matsushita. Multispectral photometric stereo for spatially-varying spectral reflectances: A well posed problem? In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 2, 3

[9] Jin Han, Yuta Asano, Boxin Shi, Yinqiang Zheng, and Imari Sato. High-fidelity event-radiance recovery via transient event frequency. In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 3

[10] Carlos Hernandez, George Vogiatzis, Gabriel J. Brostow, Bjorn Stenger, and Roberto Cipolla. Non-rigid photometric stereo with colored lights. In *Proc. of Int. Conf. Comput. Vis.*, 2007. 2

[11] Kunping Huang, Sen Zhang, Jing Zhang, and Dacheng Tao. Event-based simultaneous localization and mapping: A comprehensive survey. *arXiv preprint arXiv:2304.09793*, 2023. 3

[12] Inwoo Hwang, Junho Kim, and Young Min Kim. Ev-NeRF: Event based neural radiance field. In *Proc. of IEEE Winter Conf. App. Comput. Vis.*, 2023. 3

[13] Satoshi Ikehata. CNN-PS: CNN-based photometric stereo for general non-convex surfaces. In *Proc. of Eur. Conf. Comput. Vis.*, 2018. 2, 5, 6, 7

[14] Satoshi Ikehata. PS-Transformer: Learning sparse photometric stereo network using self-attention mechanism. In *Proc. of Brit. Mach. Vis. Conf.*, 2021. 3

[15] Satoshi Ikehata. Universal photometric stereo network using global lighting contexts. In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 3

[16] Satoshi Ikehata. Scalable, detailed and mask-free universal photometric stereo. In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 3

[17] Katsushi Ikeuchi. Determining surface orientations of specular surfaces by using the photometric stereo method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1981. 2

[18] Micah K Johnson and Edward H Adelson. Shape estimation in natural illumination. In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2011. 6

[19] Yakun Ju, Xinghui Dong, Yingyu Wang, Lin Qi, and Junyu Dong. A dual-cue network for multispectral photometric stereo. *Pattern Recognition*, 2020. 3

[20] Yakun Ju, Kin-Man Lam, Yang Chen, Lin Qi, and Junyu Dong. Pay attention to devils: A photometric stereo network for better details. In *Proc. of Int. Joint Conf. on AI*, 2021. 3

[21] Yakun Ju, Kin-Man Lam, Wuyuan Xie, Huiyu Zhou, Junyu Dong, and Boxin Shi. Deep learning methods for calibrated photometric stereo and beyond: A survey. *arXiv preprint arXiv:2212.08414*, 2022. 3

[22] Yakun Ju, Boxin Shi, Muwei Jian, Lin Qi, Junyu Dong, and Kin-Man Lam. Normattention-PSN: A high-frequency region enhanced photometric stereo network with normalized attention. *Int. J. Comput. Vis.*, 2022. 3

[23] Dongyeop Kang, Yu Jin Jang, and Sangchul Won. Development of an inspection system for planar steel surface using multispectral photometric stereo. *Optic. Eng.*, 2013. 2, 3

[24] Hyeongwoo Kim, Bennett Wilburn, and Moshe Ben-Ezra. Photometric stereo for dynamic surface orientations. In *Proc. of Eur. Conf. Comput. Vis.*, 2010. 3

[25] Leonid L Kontsevich, AP Petrov, and IS Vergelskaya. Reconstruction of shape from shading in color images. *J. Opt. Soc. Am. A*, 1994. 2, 3

[26] Junxuan Li and Hongdong Li. Self-calibrating photometric stereo by neural inverse rendering. In *Proc. of Eur. Conf. Comput. Vis.*, 2022. 3

[27] Zongrui Li, Qian Zheng, Boxin Shi, Gang Pan, and Xudong Jiang. DANI-Net: Uncalibrated photometric stereo by differentiable shadow handling, anisotropic reflectance modeling, and neural inverse rendering. In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 3

[28] Fotios Logothetis, Ignas Budvytis, Roberto Mecca, and Roberto Cipolla. PX-Net: Simple and efficient pixel-wise training of photometric stereo networks. In *Proc. of Int. Conf. Comput. Vis.*, 2021. 3

