UNCERTAINTY GUIDED DEPTH FUSION FOR SPIKE CAMERA

Anonymous authors
Paper under double-blind review

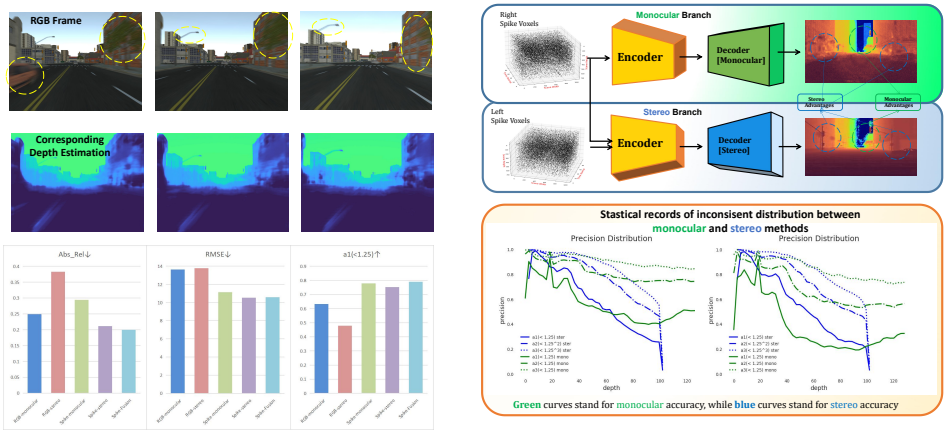
ABSTRACT

Neuromorphic spike camera captures visual streams with high frame rate in a bio-inspired way, bringing vast potential in various real-world applications such as autonomous driving. Compared with traditional cameras, spike camera data has an inherent advantage to overcome motion blur, leading to more accurate depth estimation in high-velocity circumstances. However, depth estimation with spike camera remains very challenging when using traditional monocular or stereo depth estimation algorithms, which are based on the photometric consistency. In this paper, we propose a novel and effective approach for spike depth estimation, which fuses the monocular and stereo depth estimation for spike camera based on the uncertainty of the prediction. Our approach is motivated by the fact that stereo spike depth estimation achieves better results in closer range while monocular spike depth estimation obtains better results in farther range. Therefore, we introduce an Uncertainty-Guided Depth Fusion (UGDF) framework with a joint training strategy and estimate the distributed uncertainty to fuse the monocular and stereo results. In order to demonstrate the advantage of spike depth estimation over traditional camera-based depth estimation, we contribute a spike-depth dataset named CitySpike20K, which contains 20K paired samples, for spike depth estimation. We also introduce the Spike-Kitti dataset to demonstrate the effectiveness and generalization of our method under real-world scenarios. Extensive experiments are conducted to evaluate our method on CitySpike20K and Spike-Kitti. UGDF achieves state-of-the-art results on both CitySpike20K and Spike-Kitti, surpassing all the monocular or stereo spike depth estimation baselines. To the best of our knowledge, our framework is the first end-to-end dual-task fusion framework for spike camera depth estimation. Code and dataset will be released.

1 INTRODUCTION

Neuromorphic spike camera captures visual streams with high frame rate in a bio-inspired way, bringing vast potential in various real-world applications such as robotic manipulation Tremblay et al. (2018), augmented reality Tang et al. (2019); Marchand et al. (2015), and autonomous driving Manhardt et al. (2019); Wu et al. (2019). Compared with traditional cameras, spike camera data has an inherent advantage to overcome motion blur, leading to more accurate depth estimation in high-velocity circumstances Hu et al. (2021); Dong et al. (2019); Zhu et al. (2019). Since spike cameras can capture the pixel-wise luminance intensity at high frame rate, spike depth estimation is a novel direction and ideal solution to depth estimation in high-velocity motion Wang et al. (2022).

Although there have been traditional works on monocular depth estimation Mayer et al. (2016); Kendall et al. (2017); Khamis et al. (2018); Chabra et al. (2019); Guo et al. (2019) and stereo depth estimation Xu et al. (2018); Ramamonjisoa et al. (2020); Lee & Kim (2019); Ramamonjisoa & Lepetit (2019); Fu et al. (2018); Godard et al. (2017)Liu et al. (2022). It is still very challenging to apply them to spike depth estimation since spike data lacks reliable photometric consistency. In order to solve this problem, we first design a bio-inspired neuromorphic encoding module to deal with spatially sparse spike data, and then analyze the pros and cons of monocular and stereo depth estimation. On the one hand, monocular depth estimation is inherently ill-posed and mainly depends on the semantic knowledge of features. Therefore, it is robust to the disparity error and achieves better results at long range. On the other hand, stereo depth estimation compares the local patch pairs to obtain the optimal disparity. Therefore, it obtains better results in closer range and performs worse in farther range. As shown in Figure 1 (b), we conduct analysis on the performance of monocular



(a) (b)

Figure 1: **Problem motivation**. Sub-figure (a) demonstrates the advantage of spike camera when dealing with fast-moving objects for driving depth estimation. It shows the performance gap between the spike-based depth estimation algorithm and blurred RGB-based depth estimation algorithm. Sub-figure (b) reveals the monocular depth estimation and stereo depth estimation usually have different accuracy in different depth ranges. Stereo method achieves better results in closer range while monocular method obtains better results in farther range. Such observation motivates us to fuse the predictions in a dual-task depth estimation architecture.

and stereo spike depth estimation. Stereo methods fails to regress accurate disparity and depth at further regions loaded with fast-changing binary spikes due to their special architecture design, while monocular methods maintain stable performance. Such observation or experimental stastics motivates us to fuse the monocular and stereo predictions for spike depth estimation, alleviating the problem of lacking reliable photometric consistency.

In this paper, we propose a novel Uncertainty-Guided Depth Fusion (UGDF) framework to fuse the predictions of monocular and stereo spike depth estimation. Instead of training the monocular and stereo models separately, UGDF introduces a depth estimation architecture for the dual tasks with a joint training strategy. This architecture includes three components. The first component is a shared encoder, which learns feature representations to build stereo cost volume and monocular depth regression. The second component consists of two parallel branches for monocular and stereo depth estimation. For the monocular branch, we set decoder to consist of three upsampling blocks. As for the stereo branch, we utilize a 3D hourglass-shaped convolution to aggregate the disparity dimension feature of 4D cost volume Chang & Chen (2018). Then, to fuse the predictions of both branches, instead of naive linear fusion, we introduce a novel adaptive uncertainty-guided fusion approach. Different from occlusion-aware fusion Chen et al. (2021c), which only exploits the knowledge from stereo branch, we measure the performances of monocular and stereo branches based on uncertainty regression Zhou et al. (2021). Guided by the uncertainty maps, we fuse the reliable predictions of monocular and stereo branches, taking advantage of both tasks for the final estimation. Note that previous architectures fail to converge well if directly loaded with spike data, so we develop a neuromorphic encoding method to extract significant information from spike voxels.

In addition, we contribute a spike-depth dataset named CitySpike20K, which consists of 20K paired samples, for spike depth estimation. We demonstrate the significant advantages of spike camera for high-velocity depth estimation on CitySpike20K. We also introduce the Spike-Kitti dataset to demonstrate the effectiveness and generalization of our method under real-world scenarios. Extensive experiments are conducted to show the superior performance of our framework compared with state-of-the-art monocular and stereo baselines. Our contributions can be concluded as follows:

- We propose a novel Uncertainty-Guided Depth Fusion framework to fuse the predictions of monocular and stereo spike depth estimation, alleviating the problem of lacking reliable photometric consistency for spike data.

