Spike Transformer: Monocular Depth Estimation for Spiking Camera

Jiyuan Zhang[1,*], Lulu Tang[2,3,*], Zhaofei Yu[1,†], Jiwen Lu[3], and Tiejun Huang[1,2]

[1] Department of Computer Science, Peking University
[2] Beijing Academy of Artificial Intelligence
[3] Department of Automation, Tsinghua University
jyzhang@stu.pku.edu.cn, {lulutang, lujiwen}@tsinghua.edu.cn, {yuzf12, tjhuang}@pku.edu.cn
[*]Joint First Authors, [†]Corresponding Author

Abstract. Spiking camera is a bio-inspired vision sensor that mimics the sampling mechanism of the primate fovea, which has shown great potential for capturing high-speed dynamic scenes with a sampling rate of 40,000 Hz. Unlike conventional digital cameras, the spiking camera continuously captures photons and outputs asynchronous binary spikes that encode time, location, and light intensity. Because of the different sampling mechanisms, the off-the-shelf image-based algorithms for digital cameras are unsuitable for spike streams generated by the spiking camera. Therefore, it is of particular interest to develop novel, spike-aware algorithms for common computer vision tasks. In this paper, we focus on the depth estimation task, which is challenging due to the natural properties of spike streams, such as irregularity, continuity, and spatial-temporal correlation, and has not been explored for the spiking camera. We present Spike Transformer (Spike-T), a novel paradigm for learning spike data and estimating monocular depth from continuous spike streams. To fit spike data to Transformer, we present an input spike embedding equipped with a spatio-temporal patch partition module to maintain features from both spatial and temporal domains. Furthermore, we build two spike-based depth datasets. One is synthetic, and the other is captured by a real spiking camera. Experimental results demonstrate that the proposed Spike-T can favorably predict the scene's depth and consistently outperform its direct competitors. More importantly, the representation learned by Spike-T transfers well to the unseen real data, indicating the generalization of Spike-T to real-world scenarios. To our best knowledge, this is the first time that directly depth estimation from spike streams becomes possible. Code and Datasets are available at https://github.com/Leozhangjiyuan/MDE-SpikingCamera.

Keywords: Depth estimation, Transformer, Spiking camera, Spike data

1 Introduction

Traditional frame-based cameras work at a fixed rate, providing stroboscopic synchronous sequences of images by a snapshot. The concept of the exposure time window in frame-based cameras constrains their usage in some challenging scenarios, such as high-speed scenes and high dynamic range environment, leading to motion blur or over/under exposure. Compared with those cameras, the spiking camera [9,10,28], a bio-inspired visual sensor, poses a radically different sensing modality. Instead of capturing the visual signal in an exposure interval by a snapshot, each pixel on a spiking camera sensor independently and persistently captures the incoming photons, and triggers a spike only when the accumulated photons reach a dispatch threshold. Thus the spiking camera can produce a continuous spike stream at very high temporal resolution. Those recorded spatio-temporal spike streams can be used to reconstruct the dynamic scenes at any given moment [67,73]. Different from event-based camera (also called dynamic vision sensor) [15,16,38,53] that only records the relative brightness changes at each pixel, the spiking camera records the absolute light intensity, providing both static and dynamic scene information. Benefiting from the superior properties, such as full-time imaging and free dynamic range, the spiking camera poses enormous potential in autonomous driving, unmanned aerial vehicles, and mobile robots.

Depth estimation is a fundamental task in computer vision. State-of-the-art depth prediction works concentrate more on the standard frame-based cameras [5,14,21,23,33,62]. Recently, event-based depth estimation has made significant progress [18,26,48,70,71,72]. However, there is no investigation related to depth prediction for the spiking camera. Due to the different sampling mechanisms, the off-the-shelf depth estimation models for traditional images that only record stationary scenes are unsuitable for spike streams generated by the spiking camera. Learning depth from the asynchronous spike streams poses several challenges: 1) Lack of unified backbone for spike data: within a binary and irregular data structure, continuous spike streams capture dynamic scenes at a very high temporal resolution. There is no standard network at hand that can simultaneously mine the spatial and temporal features from the dense spike streams. 2) Lack of spike depth dataset: There is no well-annotated dataset containing spike streams and the corresponding ground truth depth. It is rather sophisticated to calibrate the imaging windows and synchronize the timestamps between spiking and depth cameras.

Inspired by prior works [1,4,41,60] that utilize Transformer [55] to model spatio-temporal correlations for videos, we attempt to explore Transformer to learn the spatio-temporal features from the irregular spike data. Transformer has been successfully applied in NLP [8,6,30,45], images [2,11,20,40], and point cloud [24,63,64,66], but very little is known about its effectiveness in binary spike data. A naive way is to convert spike streams to videos composed of sequential intensity frames so that the well-developed image-based algorithms can be used to learn the spike streams. However, when a high-temporal spike stream (40,000 HZ) is converted to typical frequency images (30 FPS), the converted images will lose some temporal continuity. When spikes are transformed to images with the same frequency (40,000 FPS), the temporal information can be preserved but with a surge of computational cost.

This work focuses on dense, monocular depth estimation (MDE) from original spike streams. Two key points are investigated: 1) How to mine the spatio-temporal features from binary, irregular, and continuous spike streams? 2) How to make full use of Transformer on the unstructured spike data? At this point, a new scheme, named Spike Transformer (Spike-T), is proposed to learn both spatial and temporal spike features and subsequently estimate depth from continuous spike streams. To our best knowledge, this is the first attempt to predict depth using only spike streams. In order to unleash the potential of the spiking camera in high-speed depth estimation, we first collect and generate one synthetic dataset, denoted as 'DENSE-spike' (see Section.5.1), which comprises spike streams, and the corresponding ground truth depth maps. We further collect a real dataset named 'Outdoor-spike' using the spiking camera [28], which includes various scenes of traffic roads and city streets.