[29] Weng Fei Low and Gim Hee Lee. Robust e-NeRF: NeRF from sparse & noisy events under non-uniform motion. In *Proc. of Int. Conf. Comput. Vis.*, 2023. 3

[30] Jipeng Lv, Heng Guo, Guanying Chen, Jinxiu Liang, and Boxin Shi. Non-lambertian multispectral photometric stereo via spectral reflectance decomposition. In *Proc. of Int. Joint Conf. on AI*, 2023. 2

[31] Qi Ma, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Deformable neural radiance fields using RGB and event cameras. In *Proc. of Int. Conf. Comput. Vis.*, 2023. 3

[32] Tom Malzbender, Bennett Wilburn, Dan Gelb, and Bill Ambrisco. Surface Enhancement Using Real-Time Photometric Stereo and reflectance transformation. In *Proc. of Eurographics Symposium on Rendering Techniques*, 2006. 2

[33] Nathan Matsuda, Oliver Cossairt, and Mohit Gupta. MC3D: Motion contrast 3D scanning. In *Proc. of Int. Conf. Computational Photography*, 2015. 2, 3

[34] Manasi Muglikar, Guillermo Gallego, and Davide Scaramuzza. ESL: Event-based structured light. In *Proc. of Int. Conf. 3D Vis.*, 2021. 2, 3

[35] Manasi Muglikar, Leonard Bauersfeld, Diederik Paul Moeys, and Davide Scaramuzza. Event-based shape from polarization. In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 3

[36] Giljoo Nam and Min H Kim. Multispectral photometric stereo for acquiring high-fidelity surface normals. *IEEE Computer Graphics and Applications*, 2014. 2, 3

[37] Yunshan Qi, Lin Zhu, Yu Zhang, and Jia Li. E2NeRF: Event enhanced neural radiance fields from blurry images. In *Proc. of Int. Conf. Comput. Vis.*, 2023. 3

[38] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time. *Int. J. Comput. Vis.*, 2018. 2, 3

[39] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: an open event camera simulator. In *Proc. of Conf. on Robotics Learning*, 2018. 6

[40] Jieji Ren, Feishi Wang, Jiahao Zhang, Qian Zheng, Mingjun Ren, and Boxin Shi. DiLiGenT10^2: A photometric stereo benchmark dataset with controlled shape and material variation. In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2, 6

[41] Viktor Rudnev, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. EventNeRF: Neural radiance fields from a single colour event camera. In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 3

[42] Wonjeong Ryoo, Giljoo Nam, Jae-Sang Hyun, and Sangpil Kim. Event fusion photometric stereo network. *Neural Networks*, 2023. 3

[43] Hiroaki Santo, Masaki Samejima, Yusuke Sugano, Boxin Shi, and Yasuyuki Matsushita. Deep photometric stereo network. In *Proc. of Int. Conf. Comput. Vis. Worksh.*, 2017. 2

[44] Boxin Shi, Zhe Wu, Zhipeng Mo, Dinglong Duan, Sai-Kit Yeung, and Ping Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. In *Proc. of IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 5

[45] Boxin Shi, Zhipeng Mo, Zhe Wu, Dinglong Duan, Sai-Kit Yeung, and Ping Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019. 2, 5, 6, 7

[46] Melvyn L. Smith and Lyndon N. Smith. Dynamic photometric stereo - a new technique for moving surface analysis. *Image and Vision Computing*, 2005. 3

[47] Tsuyoshi Takatani, Yasuyuki Matsushita, Stephen Lin, Yasuhiro Mukaigawa, and Yasushi Yagi. Enhanced photometric stereo with multispectral images. In *Proc. of Int. Conf. on Mach. Vis. App.*, 2013. 2, 3

[48] Tatsunori Taniai and Takanori Maehara. Neural inverse rendering for general reflectance photometric stereo. In *Proc. of Int. Conf. on Machine Learning*, 2018. 3

[49] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Trans. Graph.*, 28(5):174, 2009. 2

[50] Feishi Wang, Jieji Ren, Heng Guo, Mingjun Ren, and Boxin Shi. DiLiGenT-Pi: Photometric stereo for planar surfaces with rich details-benchmark dataset and beyond. In *Proc. of Int. Conf. Comput. Vis.*, 2023. 6