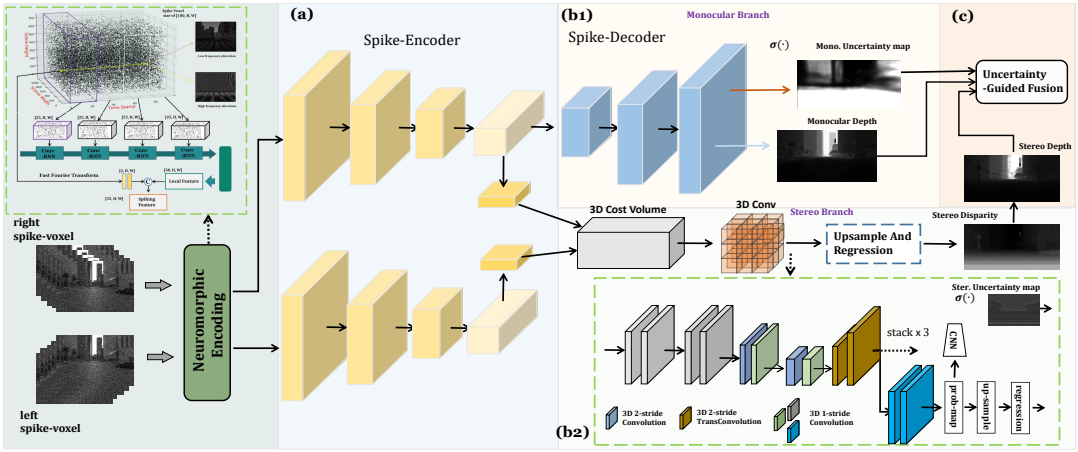


Figure 2: **Network architecture of UGDF**. The network consists of three major modules. Processed spike data pairs are sent into spike encoders, which contain 3 downsampling layers, to extract initial representation (a). Monocular and stereo branches deal with these features, and output depth and disparity respectively (b1, b2). Then, a final uncertainty-guided fusion is performed to aggregate monocular and stereo results (c).

- We introduce a dual-task depth estimation architecture along with a joint training strategy. To the best of our knowledge, we are the first to fuse dual tasks for spike depth estimation.
- We contribute a spike dataset named CitySpike20K, which contains 20K spike-depth pairs, to demonstrate the advantages of spike camera over traditional cameras on high-velocity depth estimation.
- We conduct extensive experiments to evaluate the advantages of our method against existing monocular and stereo baselines.

2 Related Work

2.1 Monocular and Stereo Depth Estimation

Monocular and stereo methods are two mainstream algorithms for depth estimation. Current popular design for monocular depth estimation is the encoder-decoder structure based on CNNs Xu et al. (2018); Ramamonjisoa et al. (2020); Lee & Kim (2019); Ramamonjisoa & Lepetit (2019); Fu et al. (2018); Godard et al. (2017)Zhuang et al. (2022) or transformers Ranftl et al. (2021); Yang et al. (2021), with ground-truth full-supervision or self-supervision Godard et al. (2017); Guizilini et al. (2020); Lyu et al. (2020). Current deep learning-based stereo depth estimation usually contains three main steps: (1) feature extraction (2) cost aggregation, and (3) disparity/depth regressionMayer et al. (2016); Kendall et al. (2017); Khamis et al. (2018); Chabra et al. (2019); Chang & Chen (2018); Guo et al. (2019); Xu & Tao (2020). The above monocular and stereo depth estimation methods are based on the photometric consistency of RGB images. However, in order to capture the images at high frame rate, spike data lacks reliable photometric consistency. Therefore, applying existing methods to spike depth estimation cannot achieve satisfying results. In this paper, we propose an uncertainty-guided depth fusion framework for high-quality spike depth estimation.

2.2 Spike Camera

Different from the RGB cameras and dynamic vision sensors, spike camera mimics the retina to record natural scenes by continuous-time spikes Yu et al. (2020); Zhu et al. (2019). Zhao et al. (2020) develop a new image reconstruction approach for the spike camera to recover high-speed motion scenes. Zheng et al. (2021a); Zhao et al. (2021b); Zheng et al. (2021b); Zhu et al. (2021) use spike or regular CNNs to reconstruct high quality and high-speed images from spike streams. Spike vision shows obvious advantages in capturing high-speed moving objects or scenes, so it provides new solutions to some long-standing problems in the field of computer vision. In this paper, we

propose a novel method for high-quality spike depth estimation by fusing monocular and stereo depth estimation.

3 PROPOSED METHOD

In this section, we present our method uncertainty-guided depth fusion framework (UGDF) to fully complement the strengths of both stereo and monocular tasks in spike data. The whole framework is demonstrated in Fig. 2 which consists of four components.

3.1 SPIKE DATA ANALYSIS

For spike camera, natural lights are captured by photoreceptors and converted to voltage under the integration of time series t. Once the voltage at a certain sensing unit reaches a threshold Θ, a one-bit spike is fired and the voltage is reset to zero at the same timeZhao et al. (2020).

$$S(i,j,t) = \begin{cases} 1, & \int_{t0_{i,j}^{pre}}^{t} I(i,j)\,dt \geq \Theta \\ 0, & \int_{t0_{i,j}^{pre}}^{t} I(i,j)\,dt < \Theta \end{cases} \tag{1}$$

The above formula reveals the basic working pipeline of the spike camera, where $I(i,j)$ represents the luminance of pixel (i,j), and $t0_{i,j}^{pre}$ represents the time that fires the last spike at pixel(i,j). An ideal Analog to Digital Conversion(ADC) process is by continuous time, but such circumstances do not exist due to the inherent limitations of the digital circus. Even so, the spike camera is still able to generate much more dense frames than RGB models like streams, at a maximum frequency of 40000HzZhao et al. (2021a); Zhu et al. (2020); Zhao et al. (2021b). Suppose we have $H \times W$ receptive field, the camera would output a $H \times W$ binary spike frame at a certain moment, and as time goes on, high-frequency spike frames are produced. However, directly performing depth estimation on spike frames remains challenging. On one hand, high contrast between 1-bit spike data makes it more difficult to distinguish local context information. On the other hand, different light intensities in the scene cause different frequencies of spike generation. So in practice, we make spike data in a fixed size time window to be a multi-channel tensor. For example, we take spike frames from continuous 100 time-steps frames and concatenate them at time dimension as $100 \times H \times W$ voxels, which then become inputs to our designed networks.

3.2 UGDF FRAMEWORK

We propose a simple yet efficient network that includes a shared spike-encoder and a spike-decoder with two branches. First, we build a neuromorphic encoding module to extract spiking features in both time domain and frequency domain. The spike voxel $V \in \mathbb{Z}^{100\times H\times W}$ is split into spike sequences $\hat{V_s} = \{v_1, v_2, ..., v_s\}$ by a fixed length of time window n, where s = $|100/n|_{\mathbb{Z}}$ and $v_s \in \mathbb{Z}^{n\times H\times W}$. Then, the spike sequences are fed into a convolutional RNN to extract temporal connections. Meanwhile, a DFT operation is performed on spike voxel V to extract global information in frequency domain. Note that we can obtain frequency domain features for each pixel $a_{i,j,r} = \sum_{k=0}^{T-1}(V_{i,j})W^{rk}$ where $W = e^{-2\pi j/T}$ and $r \in [0, T)$, we sort the real part of the imaginary frequency features, select the max value as the smooth-feature and its index as the sharp-feature. At last, we concatenate the frequency and temporal features to be the last spiking features.

we use a shared deep encoder to learn a representation to build stereo cost volume and monocular depth regression, as shown in part (a) of Figure 2. We adopt MobileNetV3Howard et al. (2019) as our encoder to make a trade-off between computation cost and model performance. The encoder contains 3 downsampling stage and the final feature map size of coding is $B \times 256 \times \frac{H}{8} \times \frac{W}{8}$.