Experimental results show that the proposed Spike-T performs well on our synthetic dataset and reliably predicts depth maps on the unseen real data. In summary, our main contributions include

- We dedicate to monocular depth estimation from continuous spike streams for the first time. One synthesized and one real captured spike-based depth datasets are first developed.
- We propose Spike Transformer (Spike-T), which adopts a spatio-temporal Transformer architecture to learn the unstructured spike data, mining the spatio-temporal characteristics of spike streams.
- To fit spike data to Transformer, we present an input spike embedding equipped with a spatio-temporal patch partition module to maintain features from both spatial and temporal domains.
- Qualitative and quantitative evaluations on the synthetic dataset demonstrate that the proposed Spike-T reliably predicts the scene's depth, and the representation learned by Spike-T transfers well to the unseen real data, indicating the generalization of the proposed model to the real scenarios.

2 Related Works

2.1 Bio-inspired Spiking Camera.

The spiking camera [9,10], also called Vidar camera [28], is a bio-inspired vision sensor that mimics the sampling mechanism of the primate fovea, achieving 1000× faster speed than conventional frame-based counterparts. Due to its distinct working principles, the spiking camera can continuously record the scene's texture theoretically. Given its huge potential in many applications, such as traffic surveillance and suspect identification, spike-based vision tasks have been rapidly investigated. By counting the time interval of spikes, Dong et al. [10] first provided an efficient coding method for spiking camera. Motivated by bio-realistic leaky integrate-and-fire (LIF) neurons and synapse connection with spike-timing-dependent plasticity (STDP) rules, Zhu et al. [74] constructed a three-layer spiking neural network (SNN) to reconstruct high-quality visual images of natural scenes. Zheng et al. [69] introduced an image reconstruction model through the short-term plasticity(STP) mechanism of the brain. Zhao et al. [68] built a hierarchical CNN architecture to reconstruct dynamic scenes, exploiting the temporal correlation of the spike stream progressively. More recently, [27] presented a deep learning pipeline to estimate optical flow from continuous spike streams, where the predicted optical flow was able to alleviate motion blur. Prior works have made significant progress in developing spiking cameras. Nevertheless, one of the essential vision tasks, depth estimation, has not been fully considered. This work thus focuses on learning depth from spike streams.

2.2 Image-based and Event-based Monocular Depth Estimation.

Image-based monocular depth estimation aims to generate a dense depth map containing 3D structure information from a single-view image. Early works on image-based depth prediction primarily based on handcrafted features related to pictorial depth cues, such as texture density and object size [50]. In more recent years, deep learning-based depth estimation models have gained traction [13,22,23,34,36,39,44,56,62]. They commonly exploit an encoder-decoder architecture with skip-connections to learn depth-related priors directly from training data, achieving impressive depth estimation performance compared to the handcrafted counterparts.

Recently, event-based monocular depth estimation has drawn increasing attention due to its unique properties [3,7,17,19,25,26,31,47,72], especially for high-speed scenes where low-latency obstacle avoidance and rapid path planning are critical. Gallego et al. [17] developed a unifying contrast maximization framework to solve several event-based vision problems, such as depth prediction and optical flow estimation, by finding the point trajectories on the image plane that are best aligned with the event-based data. Zhu et al. [72] presented a proper event representation in the form of a discretized volume and utilized an encoder-decoder mechanism to integrate several cues from the event streams. Recurrent convolutional neural networks were exploited in [26] to learn monocular depth by leveraging the temporal consistency presented in the event streams. More recently, Gehrig et al. [19] proposed a Recurrent Asynchronous Multimodal network to estimate monocular depth by combining events and frames, which generalized traditional RNNs to learn asynchronous event-based data from multiple sensors. Prior event-based works have greatly inspired our work. Unlike event-based cameras, which pay more attention to motion edges, the spiking camera captures both stationary and moving objects. Hence, spike-based vision problems need to be studied in different ways from event-based counterparts.

2.3 Transformer for dense prediction.

Self-attention-based models, in particular Transformers [55], have recently become the dominate backbone architecture in natural language processing (NLP) [6,8,30,45]. It also intrigued the vision community [2,11,20,40] due to its salient benefits, including massively parallel computing, long-distance characteristics, and minimal inductive biases. As for dense prediction tasks, Transformer has a global receptive field at every stage and can work at a constant and relatively high resolution. These attractive properties can naturally lead to fine-grained and globally coherent dense predictions [46]. Transformer-based networks have been intensively investigated for dense prediction [35,37,42,57,58,65]. Ranftl et al. [46] applied ViT [11] as the encoder backbone to estimate monocular depth. Compared with CNN backbone, it showed that more coherent predictions could be learned due to the global receptive field of Transformer. Yang et al. [61] additionally used a ResNet projection layer and attention gates in the decoder to induce the spatial locality of CNNs for monocular depth estimation. Lately, Johnston et al. [29] utilized a self-attention block to explore the general contextual information and applied a discrete disparity to regularize the training procedure. More recently, Varma et al. [54] investigated self-supervised monocular depth estimation using vision Transformer. It showed that Transformer achieves comparable performance while being more robust and generalizable when compared with CNN-based architectures. The structural superiority of Transformer has been proved by both NLP and image tasks. Previous dense prediction works also justify the capability of Transformer for depth prediction. Motivated by Swin Transformer [40], we develop a spatio-temporal Transformer network for monocular depth estimation from continuous and unstructured spike streams.