[51] Ziyun Wang, Kenneth Chaney, and Kostas Daniilidis. EvAC3D: From event-based apparent contours to 3D models via continuous visual hulls. In *Proc. of Eur. Conf. Comput. Vis.*, 2022. 3

[52] Olivia Wiles and Andrew Zisserman. SilNet: Single-and multi-view reconstruction by learning from silhouettes. In *Proc. of Brit. Mach. Vis. Conf.*, 2017. 6

[53] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 1980. 1, 2

[54] Zhuokun Yao, Kun Li, Ying Fu, Haofeng Hu, and Boxin Shi. GPS-Net: Graph-based photometric stereo network. In *Proc. of Adv. Neural Inform. Process. Syst.*, 2020. 3

[55] Qian Zheng, Yiming Jia, Boxin Shi, Xudong Jiang, Ling-Yu Duan, and Alex C Kot. SPLINE-Net: Sparse photometric stereo through lighting interpolation and normal estimation networks. In *Proc. of Int. Conf. Comput. Vis.*, 2019. 2

[56] Qian Zheng, Boxin Shi, and Gang Pan. Summary study of data-driven photometric stereo methods. *Virt. Reality & Intell. Hardware*, 2020. 2, 5

EventPS: Real-Time Photometric Stereo Using an Event Camera Supplementary Material

Bohan Yu[1,2] Jieji Ren[3] Jin Han[4,5] Feishi Wang[1,2] Jinxiu Liang[1,2] Boxin Shi[1,2*]

[1] National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University

[2] National Engineering Research Center of Visual Technology, School of Computer Science, Peking University

[3] School of Mechanical Engineering, Shanghai Jiao Tong University

[4] Graduate School of Information Science and Technology, The University of Tokyo [5] National Institute of Informatics

{ybh1998, wangfeishi, cssherryliang, shiboxin}@pku.edu.cn

jiejiren@sjtu.edu.cn, jinhan@nii.ac.jp

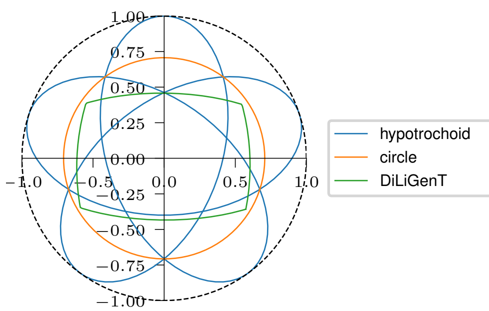

Figure 12. Three types of scanning patterns used in our experiments. All the "circle", "hypotrochoid", and "DiLiGenT" patterns are used on the synthetic dataset for training.

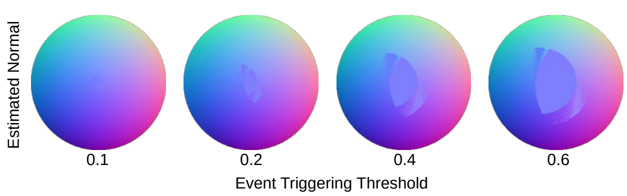

Figure 13. Normal estimation results with blind area (at the center of each sphere). As the event triggering threshold increases, the size of the blind area will also become larger.

6. Scanning Pattern and Blind Area

According to Sec. 4.1, we implement 3 scanning patterns for illumination. These scanning patterns are shown in Fig. 12. We chose the "circle" scanning pattern in our real validation platform for its mechanical feasibility. However, a blind area issue exists in this pattern.

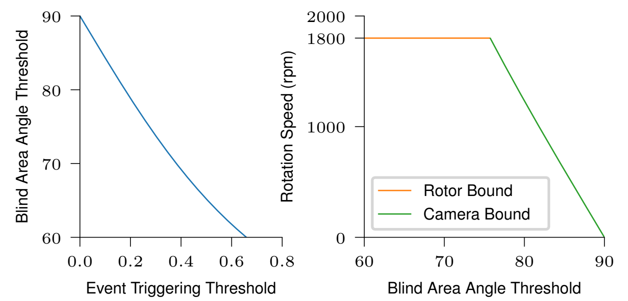

Figure 14. Blind area simulation results for a sphere object. Left: Blind area angle threshold w.r.t. event triggering threshold. Right: Rotation speed w.r.t. blind area angle threshold. Under limited event triggering rate, reducing the rotation speed allows us to set a lower event triggering threshold, achieving a higher blind area angle threshold (smaller blind area size) and better normal estimation sensitivity.