As shown in the part (b1) and (b2) of Figure 2, two parallel branches stretch away for monocular and stereo depth estimation and serves as the spike-decoder. Different from any other previous works, we fuse monocular and depth estimation in one workflow at a multi-task level. In the stereo branch, we found a common ground for monocular and stereo tasks to learn a global context representation. So we take advantage of the encoder module, and concatenate the unary features(obtained coding from spike encoders) to build a 4D cost volume($256 \times$ ***Max-Disp.*** $\times \frac{H}{8} \times \frac{W}{8}$) for stereo disparity regression, where ***Max-Disp.*** represents the maximum disparity level to regression. Then inspired by Chang & Chen (2018), we perform a 3D hourglass-shaped convolution to aggregate the disparity

dimension feature of 4D cost volume. We stack three 3D hourglasses, each of which contains two blocks with $3\times3\times3$ kernel size and 2-stride 3D convolution, and two blocks with $3\times3\times3$ kernel size and 2-stride 3D transposed convolution. The disparity map is regressed at the final 3D convolution stage via a *soft-argmin* operationKendall et al. (2017): *soft-argmin:* $= \sum_{d=0}^{Disp^*_{max}} d \times \gamma(-c_d)$, where $\gamma(\cdot)$ represents soft-max operation at disparity dimension, c_d is predicted costs for disparity d, and $Disp^*_{max}$ means length of disparity dimension of the output features. The final disparity is weighed by a normalized probability. In the training phase, three 3D hourglass disparity outputs are all involved in building loss function. And the last output of three 3D hourglasses is used for the evaluation process.

In the monocular branch, the right unary features(coding) are then sent to a decoding block for depth estimation. The decoder consists of three upsample blocks, each block contains one bilinear interpolation module and two convolutional layers along with batch normalization and Mish activation. The output layer is a 1×1 convolution which squeeze the features to **two channels**, **one** of which is used for **depth** estimation and **the other** one used for **uncertainty** allocation, as described in the next subsection.

3.3 Uncertainty Guided Fusion

Inspired by SUB-DepthZhou et al. (2021), we assume the distribution over the output of either branch can be modeled as exponential family distribution such as Laplace's distribution or Gaussian distribution. The stereo branch and monocular branch adopt the same approach while we use the monocular branch as an illustration example. Given a dataset with left and right spike frames and corresponding depth ground truth (x_l, x_r, y_l, y_r), we let our monocular branch output the mean $\hat{y}$ and variance σ of the posterior probability distribution $p(y_r|\hat{y_r}, x_r)$. We use Laplace's distribution as:

$$p(y_r|\hat{y_r}, x_r) = \frac{1}{2\sigma} exp \frac{-|\hat{y_r} - y_r|}{\sigma} \tag{2}$$

We can convert the above distribution to a log-likelihood formula like:

$$log(p(y_r|\hat{y_r}, x_r)) = -log(\sigma) + \frac{-|\hat{y_r} - y_r|}{\sigma} + const. \tag{3}$$

So according to max-posterior probability estimation, an uncertainty loss can be formulated in the form of:

$$loss_{unc.} = log(\sigma) + \frac{|\hat{y_r} - y_r|}{\sigma} \tag{4}$$

We can minimize this loss function to obtain a max-posterior probability distribution over estimated monocular depth $\hat{y}$. The uncertainty coefficient σ_m and predicted depth $\hat{y}$, which are the outputs of the part (b1), are regressed from the same decoder at the same time, so σ_m can be seen as prediction uncertainty for monocular depth estimation task. Similar to the monocular branch, a lite CNN is added behind the probability map of the stereo branch, and regresses uncertainty coefficient σ_s after a sigmoid activation.

We notice that the monocular branch outperforms the stereo branch in farther regions and the stereo branch is good at predicting closer regions. So we hope the fusion style may combine both monocular and stereo advantages. So we define a distance threshold:

$$\sigma_{dis.} = D_{max} \frac{e^{2(\sigma_m - \sigma_s)}}{1 + e^{2(\sigma_m - \sigma_s)}} \tag{5}$$

With estimated uncertainty, an uncertainty-guided fusion mask F can be defined as:

$$F_i = \begin{cases} 0, \hat{D}_{mono.} \leq \sigma_{dis.} \\ 1, \hat{D}_{mono.} > \sigma_{dis.} \end{cases} \tag{6}$$

where i represents i-th element of the uncertainty map, and σ_{dis} represents an uncertainty threshold to make fusion. To take advantage of both monocular and stereo branches, we use monocular depth prediction results $\hat{D}_{mono.}$ and stereo depth prediction results $\hat{D}_{ster.}$ to make further fusion, exploiting complementary advantages for monocular and stereo models. Instead of directly performing linear addition between two kinds of outputs, we fuse them in a more efficient uncertainty-guided way. And the uncertain-guided fusion is given:

$$\hat{D_f} = F \odot \hat{D}_{mono.} + (1 - F) \odot \hat{D}_{ster.} \tag{7}$$

3.4 UGDF LOSS FUNCTIONS

We present training strategies for baseline network without fusion and UGDF with fusion. The training loss of baseline network consists of monocular depth estimation $loss_{disp.}$ and stereo disparity regression $loss_{depth}$, which use smooth-L1 loss during the training phase under the supervision of depth ground-truth and generated disparity labels. The baseline $loss_{base}$ is shown as below:

$$loss_{base.} = loss_{disp.} + loss_{depth.} \tag{8}$$

in which $loss_{disp.}$ and $loss_{depth.}$ are shown as below:

$$loss_{disp.}(d^*, \hat{d}) = \frac{1}{N}\sum_{i=1}^{M}\sum_{j=1}^{N}\alpha_i \cdot smoothL1(d^*, \hat{d}) \tag{9}$$

$$loss_{depth.}(D, \hat{D}) = \frac{1}{n}\sum_{i} {c_i}^2 - \frac{1}{n^2}(\sum_{i} c_i)^2 + \eta \tag{10}$$

In which $\eta = 0.1$ and $c_i = log(D_i) - log(\hat{D}_i)$. And N is the size of data and M=3 is the number of stacked 3D hourglasses, $\{\alpha_1, \alpha_2, \alpha_3\} = \{0.5, 0.7, 1.0\}$. $\hat{D}$ and D represent predicted depth and depth ground truth respectively. Similarly, $\hat{d}$ means predicted disparity and d^* is generated disparity ground truth.

The training loss of UGDF consists of five losses, including monocular depth estimation $loss_{disp.}$, stereo disparity regression $loss_{depth.}$, monocular branch uncertainty $loss_{mono\_unc.}$, stereo branch uncertainty $loss_{ster\_unc.}$ and fusion $loss_{fu.}$. The whole UGDF $loss_{ugdf.}$ is shown as:

$$loss_{ugdf.} = loss_{base.} + loss_{ster\_unc.} + loss_{mono\_unc.} \tag{11}$$

where $loss_{mono-unc.}$ and $loss_{ster-unc.}$ follow Eq. 5 while $\hat{D_f}$ denotes the fusion predicted depth from two branches. To be mentioned, the depth of the stereo branch is converted from disparity under intrinsic parameters of the camera. The training details of baseline and UGDF are the same. Further training details are presented in Section 5.