3 Preliminary: Spike Generation Mechanism

Inspired by the sampling mechanism of primate fovea in retina [43,59], the spiking camera records the intensity information with spatio-temporal characteristics. It outputs binary streams in spike format, representing data with only 0 or 1. The spiking camera mainly consists of three ingredients, the photoreceptor, the accumulator, and the comparator. Specifically, an array of photosensitive pixels are spatially arranged on the photoreceptor of spiking cameras, continuously capturing photons. Secondly, the accumulator persistently converts light signals into electrical signals to increase the voltage of each unit. The comparator detects whether the accumulated voltage reaches the dispatch voltage threshold θ. When the threshold is reached, a spike is triggered, and the voltage will be reset to the preset value. To depict the spike generation mechanism, the process on one pixel can be formulated as:

$$\int_{t_{i-1}}^{t_i} \alpha I(t) dt = \theta \tag{1}$$

where $I(t)$ describes the light intensity, t_i and t_{i-1} denote the firing times of the i-th and $(i-1)$-th spikes, respectively. α is the photoelectric conversion rate. Due to the limitations of circuit technology, the unit in the output circuit read out spikes as discrete-time signals $s(x, y, n)$ periodically within a fixed interval $\varDelta t = 25$ us. A spike will be read out $s(x, y, n) = 1$ $(n = 1, 2, \ldots)$ if the pixel at spatial coordinate (x, y) fires a spike at time t, with $(n-1)\varDelta t < t \leqslant n\varDelta t$. Otherwise it reads out $s(x, y, n) = 0$. The sensor uses a high-speed polling to generate a spike frame with size of $H \times W$ at each discrete timestamp n. In a fixed interval $\Delta t \cdot T$, the camera would produce a binary spike stream $S = \{s(x, y, t)\}_{t=1}^{T}$ with size of $H \times W \times T$.

4 Spike Transformer for Monocular Depth Estimation

The spiking camera outputs spikes at each pixel independently and asynchronously. For simplicity, we use $S \in \{0, 1\}^{H \times W \times T}$ to denote a spike stream, and use $\mathcal{D} \in \mathbb{R}^{H \times W}$ to denote the depth map at one polling. The objective of monocular depth estimation for the spiking camera is to predict the original depth map $\mathcal{D}$ from the continuous binary spike stream S. To this end, we present Spike Transformer (Spike-T) for monocular depth prediction. The overall framework comprises three components: (a) spike embedding module, (b) spatio-temporal Transformer (STT) encoder, and (c) CNN-based decoder. The framework is illustrated in Fig. 1, which maintains the overall encoder-decoder architecture with hierarchical structures. Specifically, an input spike stream is first fed into the spike embedding module, obtaining several spike embeddings that preserve spatio-temporal characteristics for Transformer's input. Subsequently, we employ several STT blocks to learn spatio-temporal features for spike embeddings, using the adapted self-attention mechanism. The hierarchical features from the encoder are progressively fused for final depth prediction.

4.1 Spike Embedding

The spike embedding module consists of three steps: Temporal Partition, Feature Extraction, and Spatial Partition (see Fig. 2). As presented in Section 3, a sequence of spike frames $S = \{s(x, y, t)\}_{t=1}^{T}$ record the scene's radiance at each timestamp t. The features along the time axis are crucial to reconstruct the depth map. A multi-scale temporal window is thus introduced to maintain more temporal information.

Specifically, for each raw input S, we first partition it into n non-overlapping spike chunks along the temporal axis, using a sliding window of length $\frac{T}{n}$. Each spike chunk with shape of $H \times W \times \frac{T}{n}$ carries different local temporal features in an interval T. A lightweight feature extractor (FE), consisting of four residual blocks, is then used to project each chunk to a feature map of size $H \times W \times C$. Theoretically, the length of time window $\frac{T}{n}$ can be set as an arbitrary positive integer no more than T. Different time-scale window carries a different scale of temporal information. We thus can leverage multiple time-scale windows, e.g., $\frac{T}{1}, \frac{T}{2}, \frac{T}{4}, ... \frac{T}{n}$, to capture multi-level temporal features. Empirically, we set n to 4. Features from $\frac{T}{4}$ time window can be considered as $\frac{1}{4}$ local features. Subsequently, we set n to 1, features from $\frac{T}{1}$ time window can be seen as global features. Similarly, other time windows can be used. In our setting, only $\frac{T}{4}$ local and $\frac{T}{1}$ global features are considered. In this case, 4 spike chunks are passed

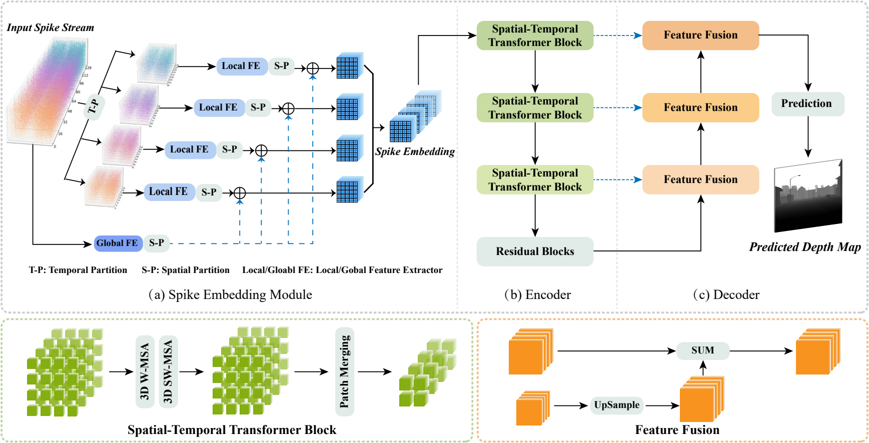


Fig. 1. The framework of Spike-T for MDE. Generally, our model is a U-shaped network consisting of three components: (a) spike embedding module, (b) spatio-temporal Transformer (STT) encoder, and (c) convolutional decoder. We first partition the input spike stream into several non-overlapping chunks by a multi-scale temporal window. Then, a spatial partition layer and Local/Global Feature Extractor (FE) are used to obtain a series of spike embeddings. Our encoder is built by several STT blocks, which implements the attention mechanism along the temporal, height, and width axes. The decoder comprises multiple feature fusion layers, in which hierarchical features are progressively fused and finally used to estimate the depth map.

through the shared local FE module separately, while the full-length spike stream is fed into the global FE module. After that, we split each feature map from FE module into $\frac{H}{2} \times \frac{W}{2}$ patches (with 2×2 patch size) in the spatial domain. By merging local and global FE features, we can obtain $\frac{H}{2} \times \frac{W}{2} \times 4$ temporal-robust feature maps. Therefore, an input spike stream with shape of $H \times W \times T$ can be partitioned into $\frac{H}{2} \times \frac{W}{2} \times 4$ spatio-temporal (ST) blocks. We treat each ST block of size $2 \times 2 \times \frac{T}{4}$ as a token. Following the practice of Transformers in NLP and image-based tasks, we term the feature of those tokens as spike embeddings, which thus can be received as inputs to Transformer.

