Super-Resolution Reconstruction from Bayer-Pattern Spike Streams

Supplementary Material

1. More about Color Spike Camera

1.1. Camera Details

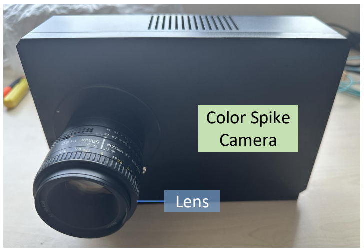


Figure 1. The color spike camera we used in the experiments to capture real-world Bayer-pattern spike stream.

In our experiments, we use a color spike camera (CSC) to capture some real-world Bayer-pattern (RGGB) spike streams for evaluation, with a sensor frequency of 20000Hz. The illustration of the CSC is shown in Fig. 1. According to the sensor manufacturer, the spatial resolution is 1000×1000. The pixel size is $17\mu m \times 17\mu m$. The data output speed is 500MHz. The threshold voltage of firing is 0.9V, with the reset time of $200ns$. Besides, we employ a 50mm 1:1.8D lens for the data capturing.

1.2. Compared with Other Cameras

Conventional digital cameras usually use a certain time window for exposure to accumulate photoelectric signals and compact them into a snapshot, with pixels of the sensor working synchronously. In contrast, CSC with ultra-high temporal resolution is a neuromorphic vision sensor that mimics the structure of human vision, the pixels of which work independently and asynchronously. When the accumulated signals reach a predetermined threshold θ, the pixel triggers a flag indicating firing a spike. Compared to most conventional cameras, CSC accumulates photons and fires spikes continuously to record dynamic scenes, resulting in a binary Bayer-pattern spike stream instead of a RAW image.

Event camera [1, 8–12] is also a kind of neuromorphic camera with high temporal resolution. Different from CSC which captures absolute light intensity via an "integrate-and-fire" mechanism, the event camera records relative light intensity changes. To be specific, event cameras are designed to generate event signals only when *light intensity changes* exceed a certain threshold. As a result, event cameras are sensitive to dynamic areas of the recorded scene.

Another sensor called *Quanta Image Sensor* (QIS) [2–6] is developed to discern individual photons through spatial and temporal oversampling. Benefiting from its single-photon sensitivity, the sensor has proven to be promising for applications in low-light conditions. Though with a similar data form (*i.e.*, binary data), the working mechanism and design motivation of QIS are different from CSC. In particular, the signal "1" or "0" for CSC means whether the amount of accumulated photoelectric signals exceeded a certain threshold. For QIS, it indicates the presence or absence of photons. In addition, QIS achieves single-photon sensing by minimizing readout noise for low-light imaging, while CSC is designed for high-speed imaging under normal illumination conditions.

2. More Method Details

2.1. ConvBlock and ResBlock

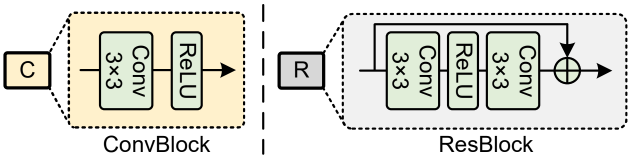


Figure 2. Illustration of the "ConvBlock" and "ResBlock".

In the structure of our proposed CSCSR network, there are some "ConvBlock"s and "ResBlock"s. The former indicates a convolution layer followed by a ReLU activation function, while the latter refers to the residual block in [7]. The structures of the two blocks are shown in Fig. 2.

2.2. Operation Details

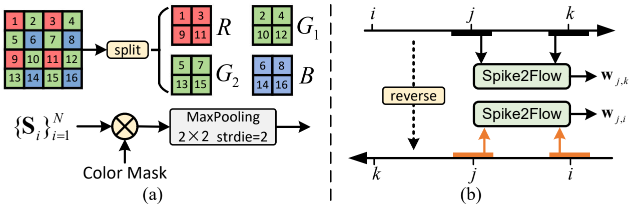


Figure 3. Illustration of the operations in our method. (a) The "split" operation. (b) The "reverse" operation.