Blind area. According to Fig. 4, using the "circle" scanning pattern, the radiance change becomes smaller as the elevation angle of the surface normal increases. When the radiance change is smaller than the triggering threshold of the event camera, no event is triggered. In this situation, we can only infer that the elevation angle is above a specific threshold. However, we cannot determine the exact normal direction. We call the part of the surface under this situation the blind area. As shown in Fig. 13, there is a blind area at the center of each sphere, where the normal elevation angle is above a threshold θ_t. The size of the blind area is related to the event triggering threshold. The elevation angle threshold θ_t of the blind area is the solution of the following equation:

$$\cos(\theta_t - \theta_{\mathbf{L}}) = \exp(2C)\cos(\pi - \theta_t - \theta_{\mathbf{L}}), \qquad (12)$$

*Corresponding author: Boxin Shi

where $\theta_{\mathbf{L}}$ is the elevation angle of the light source, and C is the event triggering threshold. The brightest reflection $(\cos(\theta_t - \theta_{\mathbf{L}}))$ and the dimmest reflection $(\cos(\pi - \theta_t - \theta_{\mathbf{L}}))$ differ by $2C$, where 2 positive events (and 2 negative events) are triggered, which is the minimal requirement to satisfy the condition to solve Eq. (10). When the elevation angle of the surface point θ is greater than the threshold θ_t, fewer than 2 null space vectors are generated, resulting in the blind area with unsolvable surface normal. We show an example simulation result in Fig. 14. From the left figure, we can see that the blind area elevation angle threshold decreases as we decrease the event triggering threshold. The more advanced "hypotrochoid" does not suffer from the blind area issue. Thanks to the varying lighting elevation angle design, events are triggered even with a flat plane. However, it is more difficult to implement on the real validation platform.

PS quality w.r.t. rotation speed. According to Sec. 4.2, we limit the light source rotation speed to 240 rpm in real dataset capturing for better quality. To reduce the blind area size, we need to decrease the event-triggering threshold. As a result, more events are triggered, reaching the event rate upper bound of the camera (100 M events per second). There are two limitations on rotation speed: rotor bound (It is unsafe to push this rotor beyond 1800 rpm. Otherwise, the high-speed moving parts may cause injury to the experiment operator if they rupture.), and camera bound (This camera generates at most 100 M events per second). In Fig. 14, we show a simulation result with sphere in the right figure. The rotation speed requires to be decreased for higher blind area elevation angle threshold (smaller bind area size) to prevent event dropping. We limit the rotation speed in compensation for a better normal estimation sensitivity. The theoretical blind area elevation angle threshold is about 87° at 240 rpm.

7. FramePS trilemma

In the photometric stereo experimental setup, the data rate of a frame-based camera is the joint effect of three terms: the number of exposure bracketing images, the number of light directions per round, and the light scanning speed. However, the frame-based camera can never achieve optimal configuration of these three terms at the same time due to its limited bandwidth. As shown in Fig. 15, we analyze the trilemma of frame-based camera in details: (TL1) Capturing a significant number of exposure bracketing images along with various light directions per round is time-consuming. Consequently, object movement during this period results in motion blur in the recovered normal map. (TL2) To balance a good number of light directions per round and scanning speed, we have to disable exposure bracketing. With the limited dynamic range, the result would be affected by surface albedo. (TL3) To achieve efficient scanning speed and exposure bracketing, the number of light directions per round must be reduced. As a result, the quality of the estimated normal maps degrades for the lack of information. When a frame-based camera tries to optimize two of these factors, it has to compromise on the third one, which affects the quality of the estimated normal maps.