4.2 Spatio-Temporal Transformer Encoder

The overall architecture of spatio-temporal Transformer is illustrated in Fig. 2, which is adapted from a Swin Transformer architecture [40,41]. Features from spiking embedding module are fed into the Transformer-based encoder, which includes three stages. Each stage consists of 2,2 and 6 STT blocks, respectively. A patch merging layer is added between two adjacent stages.

Spatial patch merging. To preserve more local temporal features, following the prior work [41], we only implement the downsampling operation in the spatial domain, maintaining the number of tokens in the temporal domain. Specifically,

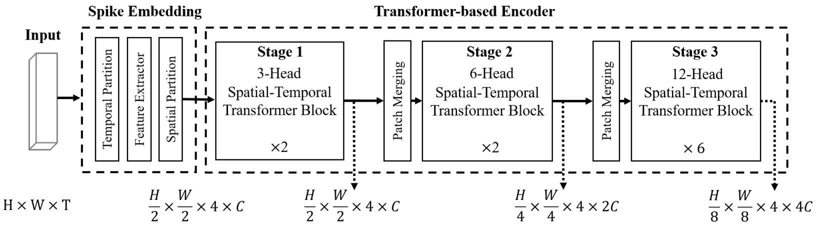


Fig. 2. Architecture of Spike Transformer.

features from each 2×2 spatial neighboring patches are first concatenated along C-channel, forming a merging feature map with $4C$ dimension. A linear layer is used to project such a $4C$-dimensional feature map to a $2C$-dimensional one. Thus, the feature dimension along C-channel will be doubled after each stage.

Spatio-temporal Transformer block. The key component of Spike-T is the STT block. Each STT block consists of a multi-head self-attention (MSA) module equipped with a 3D shifted window, followed by a feed-forward network (FFN) composed of a 2-layer MLP. Between each MSA module and FFN, a GELU layer is utilized, and a Layer Normalization(LN) is used before each MSA and FFN. Each module applies a residual connection.

In particular, for two consecutive STT blocks at one Transformer encoder stage, the MSA module in the former block acts on the $T_x \times H_x \times W_x$ spatio-temporal tokens. 3D windows with size of $W_t \times W_h \times W_w$ are then utilized to evenly partition these tokens into $\lceil \frac{T_x}{W_t} \rceil \times \lceil \frac{H_x}{W_h} \rceil \times \lceil \frac{W_x}{W_w} \rceil$ non-overlapping windows. In our implementation, T_x is set to be same as the temporal partition number($n = 4$). H_x and W_x are the current spatial token size. The 3D window size is set to $2 \times 7 \times 7$. For the MSA module in the latter SST block, we shift windows along the temporal, height, and width axes by (1, 3, 3) tokens from the previous STT block and perform attention among windows. Following the practice in [40,41], we term the first window-based MSA as 3D W-MSA, which uses the regular window partitioning configuration. The second window-based MSA is denoted as 3D SW-MSA, which applies a shifted window partitioning mechanism. The above two successive SST blocks can be formalized as [41]

$$\begin{aligned}
\hat{z}^m &= \text{3DW-MSA}(LN(z^{m-1})) + z^{m-1}, \\
z^m &= \text{FFN}(\text{LN}(\hat{z}^m)) + \hat{z}^m, \\
\hat{z}^{m+1} &= \text{3DSW-MSA}(\text{LN}(z^m)) + z^m, \\
z^{m+1} &= \text{FFN}(\text{LN}(\hat{z}^{m+1})) + \hat{z}^{m+1},
\end{aligned} \tag{2}$$

where $\hat{z}^m$ and z^m denote the output features of the 3D(S)W-MSA module and the FFN module for block m in one stage. As for STT blocks in each stage, we use 3, 6 , and 12 attention heads, respectively. In this way, features from each Transformer encoder stage can be formed as a tuple, with sizes of $\frac{H}{2} \times \frac{W}{2} \times T \times C$, $\frac{H}{4} \times \frac{W}{4} \times T \times 2C$ and $\frac{H}{8} \times \frac{W}{8} \times T \times 4C$ separately, which thus can be used for downstream spike-based tasks.

4.3 Decoder for Depth Prediction

As shown in Fig. 1, our decoder consists of two residual blocks, three feature fusion layers, and one prediction head. The output feature maps, with size of $H_i \times W_i \times T \times C_i$ from each encoder stage, are concatenated along temporal axis, reshaping the size to $H_i \times W_i \times TC_i$. After that, a convolutional layer is used to project the reshaped features back to $H_i \times W_i \times C_i$. The last feature maps, with the size of $\frac{H}{8} \times \frac{W}{8} \times 4C$, are first passed through two residual blocks with a kernel size of 3. Subsequently, features from the previous layer are upsampled through a bilinear interpolation operation and progressively fused with the following layers. A prediction head consisting of one convolutional layer is finally used to generate a $H \times W \times 1$ depth map.