In the figure of the overall architecture of our network, there are two operations, "split" and "reverse". We'll introduce details of the two operations as shown in Fig. 3. The "split" operation is to split out the spike sequence of each color channel from the Bayer-pattern spike stream, resulting in 4 downsampled spike sequences with a shape of $\frac{H}{2} \times \frac{W}{2}$. To implement the operation, we can first multiply the binary $\{\mathbf{S}_i\}_{i=1}^n$ by each color mask and use a 2×2

MaxPooling layer with stride 2 for downsampling. The "reverse" operation is to prepare the sequences for the motion estimation from time point j to i. To estimate optical flows, Spike2Flow needs two clips centered with the beginning and end time points as shown in Fig. 3 (b). To be specific, we can directly estimate the optical flow $\mathbf{w}_j^k$, while we need to reverse the sequences for another optical flow $\mathbf{w}_j^i$.

2.3. Channels for Joint Motion Estimation

To represent the Bayer-pattern spike stream clip, we split out three color channels R, G and B in the BSSR module. However, we split out four channels R, G_1, G_2 and B for joint motion estimation. As green pixels are denser, we jointly encode the two green channels for better use of color consistency in the BSSR module. For joint motion estimation, we need to extract each channel with the same spatial resolution and without missing pixels as shown in Fig. 3 (a) to meet the input requirements of the single-channel optical flow estimation method Spike2Flow [14]. Thus, we extract four channels from the Bayer-pattern spike stream clip.

2.4. Method Summary

To better introduce the pipeline of our proposed CSCSR method, we summarize it in Algorithm 1.

```
Algorithm 1: Color spike camera super-resolution
  Input: A clip of the LR spike stream $\{\mathbf{S}_i\}_{i=1}^N$
  Output: A HR color image
  Represent the input Bayer-pattern spike stream clip
    $\{\mathbf{S}_i\}_{i=1}^N$, resulting in the features of each color
    channel, $\mathbf{\Omega}^R$, $\mathbf{\Omega}^G$ and $\mathbf{\Omega}^B$;
  Jointly estimate motion from $\{\mathbf{S}_i\}_{i=1}^N$, producing
    the optical flows $\mathbf{w}_{j,i}$ and $\mathbf{w}_{j,k}$ from the middle
    time point to the first and the last time points;
  Get temporal-pixel features $\hat{\mathbf{T}}^R$, $\hat{\mathbf{T}}^G$ and $\hat{\mathbf{T}}^B$ from
    the encoded features $\mathbf{\Omega}^R$, $\mathbf{\Omega}^G$ and $\mathbf{\Omega}^B$, guided by
    the estimated optical flows $\mathbf{w}_{j,i}$ and $\mathbf{w}_{j,k}$;
  Integrate the features of each color channel $\hat{\mathbf{T}}^R$, $\hat{\mathbf{T}}^G$
    and $\hat{\mathbf{T}}^B$ to reconstruct the HR color image.
```

3. Parameter Settings

3.1. Number of Input Spike Frames

As studied in [15], the reconstruction performance converges when the number of input spike frames N is sufficiently large. More frames bring greater computational complexity but little performance improvement, so they set it to 41 according to the ablation studies. Besides, the number of input spike frames is also set to 41 in a representative learning-based spike camera reconstruction method Spk2ImgNet [13]. As a result, we also set the number of input spike frames to 41 in our method.

Stride	Length	PSNR ↑	SSIM ↑	Time ↓
1	3	33.34dB	0.9116	2.032s
	5	33.34dB	0.9109	1.973s
	7	33.35dB	0.9121	1.903s
	9	33.38dB	0.9117	1.794s
	11	33.38dB	0.9120	1.722s
	13	33.37dB	0.9123	1.662s
	15	33.39dB	0.9123	1.600s
	17	33.36dB	0.9126	1.548s
	19	33.34dB	0.9123	1.456s
	21	33.35dB	0.9117	1.420s
	23	33.33dB	0.9116	1.325s
3	5	33.34dB	0.9105	1.086s
	11	33.39dB	0.9121	1.032s
	17	33.34dB	0.9118	0.977s
9	5	32.95dB	0.9046	0.842s

Table 1. Ablation studies of sliding window length w and stride s on the REDS-based evaluation dataset.