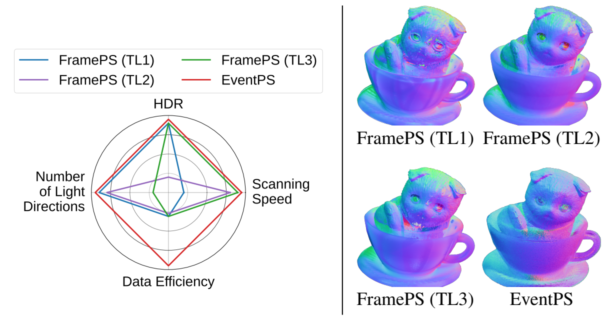

Figure 15. Left: Performance comparison between FramePS under three configurations (TL1 - TL3) and EventPS. Right: Estimated normal maps for all cases. The results demonstrate the trilemma of FramePS: (TL1) scanning speed, (TL2) HDR, and (TL3) number of light directions cannot be fulfilled simultaneously. In contrast, the proposed EventPS satisfies all three criteria with the best bandwidth efficiency.

In our EventPS, the HDR advantage and compression capability of an event camera allow us to fulfill all three criteria while maintaining bandwidth efficiency. Therefore, compared to FramePS, EventPS shows more advantages in practical scenarios.

8. Dynamic Scene Validation

Validation platform. In Fig. 16, we present a detailed depiction of the equipment utilized for dynamic scene validation. We transfer the rotation of the DC motor to a synchronized hollow drum rotor using a timing belt. In this way, we can place the event camera view point in the middle of the scanning pattern to observe the object through the central hole. Most of the frames and parts are 3D printed and the corresponding 3D mesh files will be released upon the acceptance of this paper.

Dynamic scene video. The video is available as a separate file named "EventPS_supp_video.mp4". During the dynamic scene real-time demo, we set the parameters of the event camera as follows: The "bandwidth bias (bias_fo)" is set to -35, the two "contrast sensitivity threshold biases" are both set to -10, and the "dead time bias" is set to -20.

9. Complete Evaluation Results

We show all the objects used in our experiments in Fig. 17 and the complete estimated normal map and error map in

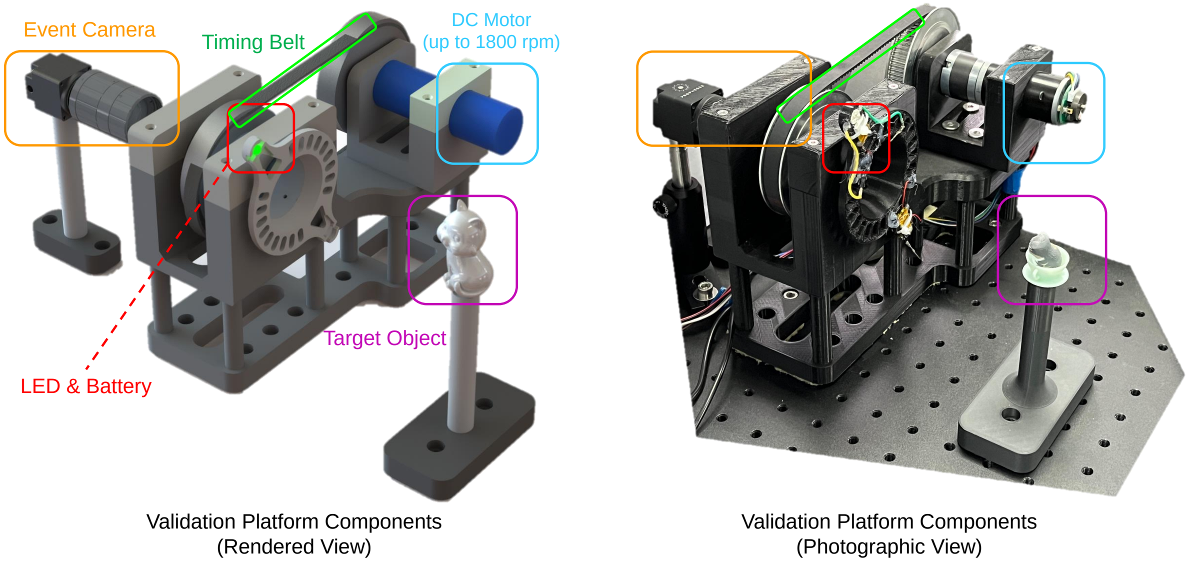

Figure 16. Components of the proposed validation platform. The rendered image (left) aligns with the viewpoint of the photographed real platform (right) and is consistent with the supplementary video.