4.4 Loss Function

Following [26], we employ a scale-invariant loss to train our depth estimation network in a supervised manner. For k-th spike stream with the size of $H \times W \times T$, the model outputs the depth map with the size of $H \times W \times 1$. We denote the predicted depth map, ground truth depth map, and their residual as $\hat{\mathcal{D}}_k$, $\mathcal{D}_k$, and $\mathcal{R}_k$, respectively. The scale-invariant loss is then defined as

$$\mathcal{L}_k = \frac{1}{n} \sum_{\mathbf{p}} (\mathcal{R}_k(\mathbf{p}))^2 - \frac{1}{n^2} \left(\sum_{\mathbf{p}} \mathcal{R}_k(\mathbf{p}) \right)^2 , \tag{3}$$

where $\mathcal{R}_k = \hat{\mathcal{D}}_k$ - $\mathcal{D}_k$, and n is the number of valid ground truth pixels $\mathbf{p}$.

5 Experiments

5.1 Dataset

Synthetic Dataset. We train our Spike-T in a supervised fashion, which requires a large-scale training dataset in the form of spike streams and the corresponding synchronous depth maps. Nevertheless, it is complicated to build a real dataset consisting of spike steam, gray image, and the corresponding depth map. Moreover, it is rather sophisticated to calibrate imaging windows and synchronize timestamps among a spiking camera, a frame-based camera, and a depth camera. Thus, we build a synthetic spike dataset. Specifically, we first choose the dataset named DENSE proposed in [26] as our database. The DENSE dataset was generated by CARLA simulator [12], including clear depth maps and intensity frames in 30 FPS under a variety of weather and illumination conditions. To obtain spike streams with a very high temporal resolution, we adopt a video interpolation method [52] to generate intermediate RGB frames between adjacent 30-FPS frames. With absolute intensity information among RGB frames, each sensor pixel can continuously accumulate the light intensity with the spike generation mechanism introduced in Section 3, producing spike streams with a high temporal resolution (128×30 FPS) that is 128 times of the video frame rate. The 'spike' version of DENSE dataset (namely DENSE-spike) contains eight sequences, five for training, two for validation, and one for testing. Each sequence consists 999 samples, and each sample is a tuple of one RGB image, one depth map, and one spike stream. Each spike stream is simulated between two consecutive images, generating a binary sequence with 128 spike frames (with size of 346×260) that depicts the continuous process of dynamic scenes.

Real Dataset. To verify the generalization of the proposed model, we further collect some natural spike sequences using a spiking camera [28]. The spatial resolution of the spiking camera is 400×250, and the temporal resolution is 40000 HZ. The real captured spike streams are recorded on city streets and roads. We denote this real dataset as 'Outdoor-Spike', which is only used for testing due to lack of the corresponding ground truth depth. In our 'Outdoor-Spike' dataset, 33 sequences of outdoor scenes are captured in a driving car from the first perspective. Each sequence contains 20000 spike frames.

5.2 Implementation Details

Depth Representation. Following [26], we convert the original depth $\mathcal{D}_{k,abs}$ into a logarithmic depth map $\mathcal{D}_k$, which can be calculated as

$$\mathcal{D}_k = \frac{1}{\beta} \log \frac{\mathcal{D}_{k,abs}}{\mathcal{D}_{max}} + 1, \tag{4}$$

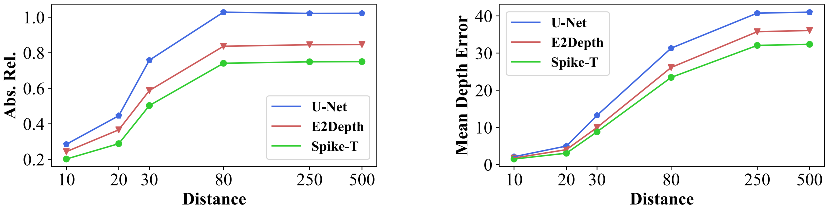
where $\mathcal{D}_{max}$ is the maximum depth in dataset, β is a hyper-parameter empirically set to let the minimum depth closed to 0. For our synthetic dataset 'DENSE-spike', we set $\beta = 5.7$, $\mathcal{D}_{max} = 1000m$.

Training Setup. We implemented the network in PyTorch. In training, we adopt ADAM optimizer [32] to optimize the network and set the initial learning rate λ set to 0.0003. Our model is trained for 200 epochs with a batch size of 16 on 2 NVIDIA A100-SXM4-80GB GPUs. We use the exponential learning rate scheduler to adjust the learning rate after 100^{th} epoch with γ set to 0.5.

Metrics. We adopt several important metrics, including absolute relative error (*Abs Rel.*), square relative error (*Sq Rel.*), mean absolute depth error (MAE), root mean square logarithmic error ($RMSE\ log$) and the accuracy metric ($Acc.\delta$). Detailed formulations can be found in the Appendix.

5.3 Experiment Results

In this section, we evaluate the performance of our Spike-T on both synthetic and real captured datasets, and compare Spike-T with two model architectures, U-Net [49] and E2Depth [26]. Three models are all trained on the synthetic 'DENSE-spike' dataset. To verify the generalization and transferability of our Spike-T, we further utilize the real dataset 'Outdoor-Spike' for testing, and give qualitative visualization results. Finally, an ablation study of Spike-T is presented.



(a) Abs. Rel. Difference (b) Mean Depth Error

Fig. 3. Results of absolute relative difference and mean depth error in different clip distances. Curves in green, red and blue represent our Spike-T, E2Depth [26] and U-Net [49], respectively.

Table 1. Quantitative comparison on the DENSE-spike dataset with UNet [49] and E2Depth [26]. We present results on validation set and test set. ↓ indicates lower is better and ↑ indicates higher is better.