3.2. Length and Stride of Sliding Window

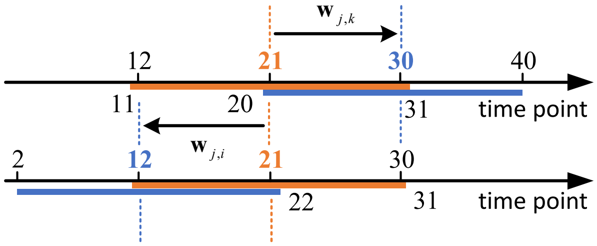


Figure 4. Optical flows estimated from the middle time point to the end time point and the beginning time point.

The middle time point corresponds to the 21st spike frame. According to the setting of the motion estimation method [14], the number of spike frames for each time point is 21, and the interval between time points should be a multiple of 3. To estimate the motion from the middle time point to the last time point, the right boundary of the last time point farthest from the middle time point is $21+3\times3+10 = 40 < N = 41$. Similarly, the left boundary of the first time point farthest from the middle time point is $21 - 3 \times 3 - 10 = 2 > 1$. Therefore, the index of the last time point and the first time point is 30 and 12 as shown in Fig. 4. For the following temporal pixel search, we need to cover the first, middle and last time points in the BSSR module. The channel number of encoded features as well as the number of time points N' in BSSR can be obtained by

$$N' = \frac{N - w + s}{s}, \tag{1}$$

where $N \in \mathbb{Z}^+$ denotes the number of spike frames, w ($w = 2k+1, k \in \mathbb{Z}^+$) and $s \in \mathbb{Z}^+$ denotes the length and stride of the temporal sliding window. To meet the requirements of covering the three time points and $N' \in \mathbb{Z}^+$, we get $s \in \{1, 3, 9\}$. Then we have three settings: $\{s = 1, w = 2k+1 \leq 23, k \in \mathbb{Z}^+\}$, $\{s = 3, w \in \{5, 11, 17\}\}$, and $\{s = 9, w = 5\}$. To investigate the settings, we perform ablation studies on the parameters as shown in Table 1. According to the results, most cases share similar performance, which shows the stability of our network. In particular, there is a noticeable performance drop in the last case with $w = 5$ and $s = 9$. This is due to some frames being skipped when the stride s is larger than the length w. Finally, we set the window length w and stride s to 11 and 3, considering the balance of performance and running time.

4. Appendix of Experiments

4.1. Simulator Summary

Besides the figure of our CSC simulator in the paper, we also summarize the pipeline in Algorithm 2 for a better introduction. The input of the simulator consists of a sequence of video frames, spike firing threshold, super-resolution scale and pattern of CFA (*e.g.*, RGGB). The output is a LR Bayer-pattern spike stream-HR color image pair. The codes of the simulator will be publicly available.

4.2. Computational Complexity

To better compare the methods, we perform the comparison of computational complexity on the REDS-based evaluation dataset in Table 2. With competitive running time, our method achieves the best performance.

Method	Parameters ↓	Time ↓	PSNR ↑
TFI+TSCNN	1.45M	1.244s	29.24dB
TFP+TSCNN	1.45M	1.181s	27.43dB
TFI+Real-RawVSR	4.48M	0.363s	31.03dB
TFP+Real-RawVSR	4.48M	0.496s	30.81dB
3DRI+SwinIR	11.75M	20.928s	31.13dB
3DRI+BasicVSR	6.29M	16.781s	31.60dB
VidarSR	10.16M	4.612s	30.81dB
VidarSR*	10.16M	13.820s	30.27dB
SpikeSR-Net	2.64M	2.4265	32.38dB
SpikeSR-Net*	2.64M	10.043s	29.66dB
CSCSR (ours)	5.40M	1.032s	33.39dB

Table 2. Computation complexity comparison on the REDS-based evaluation dataset, including the comparison of trainable parameter number, running time and PSNR performance.