4 SPIKE-DEPTH DATASET: CITYSPIKE20K

This section introduces different aspects of the dataset we propose. The dataset includes RGB scenes, and their corresponding spike frames and depth maps. All these data are generated by a simulated spike camera in Unity3D virtual environment. The dataset describes 11 sequences of city street scenes containing 6 day scenes and 5 night scenes. Our proposed CitySpike20K dataset provides a depth estimation benchmark for spike data. The scenes are created via a simulated spike camera, recording a fast-moving car in the street scene, at a frequency of 1000Hz. The resolution for recorded data is 1024×768. We also build a depth field for these scenes and store them as 24-bit depth maps. The ground truth information is of 0.3-1000m absolute depth. In addition, we provide the focus f and baseline length $base_{len}$ of the stereo camera in supplement. We also convert depth D to disparity $disp$ under the function $disp. = \frac{f * base_{len}}{D}$. A visualization of our dataset is shown in **appendix**. Besides, in terms of sensor-collaboration, we provide 842 pairs of RGB images from regular stereo cameras, dense spike frames from stereo Vidar, as well as depth maps from stereo depth cameras. Three kinds of data are organized in a one-to-one corresponding way. Besides, we provide a demo sequence of 40000Hz frequency spike data, recording a 91km/h car driving in the city street. This demo is for evaluating the depth estimation algorithm when loaded with high-frequency spike data.

5 EXPERIMENTS

In this section, we conduct extensive experiments to show the advantages of UGDF. Then, we extensively evaluate UGDF by comparing it with the state-of-the-art and classic depth estimation methods which have shown great performance on RGB depth datasets such as KITTIUhrig et al. (2017) and NYUD-V2Nathan Silberman & Fergus (2012). We also conduct comprehensive ablation studies to evaluate the contribution of each component in the last subsection. Due to space limitations, some details of experiments and results are provided in the supplementary materials.

Table 1: Quantitative results on CitySpike20K (decribed as CS20K below) **validation** set. Evaluation metrics are as described in section 3. We make comparison with GwcNetGuo et al. (2019), CFNetShen et al. (2021), PSMNetChang & Chen (2018). The evaluation metrics are as introduced in subsection 4.2. We also consider model parameter size to be one of the compared targets.

Dataset	Method	Approach	Modality	Abs_Rel↓	RMSE ↓	Sq_Rel ↓	RMSE_log ↓	a1 ↑	a2 ↑	a3 ↑
CS20K	PSMNet	Ster.	RGB	0.4564	15.484	12.990	0.734	0.469	0.668	0.743
	GwcNet	Ster.	RGB	0.419	19.724	9.753	0.632	0.469	0.685	0.767
	CFnet	Ster.	RGB	0.4038	14.928	8.870	0.437	0.593	0.677	0.786
CS20k	**UGDF**(Ours)	Fusion	Spike	**0.2282**	**11.075**	**4.699**	**0.305**	**0.754**	**0.879**	**0.942**

Table 2: Quantitative results on CitySpike20K **test** set. We add two monocular algorithms as baselines which are DPTRanftl et al. (2021) and UNetRonneberger et al. (2015).

Dataset	Method	Approach	Modality	Abs_Rel↓	RMSE ↓	Sq_Rel ↓	RMSE_log ↓	a1 ↑	a2 ↑	a3 ↑
CS20K	UNet	Mono.	RGB	0.3612	19.217	6.981	0.502	0.569	0.765	0.893
	DPT	Mono.	RGB	0.249	13.641	4.349	0.379	0.632	0.817	0.925
	PSMNet	Ster.	RGB	0.4341	16.294	9.247	0.840	0.411	0.626	0.712
	GwcNet	Ster.	RGB	0.3931	18.680	8.745	0.577	0.492	0.704	0.787
	CFnet	Ster.	RGB	0.3825	13.794	7.925	0.496	0.467	0.723	0.836
CS20k	**UGDF**(Ours)	Fusion	Spike	**0.1997**	**10.953**	<u>4.879</u>	<u>0.412</u>	**0.790**	**0.888**	**0.945**

5.1 Implementation Details

We train our proposed UGDF network on spike-depth pairs, including stereo spike frames and right depth-ground-truth. The whole training phase contains 200 epochs and takes about 16 hours with the batch size of 4 on two NVIDIA-Tesla P100 GPUs, for 256×512 resolution spiking frames.

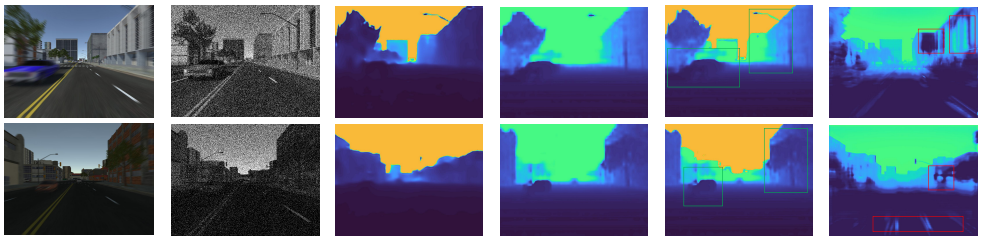
We utilize 24-bit 0-1000m absolute depth ground-truth to supervise training for the monocular branch. We normalize depth ground truth D to $D^* \in (0, 1)$, with the function $D^* = D/1000$. Meanwhile, the disparity is transformed from depth with camera intrinsics. As for optimization, we use Adam optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$. We set an initial learning rate of 1e-3 and decay to 0.33e-3 at epoch 35 for the sake of a more smooth optimizing process.

In this section, we compare UGDF against the state-of-the-art and classic depth estimation methods on CitySpike20K dataset.

Data processing Our proposed dataset contains 20K frames of spike-depth pairs. We split 7 out of 10 total sequences for training, 2 sequences for testing and 1 sequence for validating. All the data used for training and validating is sampled every 100 time-stamps to form a spike voxel. So we obtain 140 training pieces and 40, 20 for testing and validating our framework. To emphasize the advantage of spike data, we use blurred 30fps RGB frames in our dataset to train the RGB-based baseline methods. So there are 571 training pieces, 142 testing pieces and 111 validation pieces for baseline methods.

Baseline methods To demonstrate the effectiveness of UGDF, we compare it with some state-of-the-art and classic depth estimation methods which have shown remarkable performance on our proposed CitySpike20k dataset. For monocular methods, we choose classical UNet Ronneberger et al. (2015) and DPTRanftl et al. (2021). UNet has been demonstrated a successful design on semantic segmentationRonneberger et al. (2015); Baheti et al. (2020) and image reconstructionChen et al. (2021b;a). we adopt its proposed structure and evaluate it on our spike-depth estimation task. In DPT, we use Vit-b16 as the backbone and 224x224 as input resolution. PSMNetChang & Chen (2018) uses subtract and concatenation method to build a 3D cost volume. GwcNetGuo et al. (2019) proposes group-wise correlation to reduce computation while conducting 3D convolution. CFNetShen et al. (2021) employs a variance-based uncertainty estimation to adaptively search disparity space.