Figure 17. All the objects used in our experiments. Upper: Static objects in the real dataset for quantitative experiments. Lower: Rotating or deformable objects used for dynamic scene qualitative experiments.

Fig. 18 (on DiLiGenT-Ev semi-real dataset) and Fig. 19 (on the real dataset).

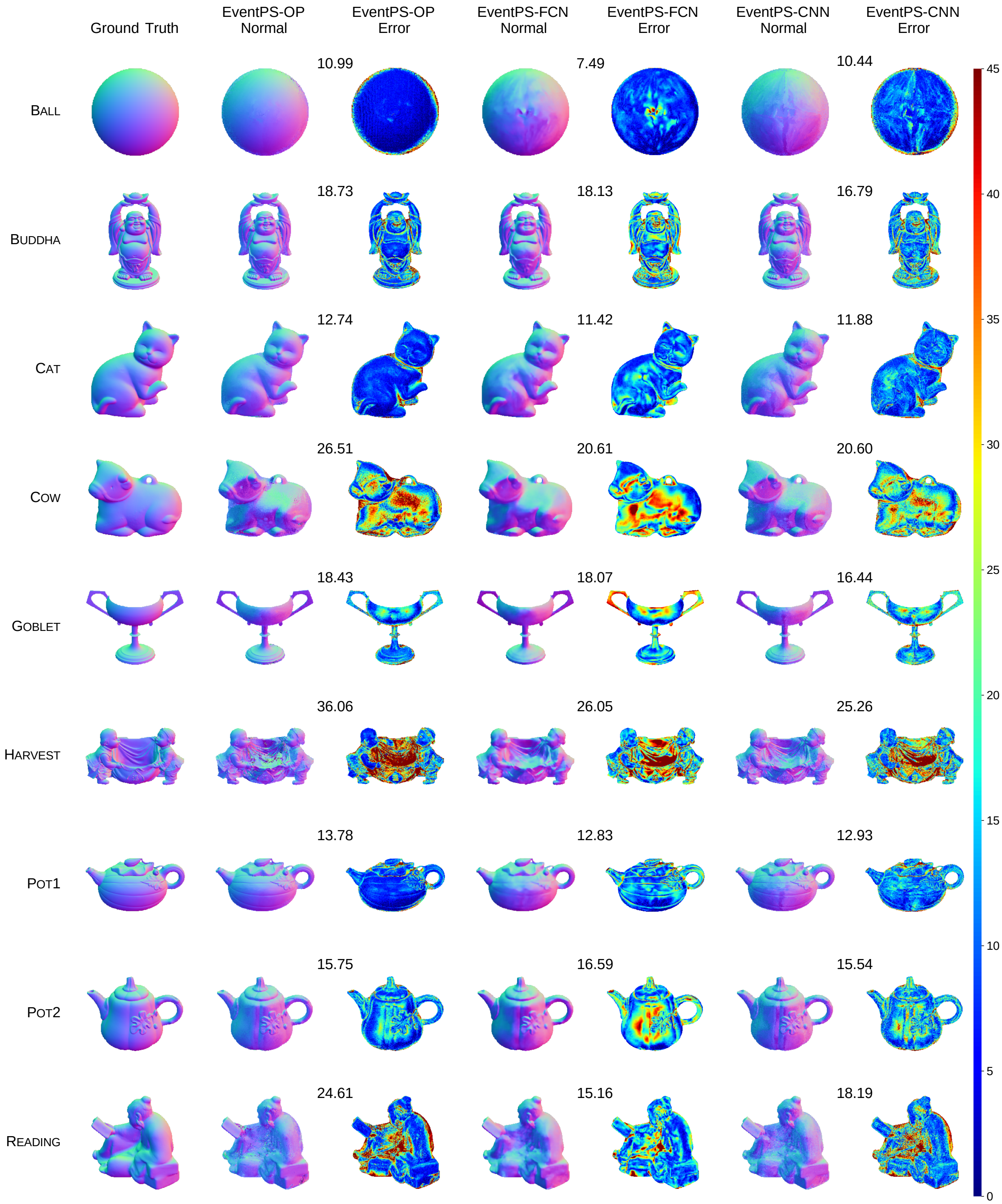

Figure 18. Complete evaluation results on the DiLiGenT-Ev semi-real dataset. The results demonstrate consistent and stable performance among all objects of our EventPS algorithms.