4.3. Assembled Motion Estimation

In our ablation study, we replace our joint motion estimation strategy with an assembled motion estimation. To better introduce the strategy in Case (E), we present the illustration in Fig. 5. To be specific, we first split out spike sequences of each channel due to the input requirements of the single-channel optical flow estimation method [14]. After the independent motion estimation of each sequence, we assemble the optical flows $\mathbf{w}_{j,i}^{R}$, $\mathbf{w}_{j,i}^{G_1}$, $\mathbf{w}_{j,i}^{G_2}$ and $\mathbf{w}_{j,i}^{B}$ to $\mathbf{w}_{j,i}$ according to the color layout as shown in Fig. 5. Similarly, we obtain $\mathbf{w}_{j,k}$ by the assembly of $\mathbf{w}_{j,k}^{R}$, $\mathbf{w}_{j,k}^{G_1}$, $\mathbf{w}_{j,k}^{G_2}$ and $\mathbf{w}_{j,k}^{B}$.

Algorithm 2: Color spike camera simulator

```
Input: A sequence of video frames, spike firing
       threshold $\theta$, super-resolution scale $s$ and
       pattern of color filter array
Output: A clip of LR Bayer-pattern spike stream
        $\{\mathbf{S}_i\}_{i=1}^{N}$ and corresponding HR image $\mathbf{H}$
Generate $N$ latent intensity frames from the input
  video frames by a frame interpolation method;
for $i \leftarrow 1$ to $N$ do
    Accumulate signals from the corresponding
      color channel of the $i$-th latent intensity frame
      $\mathbf{I}_i$ according to the CFA pattern and scale $s$,
      resulting in the accumulated signals $\mathbf{A}_i$;
    for each pixel $(x, y)$ do
        if $\mathbf{A}_i(x, y) \geq \theta$ then
            $\mathbf{S}_i(x, y) = 1$; // Fire a spike
            $\mathbf{A}_i(x, y) = 0$; // Reset
        else
            $\mathbf{S}_i(x, y) = 0$; // Fire no spike
        end
    end
    if $i = \frac{N+1}{2}$ then
        $\mathbf{H} = \mathbf{I}_i$; // Set the latent
            intensity frame of the
            middle time point as the
            ground truth HR image
    end
end
```

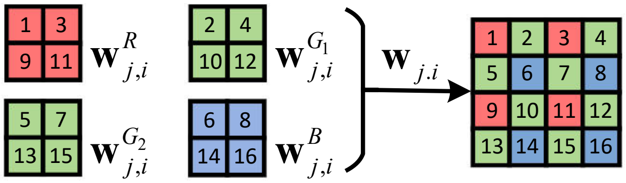


Figure 5. Illustration of the assembled motion estimation strategy in Case (E) of our Ablation Study.

4.4. More Visual Results

To further demonstrate the performance of our proposed CSCSR method, we supplement more visual comparison re-