Main Results and Analysis for CS20K Table 1 and Table 2 show quantitative results with the comparison of the RGB methods. We experiment with classic monocular depth estimation works, as well as stereo depth estimation methods. We can see that under the uncertainty-guided fusion, our result gets the top performance among all the methods. Compared with the best monocular method, UGDF reduces 4.93%, 2.688 error in terms of AbsRel.and RMSE metric respectively. For stereo methods, we also gained improvements on all metrics. We also show the qualitative comparison



(a) RGB Frame (b) Spike Frame (c) Mono. result (d) Ster. result (e) Fusion result (f) RGB result

Figure 3: Visualization of depth estimation on CitySpike20K. Pic. a is 30hz RGB data, and b is one spike frame in a spike voxel. Pic. c-e is output result of our method, and f is UNet output for RGB depth estimation.

Table 3: Quantitative results on Spike-Real **test** set. Our UGDF framework still obtains performance increase to two branches.

Dataset	Method	Approach	Modality	Abs_Rel↓	RMSE ↓	Sq_Rel ↓	RMSE_log ↓	a1 ↑	a2 ↑	a3 ↑
Real	PSMNet	Ster.	Image	0.3743	2.228	0.413	0.843	0.451	0.703	0.838
Real	**UGDF**(Ours)	Ster.	Spike	0.2722	1.264	0.376	0.348	0.581	0.819	0.906
		Mono.		0.4037	1.552	1.017	0.382	0.528	0.796	0.889
		Fusion		**0.2693**	1.237	0.413	0.374	0.533	0.795	0.899

in Figure 3. As can be seen, our method achieves better depth estimation compared with blurred RGB-based methods. Other visualization results are in our supplement.

Evaluation on Spike-Real dataset We also train and evaluate our network on a dataset captured by a stereo real-world Vidar in a series of outdoor scenes. The dataset contains 40 sequences of outdoor scenes and we split 33 sequences for training and 7 sequences for testing. Table 3 shows results evaluated on its test set.

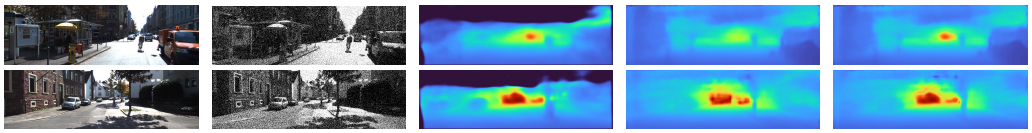
Evaluation on Spike-Kitti dataset To verify the performance in the real-world context, we transform the RGB data of "City" scene from kittiGeiger et al. (2012) dataset into spike modality, and evaluate on the official validation set. Note that all the data we use is only from "City" scene. The resulting performance is shown in Table 4 and Figure 4. Specifically, we insert frames on kitti data using XVFISim et al. (2021) by 128 times to obtain continuous spike streams.

5.2 Ablation Study

We carry out ablation experiments from two aspects. The first of those is to explore the effect of different choices of time-window widths, and the other is to verify the effectiveness of uncertainty-guided fusion design.

Effectiveness of Uncertainty Guided Fusion We conduct experiments to verify the effectiveness of monocular and stereo uncertainty jointly guided fusion. In order to demonstrate the benefits of joint-guided fusion, we first compare it with a linear additive ensemble fusion manner. To be specific, we make a uniform linear addition of monocular and stereo estimation results, denoted as E. Fusion in Table 5 As can be seen, the linear additive fusion manner is inferior to other fusion methods. In addition, we visualize the improvement gap between fusion results and the other two branches. We can see the advantages of our UGDF framework. Firstly it combines both advantages of stereo and monocular estimation. And secondly, it brings substantial improvements, rather than the compromising fusion of ensemble style.

Time window Width of Neuromorphic Encoding In our framework, we apply a kind of neuromorphic encoding method to effectively extract the feature of spike data. As we have described in Section 3.2, we chunk the spike voxel into spike sequences by the time window of 24 to obtain better local representations. Then the sequences are sent into the Conv-RNN to extract temporal connections between different sequences. Theoretically, applying a smaller time window is beneficial to extract local connections between spike sequences, yet increases the convergence and inference time



(a) RGB Frame (b) Spike Frame (c) Mono. result (d) Ster. result (e) Fusion result

Figure 4: Example outputs of UGDF depth estimation framework on Spike-Kitti. As can be seen, our method still works well and makes sense loaded with real-world data.

Table 4: Quantitative results on Spike-Kitti **val** set. Our UGDF framework compensates both branches and achieves promising performance. We set max-distance as 80m to evaluate the accuracy.

Dataset	Method	Approach	Modality	Abs_Rel↓	RMSE ↓	Sq_Rel ↓	RMSE_log ↓	a1 ↑	a2 ↑	a3 ↑
Spike-Kitti	**UGDF**(Ours)	Ster.	Spike	0.1250	4.283	0.717	0.188	0.830	0.957	0.986
		Mono.		0.1706	5.067	1.127	0.242	0.753	0.910	0.968
		Fusion		**0.1247**	4.281	0.721	0.189	0.829	0.957	0.985

Table 5: Ablation study on the effect of fusion on CitySpike 20K. We present the result of applying different branch respectively. The designed depth estimation fusion method shows strong-efficiency and gains improvements for both branches.

Split	Branch	Abs_Rel	Sq_Rel	a1
Valid	Mono.	0.3302 (**0.102**↓)	12.759	0.738
	Ster.	0.2543 (0.026↓)	3.995	0.613
	E. Fusion	0.2652 (0.037↓)	5.712	0.706
	U. Fusion	0.2282	4.699	0.754

Split	Branch	Abs_Rel	Sq_Rel	a1
Test	Mono.	0.2944 (**0.095**↓)	12.508	0.779
	Ster.	0.2118 (0.012↓)	3.780	0.753
	E. Fusion	0.2347 (0.035↓)	4.018	0.761
	U. Fusion.	0.1997	4.897	0.791

Table 6: Ablation results on test split of CS20K for window-width of neuromorphic encoding. The runtime statistics are made on RTX 2080ti GPU for a single forward pass of the network with the batch-size of 1.

Width	Time	**Error**				
$0 < s < 100$	(ms)	Abs_Rel	Sq_Rel	a1	a2	a3
8	5.3	0.2301	5.146	0.756	0.877	0.942
16	2.8	0.2143	4.699	0.764	0.892	0.946
24	2.1	0.1997	4.879	0.790	0.888	0.945
32	1.6	0.2552	5.612	0.726	0.876	0.934

on hardware. So we set different widths of encoding time-window and train the whole network for 200 epochs. We make evaluations on the test set of CitySpike20K. The results are shown in Table 6, and we can see that 24 turns out to be the optimal choice of time-window widths for shorter inference time and better precision.

6 Conclusion

In this paper, we propose an uncertainty-guided depth fusion framework for spike data, consisting of four modules including neuromorphic encoding module, spike encoder, spike decoder for monocular and stereo tasks, and uncertainty-guided fusion. The major motivation of our work is monocular and stereo depth estimation shows different advantages in different scenarios for spike data. So it is critical to explore an effective fusion method to leverage the advantages of both tasks. Different from previous works, we fuse monocular and stereo depth prediction results according to individual adaptive uncertainty estimations. We also generate a spike dataset for depth estimation which contains 20K paired spike-depth data (CitySpike20K), along with its technical details and evaluation metrics. We demonstrate the good efficiency of spike data when applied to fast-moving circumstances. Extensive experiments are conducted to validate the effectiveness of our proposed UGDF. We hope this paper can inspire future works on spike data depth estimation.

REFERENCES

Bhakti Baheti, Shubham Innani, Suhas Gajre, and Sanjay Talbar. Eff-unet: A novel architecture for semantic segmentation in unstructured environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 358–359, 2020.

