APPENDIX FOR UGDF

Anonymous authors
Paper under double-blind review

This is a copy of appendix which we have already attached one in the submitted paper. For your reading convenience, we sincerely suggest to review the original paper, many thanks!

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.

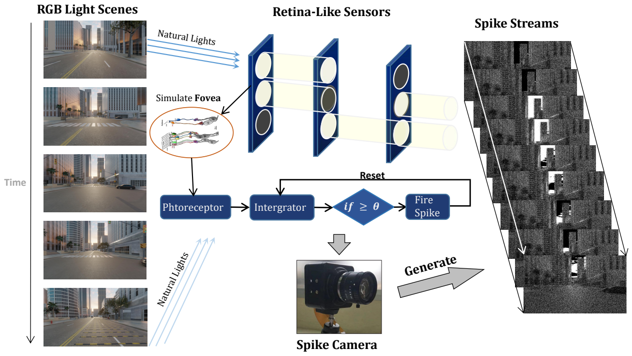

Figure 1: Spike camera is capable of of generating 2-bit spike streams via a retina-like process.

Table 1: 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**

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:

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

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

and RMSE defined:

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

we also introduce $RMSE\_log$ metric:

$$RMSE\_log = \sqrt{\frac{1}{N}\sum_{i=1}^{N}||log(\hat{D}_i) - log(D_i)||^2} \tag{3}$$

and Sq_Rel metric as here:

$$Sq\_Rel = \frac{1}{N}\sum_{i=1}^{N}\frac{||D_i - \hat{D}_i||^2}{D_i} \tag{4}$$

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 $j \in \{1, 2, 3\}$

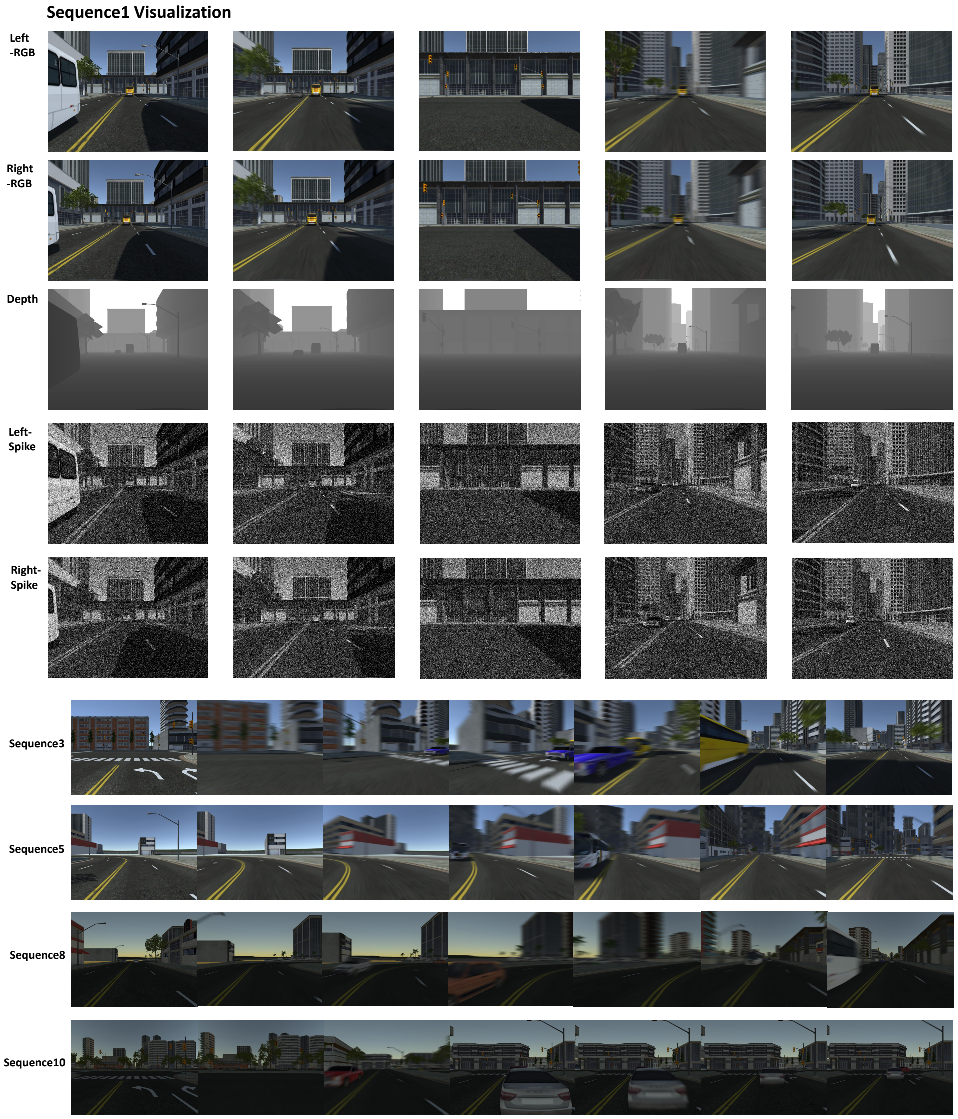

Figure 2: A visualization of our proposed CitySpike20k dataset. We generate it by Unity3D engine and simulate a vivid city environment along with dense depth maps and spike data.