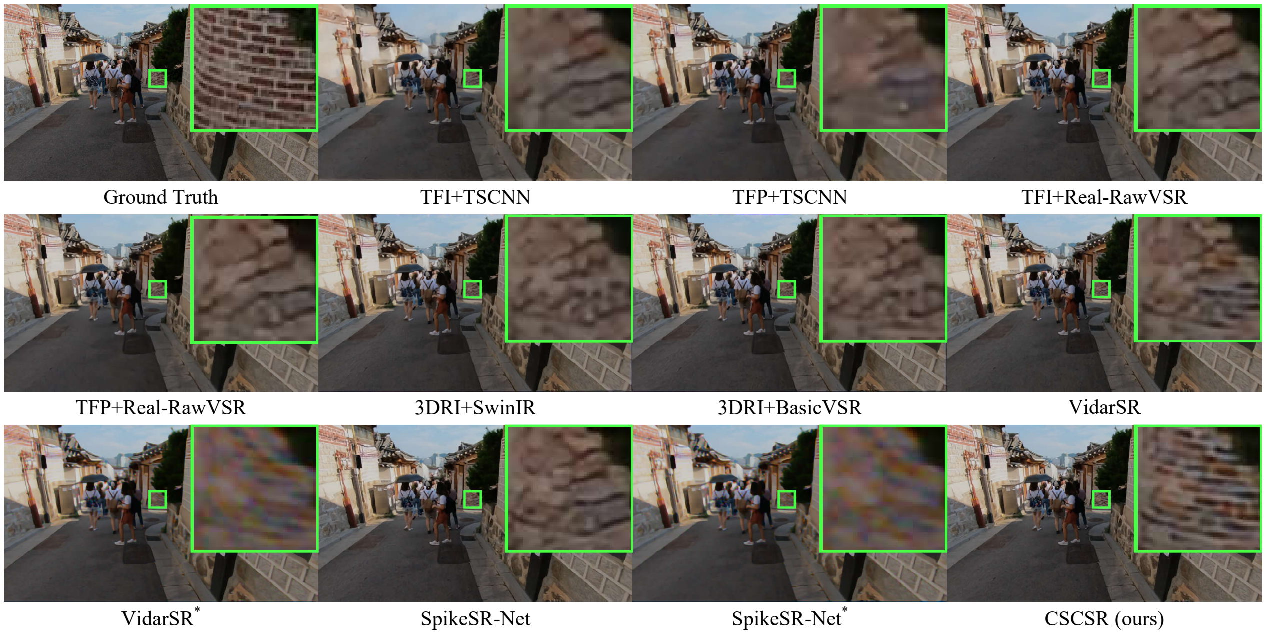

Ground Truth
TFI+TSCNN
TFP+TSCNN
TFI+Real-RawVSR
TFP+Real-RawVSR
3DRI+SwinIR
3DRI+BasicVSR
VidarSR
VidarSR*
SpikeSR-Net
SpikeSR-Net*
CSCSR (ours)

Figure 6. Visual comparison results of color spike camera super-resolution (×4). Please enlarge the figure for better comparison.