Rohan Chabra, Julian Straub, Christopher Sweeney, Richard Newcombe, and Henry Fuchs. Stereodrnet: Dilated residual stereonet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5410–5418, 2018.

Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 182–192, 2021a.

Xiangyu Chen, Yihao Liu, Zhengwen Zhang, Yu Qiao, and Chao Dong. Hdrunet: Single image hdr reconstruction with denoising and dequantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 354–363, June 2021b.

Zhi Chen, Xiaoqing Ye, Wei Yang, Zhenbo Xu, Xiao Tan, Zhikang Zou, Errui Ding, Xinming Zhang, and Liusheng Huang. Revealing the reciprocal relations between self-supervised stereo and monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15529–15538, 2021c.

Siwei Dong, Lin Zhu, Daoyuan Xu, Yonghong Tian, and Tiejun Huang. An efficient coding method for spike camera using inter-spike intervals. *arXiv preprint arXiv:1912.09669*, 2019.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2002–2011, 2018.

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision Pattern Recognition*, 2012.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 270–279, 2017.

Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2485–2494, 2020.

Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3273–3282, 2019.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Liwen Hu, Rui Zhao, Ziluo Ding, Lei Ma, Boxin Shi, Ruiqin Xiong, and Tiejun Huang. Optical flow estimation for spiking camera. *arXiv preprint arXiv:2110.03916*, 2021.

Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE international conference on computer vision*, pp. 66–75, 2017.

Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Jae-Han Lee and Chang-Su Kim. Monocular depth estimation using relative depth maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2019.

Biyang Liu, Huimin Yu, and Yangqi Long. Local similarity pattern and cost self-reassembling for deep stereo matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 1647–1655, 2022.

Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: High resolution self-supervised monocular depth estimation. *arXiv preprint arXiv:2012.07356*, 6, 2020.

Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2069–2078, 2019.

Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2015.

Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4040–4048, 2016.

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

Michael Ramamonjisoa and Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

Michael Ramamonjisoa, Yuming Du, and Vincent Lepetit. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12179–12188, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13906–13915, 2021.

Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: Extreme video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14489–14498, 2021.

Fulin Tang, Yihong Wu, Xiaohui Hou, and Haibin Ling. 3d mapping and 6d pose computation for real time augmented reality on cylindrical objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(9):2887–2899, 2019.

Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.

Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.

Yixuan Wang, Jianing Li, Linlin Zhu, Xijie Xiang, Tiejun Huang, and Yonghong Tian. Learning stereo depth estimation with bio-inspired spike cameras. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022.

Di Wu, Zhaoyong Zhuang, Canqun Xiang, Wenbin Zou, and Xia Li. 6d-vnet: End-to-end 6-dof vehicle pose estimation from monocular rgb images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Qingshan Xu and Wenbing Tao. Learning inverse depth regression for multi-view stereo with correlation cost volume. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12508–12515, 2020.

Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16269–16279, 2021.

Zhaofei Yu, Jian K Liu, Shanshan Jia, Yichen Zhang, Yajing Zheng, Yonghong Tian, and Tiejun Huang. Toward the next generation of retinal neuroprosthesis: visual computation with spikes. *Engineering*, 6(4):449–461, 2020.

Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip H.S. Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Jing Zhao, Ruiqin Xiong, and Tiejun Huang. High-speed motion scene reconstruction for spike camera via motion aligned filtering. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2020. doi: 10.1109/ISCAS45731.2020.9181055.

Jing Zhao, Jiyu Xie, Ruiqin Xiong, Jian Zhang, Zhaofei Yu, and Tiejun Huang. Super resolve dynamic scene from continuous spike streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2533–2542, October 2021a.

Jing Zhao, Ruiqin Xiong, Hangfan Liu, Jian Zhang, and Tiejun Huang. 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, June 2021b.

Yajing Zheng, Lingxiao Zheng, Zhaofei Yu, Boxin Shi, Yonghong Tian, and Tiejun Huang. 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, June 2021a.

Yajing Zheng, Lingxiao Zheng, Zhaofei Yu, Boxin Shi, Yonghong Tian, and Tiejun Huang. High-speed image reconstruction through short-term plasticity for spiking cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6358–6367, 2021b.

Hang Zhou, Sarah Taylor, and David Greenwood. Sub-depth: Self-distillation and uncertainty boosting self-supervised monocular depth estimation. *arXiv preprint arXiv:2111.09692*, 2021.

Lin Zhu, Siwei Dong, Tiejun Huang, and Yonghong Tian. A retina-inspired sampling method for visual texture reconstruction. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1432–1437. IEEE, 2019.

Lin Zhu, Siwei Dong, Jianing Li, Tiejun Huang, and Yonghong Tian. Retina-like visual image reconstruction via spiking neural model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1438–1446, 2020.

Lin Zhu, Jianing Li, Xiao Wang, Tiejun Huang, and Yonghong Tian. Neuspike-net: High speed video reconstruction via bio-inspired neuromorphic cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2400–2409, 2021.

Chuanqing Zhuang, Zhengda Lu, Yiqun Wang, Jun Xiao, and Ying Wang. Acdnet: Adaptively combined dilated convolution for monocular panorama depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3653–3661, 2022.

A Appendix I: Introduction to Spike Camera

RGB Camera:

$$Light(x, y, t) \rightarrow \{I_r(x, y, t), I_g(x, y, t), I_b(x, y, t)\}$$

After a fixed time interval ΔT for camera exposure ΔT_e and internal circus analog-digital conversion and quantizationΔT_{adc}, while $\Delta T = \Delta T_e + \Delta T_{adc}$ and $\Delta T_e \gg \Delta T_{adc}$. A final digital image is generated as $D_{rgb}(x, y, t + \Delta T)$. So the average of $1/\Delta t$ is the frame ratio of RGB camera. However, if the RGB cameras are applied to capture a very fast object, like a 91km/h car in our CitySpike20K dataset, a line-shaped motion blur would be generated

Spike Camera:

In contrast, spike camera is a kind of event-camera, which means the imaging process of the spike camera is event-driven. Every pixel in the spike camera imaging unit is isolated, they don't share a united imaging process and are activated when the imaging condition is met, as described in Sec 3.1 in our paper. This high-frequency event-driven imaging approach guarantees almost no blur in the imaging process. And generated spike streams from the spike camera are discrete and sparse point sets like lidar in 3D space. Given a time window, the spike voxel can be divided into spike seqences $S = \{x_n, y_n, t_n; n = 1, 2, 3..., N\}$. It's worth noting that during the training phase, **one spike voxel corresponds to one depth map**.

So to sum up, spike camera is not restricted by fixed exposure time interval. So ideally the spike camera generates images like streams without imaging frequency, in a continuous time integration. However, Δ_{adc}, no matter how short it is, does exist in all kinds of circus. So in practice, the ADC frequency of the spike camera determines the output frame ratio and reaches as high as 40000hz, meaning that 40000 one-bit frames are generated per second(no matter the spikes are generated or not in the pixels, a spike frame always output at certain timestamps with the frequency 40000Hz). So the Δ_{adc} decides the frequency of spike frames, and the illuminance of pixels(dark or bright) decides the frequency of spike generation (0 or 1) of specific pixels. Back to our motivation, RGB images may not be reliable enough for scene understanding with high driving speed duo to the existence of blur, so we introduce spike vision to tackle this problem.