Dataset	**Model**	**Abs Rel** ↓	**Sq Rel** ↓	**RMS log** ↓	**SI log** ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Test Set	U-Net	0.815	27.878	0.777	0.459	0.653	0.725	0.778
	E2Depth	0.674	20.316	0.765	0.441	0.639	0.729	0.789
	Spike-T (Ours)	**0.606**	**18.388**	**0.706**	**0.395**	**0.682**	**0.762**	**0.813**
Val Set	U-Net	0.306	7.101	0.394	0.139	0.833	0.909	0.939
	E2Depth	0.291	5.796	0.411	0.168	0.821	0.894	0.928
	Spike-T (Ours)	**0.262**	**4.703**	**0.364**	**0.125**	**0.850**	**0.913**	**0.944**

A. Qualitative and Quantitative Comparisons. We first compare our Spike-T with two dense prediction networks, namely U-Net and E2Depth. Both them and our Spike-T follow the encoder-decoder architecture with multi-scale fusion manner but utilize different encoding mechanisms. In particular, U-Net employs 2D convolutional layers as its encoder and focuses on spatial feature extraction, while E2Depth applies ConvLSTM [51] layers that combine CNN and LSTM to capture the spatial and temporal features. By contrast, our Spike-T employs transformer-based blocks to learn the spatio-temporal features simultaneously. Thus, both U-Net and E2Depth can be seen as our direct competitors.

Table. 1 reports the quantitative comparison on ‘Dense-spike’ dataset. On both validation and testing sets, the proposed Spike-T consistently outperforms the other two methods on all metrics. Furthermore, our method achieves significant improvement on the metrics of Abs.Rel, which is the most convictive metric in depth estimation tasks. The major difference between the three methods lies in the encoder architecture. These experimental results indicate that our Spike-T with the Transformer-based encoder is more efficient in capturing the spatio-temporal features from irregular, continuous spike streams.

We also evaluate our method at depths of 10m, 20m, 30m, 80m, 250m, 500m. Fig. 3 illustrates how absolute relative error and mean depth error change with depths on validating sequences. The results show that our method performs more accurate depth prediction at all distances, especially at the larger distances.

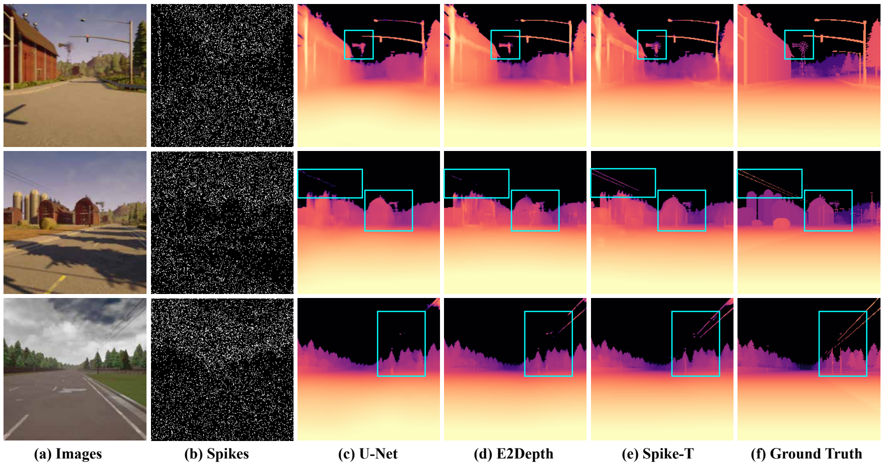


Fig. 4. Visualization results on synthetic dataset 'DENSE-spike'. Boxes in cyan marked in pictures provide comparisons in details.

The qualitative comparison is shown in Fig. 4. For a clear comparison, we mark some cyan boxes on fine-grained objects. As we can see, more details, including tiny structures, sharp edges, and contours, can be estimated by our Spike-T. The quantitative and qualitative results demonstrate that our method is more suitable for continuous spike streams generated by the spiking camera, and can learn valid and robust features from the spatial and temporal domain.

B. Evaluation on Real-World dataset. We evaluate our method by training networks on the synthetic dataset and testing on the real dataset 'Outdoor-Spike'. It is a more challenging dataset captured from outdoor scenes with various motions and noises from the real world. Fig. 5 displays some examples with real spikes, gray images, and the predicted depth compared with the baseline U-Net and E2Depth. The visualization results verify that acceptable depth prediction results are achieved in real-world scenarios.

As shown in Fig. 5(c), depth maps predicted by U-Net and E2Depth include blur artifacts and lose some details, leading to ambiguity between foreground and background. By contrast, depth maps predicted by Spike-T are better with more details of contours, and more precise depth variation can be provided. Overall, despite the domain gap between synthetic and real data, our Spike-T can reasonably predict the real scene's depth, indicating the model's transferability to real-world scenes.

C. Comparison to depth estimation methods using images or events. We give more comparison with depth prediction methods that use images or events. Moreover, we captured some fast-moving and shaky scenes with a synchronized spiking camera and a traditional camera. Visualization results demon-

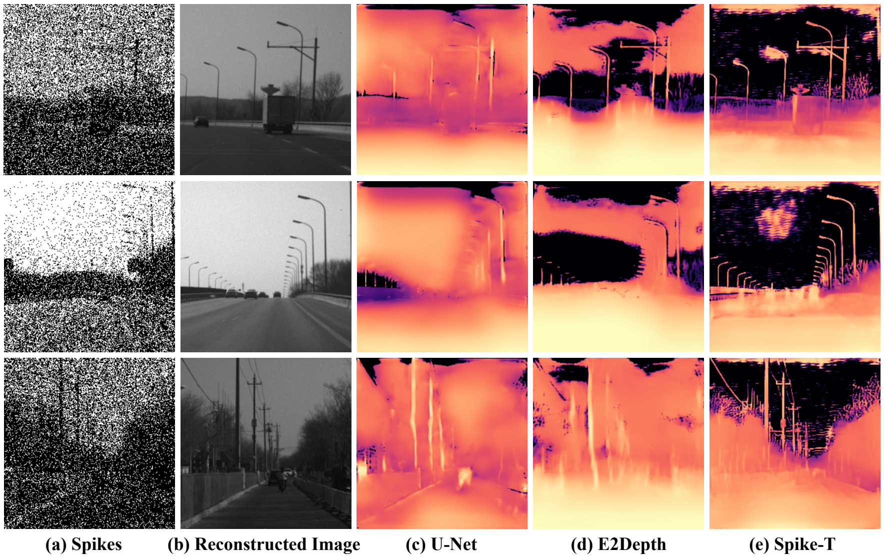


Fig. 5. Visualization results on real-world data (from 'Outdoor-spike'). (a) The spike frame at the predicted timestamp. (b) Reconstructed images with [69]. (c-e) Predicted depth maps with U-Net, E2Depth, and our Spike-T.