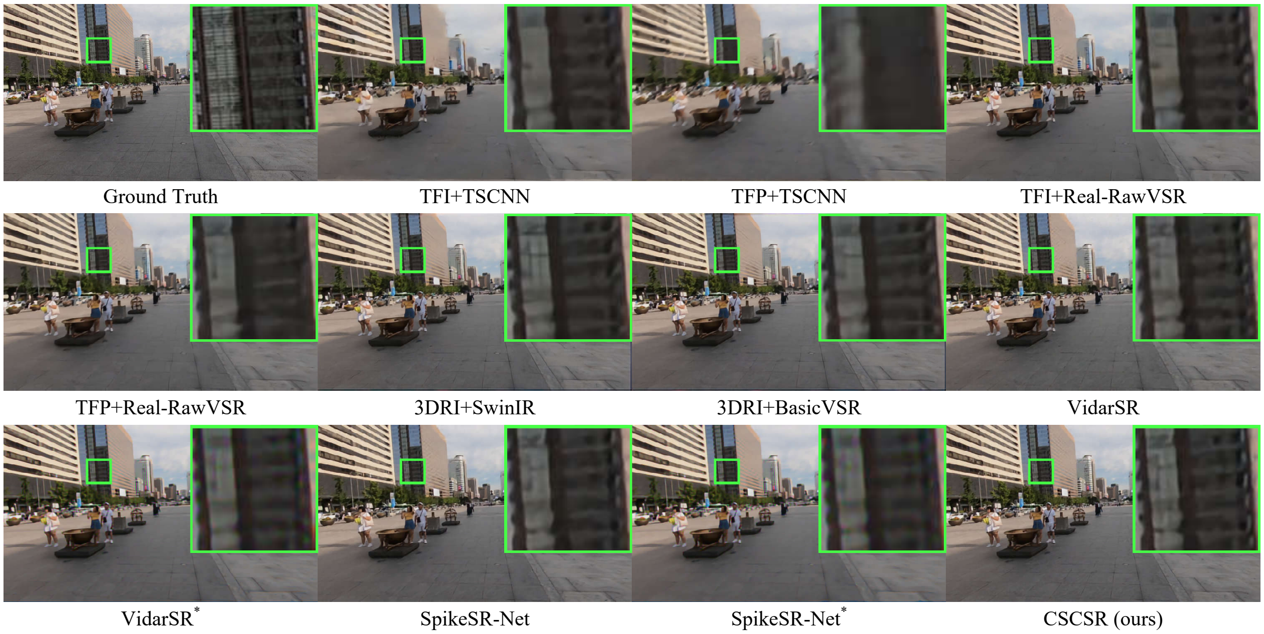

Ground Truth
TFI+TSCNN
TFP+TSCNN
TFI+Real-RawVSR
TFP+Real-RawVSR
3DRI+SwinIR
3DRI+BasicVSR
VidarSR
VidarSR*
SpikeSR-Net
SpikeSR-Net*
CSCSR (ours)

Figure 7. Visual comparison results of color spike camera super-resolution (×4). Please enlarge the figure for better comparison.

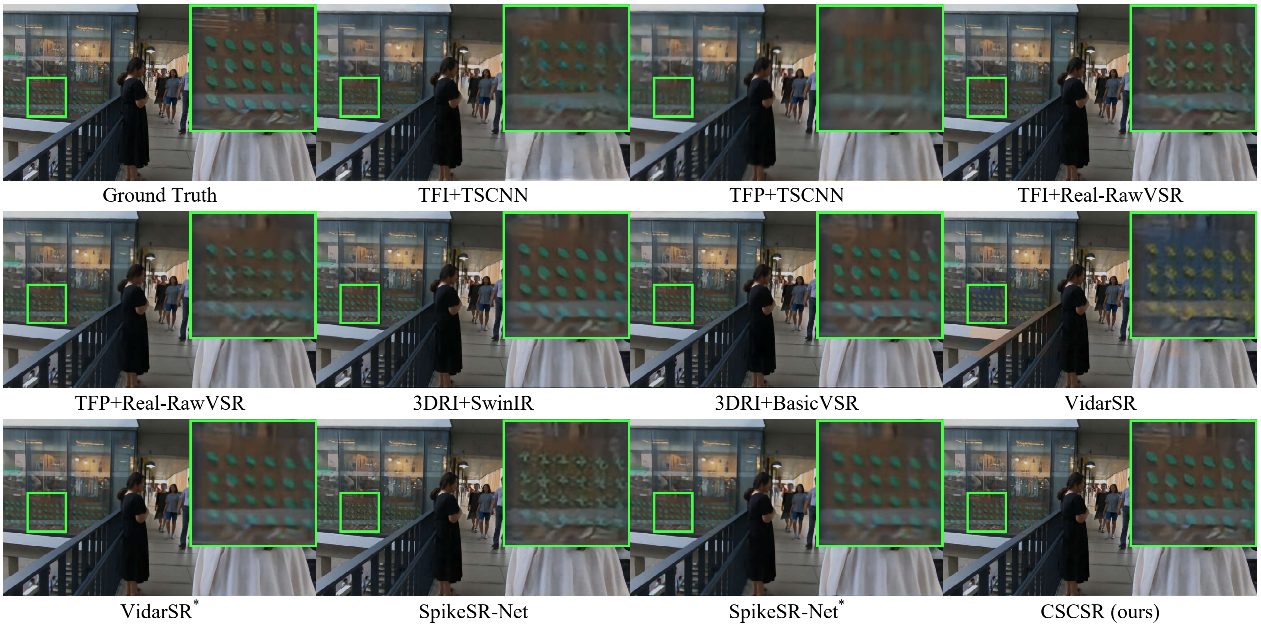
sults as shown in the following figures. Compared to other methods, our CSCSR network produces HR color images with better details and visual quality. Please enlarge the figures for better visual comparison.