$$aj \quad accuracy : \% \quad of \quad D_i \quad s.t. \quad max(\frac{\hat{D}_i}{D_i}, \frac{D_i}{\hat{D}_i}) = \delta < T = 1.25^j \tag{5}$$

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.

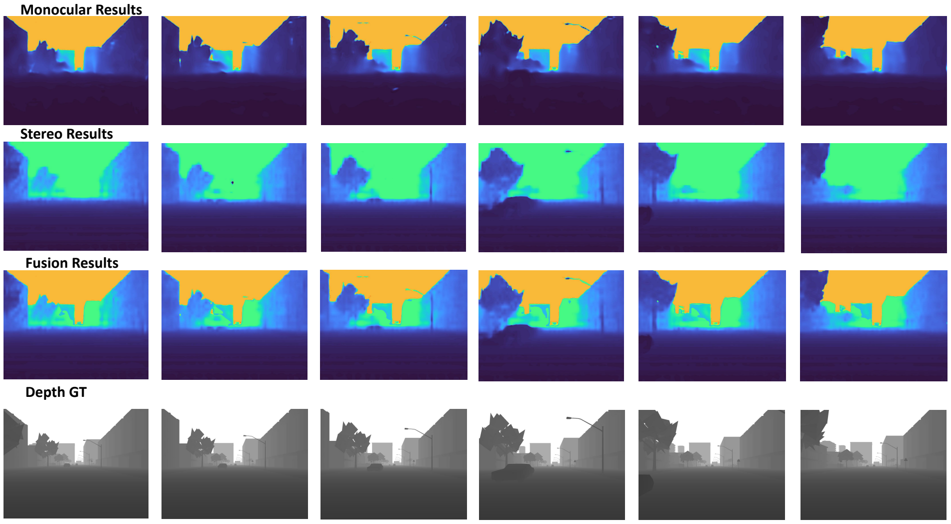

Figure 3: More prediction results on CitySpike20K dataset. As can be seen, the stereo estimation results and the monocular estimation results fuse efficiently by our framework

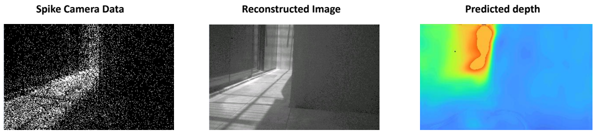

Figure 4: A visualization for Spike-Real dataset and prediction results from its test set.

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 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-

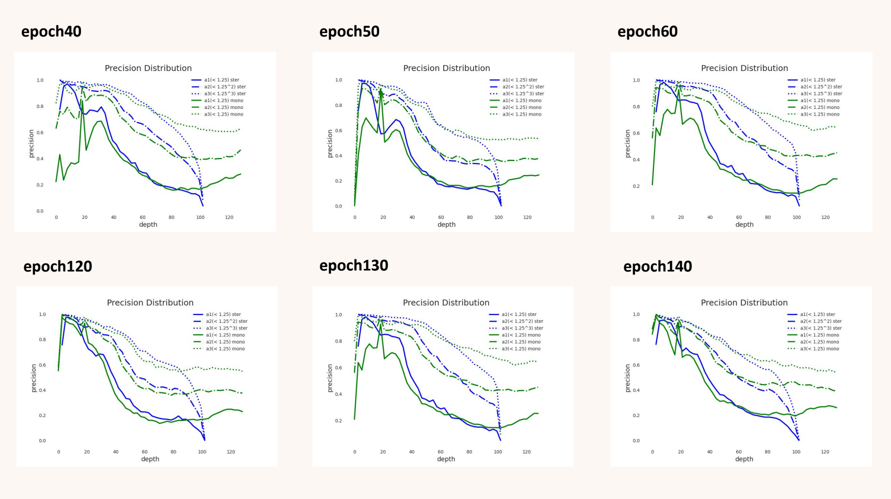

Figure 5: Accuracy statistics on CitySpike20K test set. The green lines and blue lines represent the monocular and stereo accuracies respectively.

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. 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.

REFERENCES

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.

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.

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.

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.

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.

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.

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.