Table 2. Ablation study on different temporal window mechanisms.

Model	**Abs Rel** ↓	**Sq Rel** ↓	**RMS log** ↓	**SI log** ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Global Temporal	0.416	11.256	0.478	0.218	0.793	**0.867**	**0.902**
Local Temporal	0.389	10.884	0.490	0.230	0.791	0.860	0.896
Multi-Scale Temporal	**0.376**	**9.265**	**0.478**	**0.215**	**0.794**	0.863	0.901

strate the advantage of spiking cameras over conventional ones under some challenging scenarios (Refer to the Appendix).

5.4 Ablation Studies

A. Effect of multi-scale temporal window. As presented in Section 4.1, we introduce a multi-scale temporal window at the spike embedding stage to preserve more temporal information. To verify its effectiveness, we implement three ablation studies, termed as 'Global Temporal', 'Local Temporal' and 'Multi-scale Temporal', respectively. Specifically, 'Global Temporal' means feature embedding with only global features (temporal partition number $n = 1$), which pays more attention to spatial features. 'Local Temporal' indicates feature embedding using only local features ($n = 4$), while 'Multi-scale Temporal' denotes features embedding from both local ($n = 4$) and global ($n = 1$) features. The comparison results are shown in Table. 2. Spiking embedding with only 'Local Temporal'

Table 3. Ablation results on different patch partitioning manner.

Dataset	**Model**	**Abs Rel** ↓	**Sq Rel** ↓	**RMS log** ↓	**SI log** ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Test Set	3D Partition	0.868	31.771	0.822	0.500	0.636	0.729	0.780
	S-T Partition	**0.606**	**18.388**	**0.706**	**0.395**	**0.682**	**0.762**	**0.813**
Val Set	3D Partition	0.310	7.458	0.386	0.139	0.841	0.907	0.938
	S-T Partition	**0.262**	**4.703**	**0.364**	**0.125**	**0.850**	**0.913**	**0.944**

outperforms that with only ‘Global Temporal’ on the most crucial metric Abs Rel, indicating that the spatio-temporal correlations involved in local features are more informative than that contained in global parts. Furthermore, a multi-scale temporal window combining both local and global features is superior to the above two settings. It demonstrates that more spatio-temporal features can be learned from the unstructured and successive spike streams with the multi-scale temporal window mechanism. More detailed studies on hyperparameter T and n can be found in the Appendix.

B. Effect of S-T patch partition. The patch partitioning operation in the standard Video Swin Transformer is implemented with a 3D convolutional layer. It splits the original input into several 3D blocks. We conduct an ablation study on the patch partitioning manner, comparing our S-T partition with standard Conv3D-based partition by replacing the spike embedding module with a Conv3D layer. The quantitative results are presented in Table. 3. Our method with S-T patch partitioning performs better on all metrics, which indicates that the S-T partition is more suitable to extract features from spike streams.

6 Conclusions

We present Spike Transformer for monocular depth estimation of the spiking camera. To favorably apply Transformer on spike data, an effective spike representation, termed as spiking embedding, is first proposed. Then a modified Swin Transformer architecture is employed to learn the spatio-temporal spike features. Furthermore, two spike-based depth datasets are carefully built. Experiments on both synthetic and real datasets show that our Spike-T can reliably predict the depth maps and express superiority to its direct competitors.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Grant 62176003 and No.62088102.

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6836–6846 (2021)
2. Bao, H., Dong, L., Wei, F.: BEiT: BERT pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)
3. Baudron, A., Wang, Z.W., Cossairt, O., Katsaggelos, A.K.: E3D: Event-based 3D shape reconstruction. arXiv preprint arXiv:2012.05214 (2020)
4. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding. arXiv preprint arXiv:2102.05095 (2021)
5. Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth estimation using adaptive bins. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4009–4018 (2021)
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in Neural Information Processing Systems (NeurIPS) **33**, 1877–1901 (2020)
7. Chaney, K., Zihao Zhu, A., Daniilidis, K.: Learning event-based height from plane and parallax. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR). pp. 0–0 (2019)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Dong, S., Huang, T., Tian, Y.: Spike camera and its coding methods. In: 2017 Data Compression Conference (DCC). pp. 437–437 (2017)
10. Dong, S., Zhu, L., Xu, D., Tian, Y., Huang, T.: An efficient coding method for spike camera using inter-spike intervals. In: 2019 Data Compression Conference (DCC). pp. 568–568. IEEE (2019)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
12. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Conference on Robot Learning. pp. 1–16 (2017)
13. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2015)
14. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2002–2011 (2018)
15. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., et al.: Event-based vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(1), 154–180 (2020)
16. Gallego, G., Gehrig, M., Scaramuzza, D.: Focus is all you need: Loss functions for event-based vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12280–12289 (2019)