B Appendix II : Proposed Dataset: CitySpike20K

B.1 Introduction and Visualization

We propose CitySpike20K, a spike-depth dataset to help explore the depth estimation algorithms for spike camera. The dataset is generated by Unity3D and contains 10 sequences, 5 of which are day scenes and 5 others are night scenes. In the dataset, the frequency of the spike data and corresponding depth GTs is 1000Hz. Besides, we supply 30Hz RGB images for each scenes as well as 1000Hz RGB images that aligned with spike data.

To fully simulate the city environments, we add moving automobiles and dynamic traffic lights. We set 5-10 moving automobiles including buses, cars, vans and trucks for each scene. Figure 2 gives a visualization of CitySpike20K which contains RGB frames, spike data and depth maps. Specifically, we split scene03, scene07 for testing, scene09 for validation and others for training.

B.2 Evaluation Metric

We conducted to evaluate the effectiveness of supervised depth estimation model on CitySpike20K. Our evaluation metrics for depth estimation is described as follows:

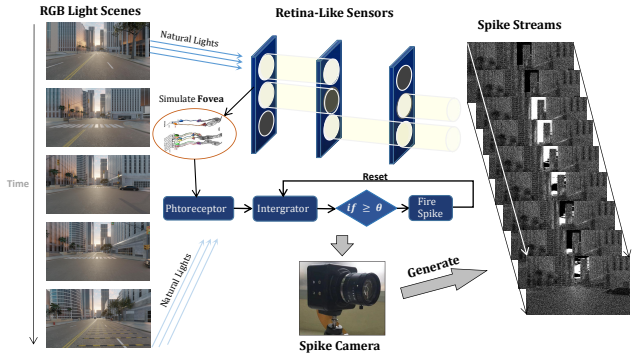


Figure 5: Spike camera is capable of of generating 2-bit spike streams via a retina-like process.

Table 7: Quantitative results on CitySpike20K-demo. Evaluation metrics are as described above. We make comparison with DORNFu et al. (2018), GwcNetGuo et al. (2019), CFNetShen et al. (2021), StereoNetKhamis et al. (2018), PSMNetChang & Chen (2018), and GANetZhang et al. (2019) . The evaluation metrics are as introduced in subsection 4.2. We also consider model parameter size to be one of compared targets.

Dataset	Method	Approach	Abs_Rel↓	RMSE ↓	Sq_Rel ↓	RMSE_log ↓	a1 ↑	a2 ↑	a3 ↑
demo	UNetRonneberger et al. (2015)	Mono.	0.2518	23.993	9.008	0.357	0.68	0.896	0.932
	DORNFu et al. (2018)	Mono.	0.3857	25.258	10.691	0.438	0.409	0.841	0.917
	EigenEigen et al. (2014)	Mono.	0.4262	25.154	20.363	0.459	0.542	0.800	0.893
demo	GC-NetKendall et al. (2017)	Ster.	0.2350	37.158	12.743	0.401	0.614	0.809	0.868
	GwcNetGuo et al. (2019)	Ster.	0.1880	24.152	7.469	**0.304**	0.757	0.895	0.953
	CFnetShen et al. (2021)	Ster.	0.2281	25.905	**5.557**	0.397	0.610	0.847	0.926
	SteroNetKhamis et al. (2018)	Ster.	0.2890	50.765	19.772	0.690	0.563	0.727	0.823
	PSMNetChang & Chen (2018)	Ster.	0.1886	28.496	7.354	0.340	0.723	0.887	0.941
	GANet-1Zhang et al. (2019)	Ster.	0.3270	49.068	19.505	0.865	0.586	0.764	0.851
	GANetZhang et al. (2019)	Ster.	0.2963	47.202	17.598	0.714	0.576	0.771	0.857
demo	**Ours**	Fusion	**0.1715**	**22.793**	11.217	0.306	**0.791**	**0.928**	**0.961**

Given an estimated depth map $\hat{D}$, and its corresponding ground truth D, $N = H \times W$, Abs_Rel is quantified as:

$$Abs\_Rel = \frac{1}{N}\sum_{i=1}^{N}\frac{|D_i - \hat{D}_i|}{D_i} \tag{12}$$

and RMSE defined:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}||D_i - \hat{D}_i||^2} \tag{13}$$

we also introduce $RMSE\_log$ metric:

$$RMSE\_log = \sqrt{\frac{1}{N}\sum_{i=1}^{N}||log(\hat{D}_i) - log(D_i)||^2} \tag{14}$$

and Sq_Rel metric as here:

$$Sq\_Rel = \frac{1}{N}\sum_{i=1}^{N}\frac{||D_i - \hat{D}_i||^2}{D_i} \tag{15}$$

Above metrics measure output errors from different statistic aspect, weighting the distance between predictions and ground-truth labels, where lower values mean better model performance. Below metrics are for evaluation of whether predictions are accurate within certain range of ground-truth, and higher values mean better performance. Note that $j \in \{1, 2, 3\}$

$$aj \quad accuracy : \% \quad of \quad D_i \quad s.t. \quad max(\frac{\hat{D}_i}{D_i}, \frac{D_i}{\hat{D}_i}) = \delta < T = 1.25^j \tag{16}$$

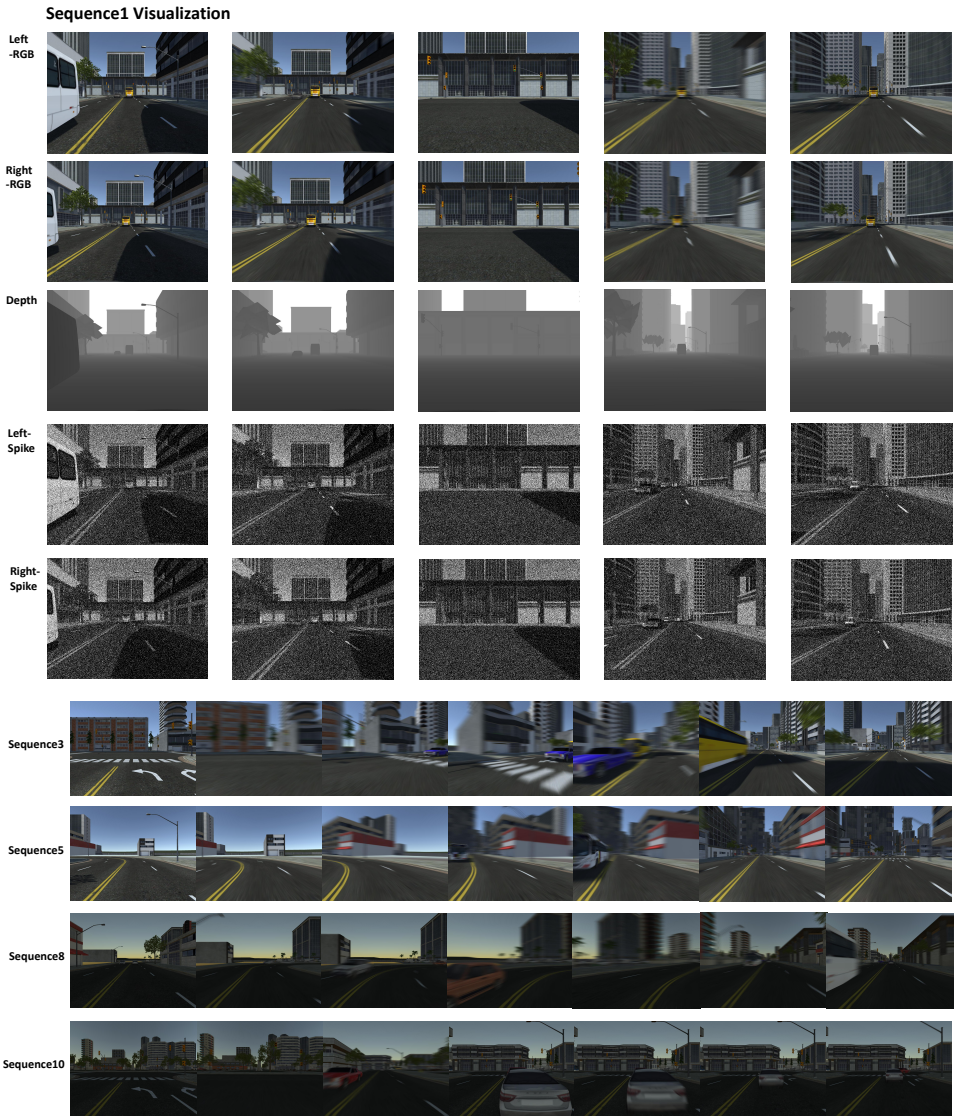


Figure 6: A visualization of our proposed CitySpike20k dataset. We generate it by Unity3D engine and simulate a vivid city environment along with dense depth maps and spike data.