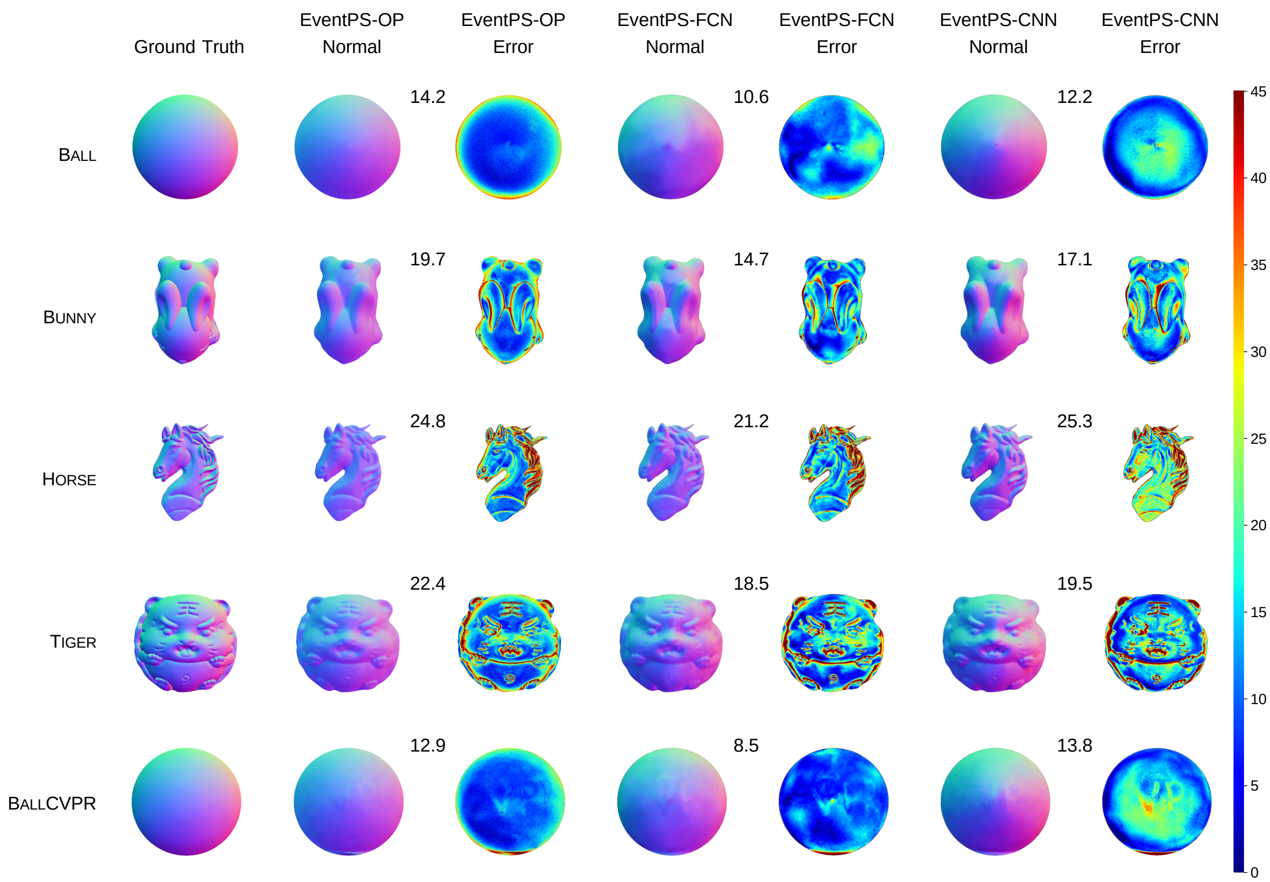

Figure 19. Complete evaluation results on the real dataset. The results demonstrate consistent and stable performance among all objects of our EventPS algorithms.