17. Gallego, G., Rebecq, H., Scaramuzza, D.: A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3867–3876 (2018)
18. Gehrig, D., Gehrig, M., Hidalgo-Carrió, J., Scaramuzza, D.: Video to events: Recycling video datasets for event cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3586–3595 (2020)
19. Gehrig, D., Rüegg, M., Gehrig, M., Hidalgo-Carrió, J., Scaramuzza, D.: Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction. IEEE Robotics and Automation Letters **6**(2), 2822–2829 (2021)
20. Girdhar, R., Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
21. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 270–279 (2017)
22. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
23. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3828–3838 (2019)
24. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational Visual Media **7**(2), 187–199 (2021)
25. Haessig, G., Berthelon, X., Ieng, S.H., Benosman, R.: A spiking neural network model of depth from defocus for event-based neuromorphic vision. Scientific Reports **9**(1), 1–11 (2019)
26. Hidalgo-Carrió, J., Gehrig, D., Scaramuzza, D.: Learning monocular dense depth from events. In: 2020 International Conference on 3D Vision (3DV). pp. 534–542. IEEE (2020)
27. Hu, L., Zhao, R., Ding, Z., Xiong, R., Ma, L., Huang, T.: Scflow: Optical flow estimation for spiking camera. arXiv preprint arXiv:2110.03916 (2021)
28. Huang, T., Zheng, Y., Yu, Z., Chen, R., Li, Y., Xiong, R., Ma, L., Zhao, J., Dong, S., Zhu, L., et al.: 1000x faster camera and machine vision with ordinary devices. Engineering (2022)
29. Johnston, A., Carneiro, G.: Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4756–4765 (2020)
30. Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: Spanbert: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics **8**, 64–77 (2020)
31. Kim, H., Leutenegger, S., Davison, A.J.: Real-time 3D reconstruction and 6-dof tracking with an event camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 349–364. Springer (2016)
32. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
33. Kopf, J., Rong, X., Huang, J.B.: Robust consistent video depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1611–1621 (2021)

34. Lee, J.H., Kim, C.S.: Monocular depth estimation using relative depth maps. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
35. Lee, Y., Kim, J., Willette, J., Hwang, S.J.: Mpvit: Multi-path vision transformer for dense prediction. arXiv preprint arXiv:2112.11010 (2021)
36. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
37. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1833–1844 (2021)
38. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120db 15μs latency asynchronous temporal contrast vision sensor. IEEE Journal of Solid-state Circuits **43**(2), 566–576 (2008)
39. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
40. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10012–10022 (2021)
41. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. arXiv preprint arXiv:2106.13230 (2021)
42. Liu, Z., Luo, S., Li, W., Lu, J., Wu, Y., Sun, S., Li, C., Yang, L.: Convtransformer: A convolutional transformer network for video frame synthesis. arXiv preprint arXiv:2011.10185 (2020)
43. Masland, R.H.: The neuronal organization of the retina. Neuron **76**(2), 266–280 (2012)
44. Miangoleh, S.M.H., Dille, S., Mai, L., Paris, S., Aksoy, Y.: Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). pp. 9685–9694 (2021)
45. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog (2019)
46. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12179–12188 (2021)
47. Rebecq, H., Gallego, G., Mueggler, E., Scaramuzza, D.: Emvs: Event-based multiview stereo—3D reconstruction with an event camera in real-time. International Journal of Computer Vision **126**(12), 1394–1414 (2018)
48. Rebecq, H., Gallego, G., Scaramuzza, D.: EMVS: event-based multi-view stereo. In: British Machine Vision Conference (BMVC) (2016)
49. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI. pp. 234–241. Springer (2015)
50. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(5), 824–840 (2008)
51. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. Advances in Neural Information Processing Systems **28** (2015)

52. Sim, H., Oh, J., Kim, M.: Xvfi: Extreme video frame interpolation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 14489–14498 (2021)
53. Son, B., Suh, Y., Kim, S., Jung, H., Kim, J.S., Shin, C., Park, K., Lee, K., Park, J., Woo, J., et al.: A 640× 480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation. In: IEEE International Solid-State Circuits Conference (ISSCC). pp. 66–67 (2017)
54. Varma, A., Chawla, H., Zonooz, B., Arani, E.: Transformers in self-supervised monocular depth estimation with unknown camera intrinsics. arXiv preprint arXiv:2202.03131 (2022)
55. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
56. Wang, C., Buenaposada, J.M., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
57. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 568–578 (2021)
58. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8741–8750 (2021)
59. Wässle, H.: Parallel processing in the mammalian retina. Nature Reviews Neuroscience **5**(10), 747–757 (2004)
60. Weng, W., Zhang, Y., Xiong, Z.: Event-based video reconstruction using transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2563–2572 (2021)
61. Yang, G., Tang, H., Ding, M., Sebe, N., Ricci, E.: Transformer-based attention networks for continuous pixel-wise prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16269–16279 (2021)
62. You, Z., Tsai, Y.H., Chiu, W.C., Li, G.: Towards interpretable deep networks for monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12879–12888 (2021)
63. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: Pointr: Diverse point cloud completion with geometry-aware transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12498–12507 (2021)
64. Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J.: Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19313–19322 (2022)
65. Yuan, Y., Fu, R., Huang, L., Lin, W., Zhang, C., Chen, X., Wang, J.: Hrformer: High-resolution transformer for dense prediction. arXiv preprint arXiv:2110.09408 (2021)
66. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16259–16268 (2021)
67. Zhao, J., Xie, J., Xiong, R., Zhang, J., Yu, Z., Huang, T.: Super resolve dynamic scene from continuous spike streams. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2533–2542 (2021)

68. Zhao, J., Xiong, R., Liu, H., Zhang, J., Huang, T.: Spk2imgnet: Learning to reconstruct dynamic scene from continuous spike stream. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11996–12005 (2021)
69. Zheng, Y., Zheng, L., Yu, Z., Shi, B., Tian, Y., Huang, T.: High-speed image reconstruction through short-term plasticity for spiking cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6358–6367 (2021)
70. Zhou, Y., Gallego, G., Rebecq, H., Kneip, L., Li, H., Scaramuzza, D.: Semi-dense 3D reconstruction with a stereo event camera. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 235–251 (2018)
71. Zhu, A.Z., Chen, Y., Daniilidis, K.: Realtime time synchronized event-based stereo. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 433–447 (2018)
72. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 989–997 (2019)
73. Zhu, L., Dong, S., Huang, T., Tian, Y.: A retina-inspired sampling method for visual texture reconstruction. In: IEEE International Conference on Multimedia and Expo (ICME). pp. 1432–1437. IEEE (2019)
74. Zhu, L., Dong, S., Li, J., Huang, T., Tian, Y.: Retina-like visual image reconstruction via spiking neural model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)