C Appendix III : Performance on other datasets

C.1 Real-Dataset

As we have described in our submitted paper, we also evaluate our framework on a real-recorded dataset by a spike camera. The dataset contains 40 sequences data and each of which includes 3-6 $[400 \times 250 \times 400]$ spike voxels in the format of $[T \times H \times W]$. We split 33 sequences for training and 7 for testing.

C.2 Kitti

To demonstrate that our UGDF framework still works in real-world scenes, we carry out experiment on a spike-kitti dataset. To convert KittiGeiger et al. (2013) from RGB modality to spike modality, we first make frame interpolation using XVFISim et al. (2021) by 128 times. Then we use a Simulated-Vidar code script to generate spike data from RGB Kitti images to form spike voxels in the format $(128 \times 375 \times 1242)$, where 128 represents the time dimension and (375×1242) is the

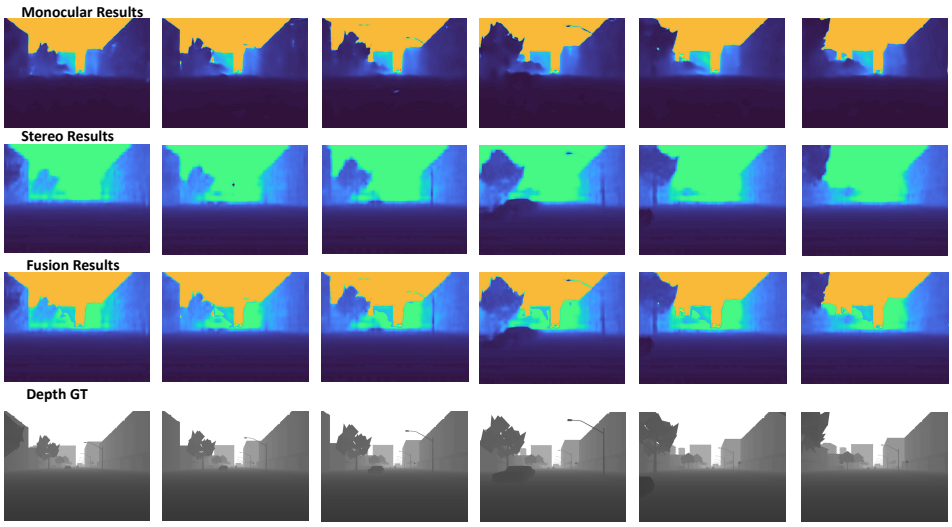


Figure 7: More prediction results on CitySpike20K dataset. As can be seen, the stereo estimation results and the monocular estimation results fuse efficiently by our framework

Figure 8: A visualization for Spike-Real dataset and prediction results from its test set.

original size of Kitti RGB images. We maintain the same way to operate neuromorphic encoding as what we design for CitySpike20K dataset in our submitted paper. As mentioned above, we set this experiment to further explore the effectiveness of our fusion strategy. We train our framework for 50 epochs on 4 RTX-2080Ti GPUs.

Specifically, we use official validation sequences 2011_09_26_drive_0002_sync ,2011_09_26_drive_0005_sync,2011_09_26_drive_0013_sync, 2011_09_26_drive_0095_sync, 2011_09_26_drive_0113_sync for validation, and 2011_09_26 other official training sequences to train our framework.

C.3 CitySpike20K-demo

In addition to 10 sequences of 1000Hz spike data we provide in the CitySpike20K dataset, we still supply a 40000Hz demo to simulate real spike as possible as we could. The demo contains 60K paired data and records a 1.5 seconds video of a fast-driving car in the city street. Different from our submitted papers, we use this demo to evaluate the performance of models to directly load with spike data. Considering existing methods for monocular or stereo depth estimation are mostly based on RGB 3-channel data, we change the input channel of the models to the time-window width of applied spike sequences, i.e. 32 as we adopted. And we use the first half of the demo for training and the second half for testing. Table 7 records relevant results compared with state-of-the-art traditional methods. Note that we adopt ResNet-50 instead of MobileNetV3 as backbone for this part experiments.

D Appendix IV : Statistics to Support Our Motivation

There are two clues to inspire our motivations. The first of which is that, the spike camera has its unique advantages to deal with fast-moving circumstances when operating depth estimation task.

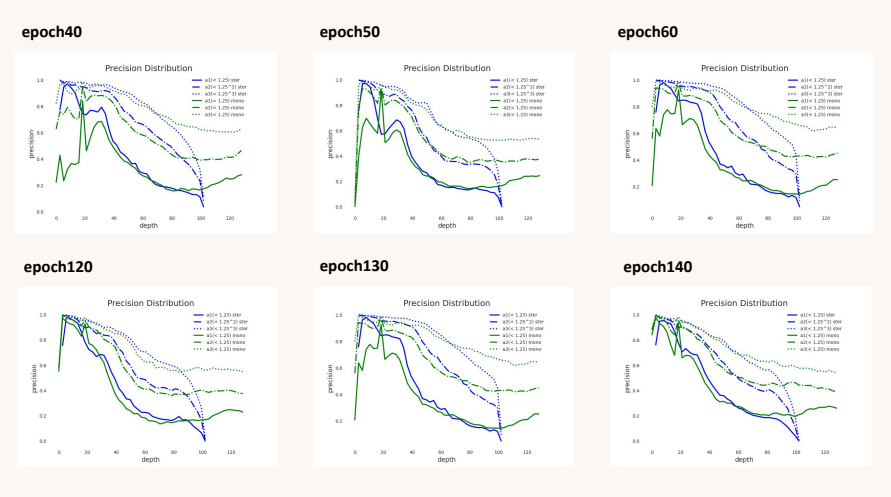


Figure 9: Accuracy statistics on CitySpike20K test set. The green lines and blue lines represent the monocular and stereo accuracies respectively.

And the second is that, the monocular strategy and stereo strategy share some distinct advantages to finish depth estimation task while loaded with spike data. We supply statistical results to prove our second motivation. On CitySpike20K dataset, we make a1, a2, a3 accuracy calculation in different depth intervals according to depth GT while evaluating our network. We transform the stereo disparities into depths, and count a1, a2, a3 accuracy for two branches respectively in the same metrics. Then we plot them in one coordinate. Figure 9 shows statistical results on test set. As seen, the stereo branch suffers from great accuracy decrease for far regions, while monocular branch still maintains certain reliability. Similarly, the stereo branch is more stable and accurate than the monocular branch for closer regions.