References

[1] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 1

[2] Claudio Bruschini, Samuel Burri, Scott Lindner, Arin C Ulku, Chao Zhang, I Michel Antolovic, Martin Wolf, and Edoardo Charbon. Monolithic spad arrays for high-performance, time-resolved single-photon imaging. In *2018 International Conference on Optical MEMS and Nanophotonics (OMN)*, pages 1–5. IEEE, 2018. 1

[3] Neale AW Dutton, Istvan Gyongy, Luca Parmesan, Salvatore

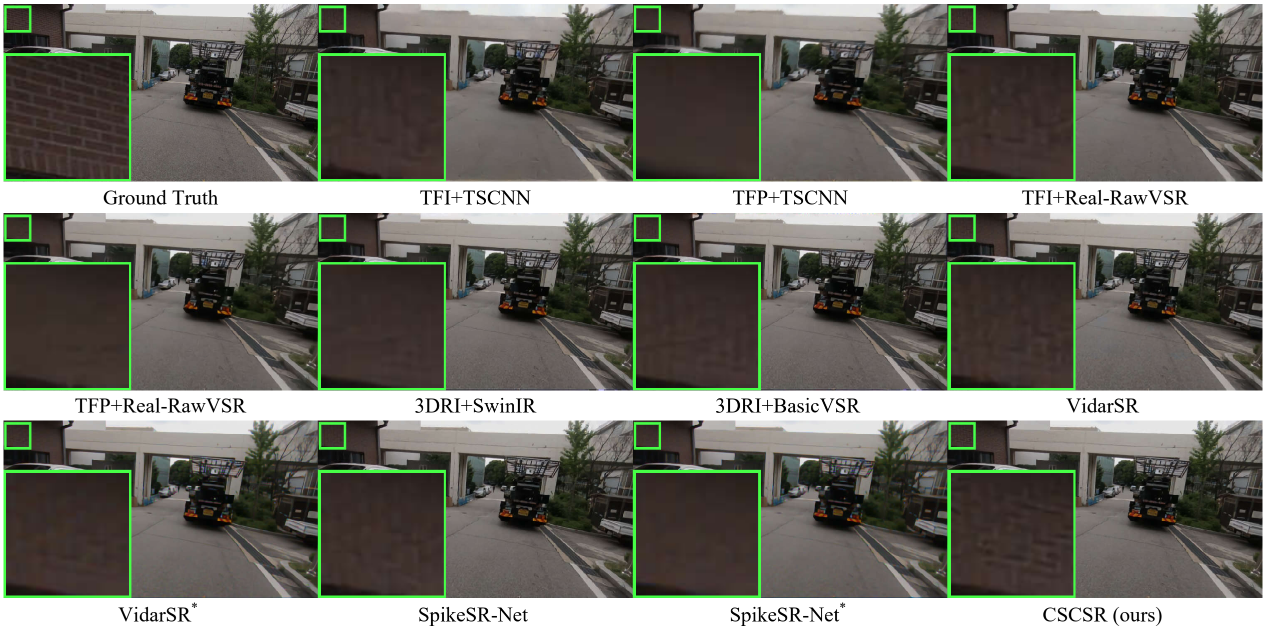


Ground Truth TFI+TSCNN TFP+TSCNN TFI+Real-RawVSR

TFP+Real-RawVSR 3DRI+SwinIR 3DRI+BasicVSR VidarSR

VidarSR* SpikeSR-Net SpikeSR-Net* CSCSR (ours)

Figure 8. Visual comparison results of color spike camera super-resolution (×4). Please enlarge the figure for better comparison.



Ground Truth TFI+TSCNN TFP+TSCNN TFI+Real-RawVSR

TFP+Real-RawVSR 3DRI+SwinIR 3DRI+BasicVSR VidarSR

VidarSR* SpikeSR-Net SpikeSR-Net* CSCSR (ours)

Figure 9. Visual comparison results of color spike camera super-resolution (×4). Please enlarge the figure for better comparison.

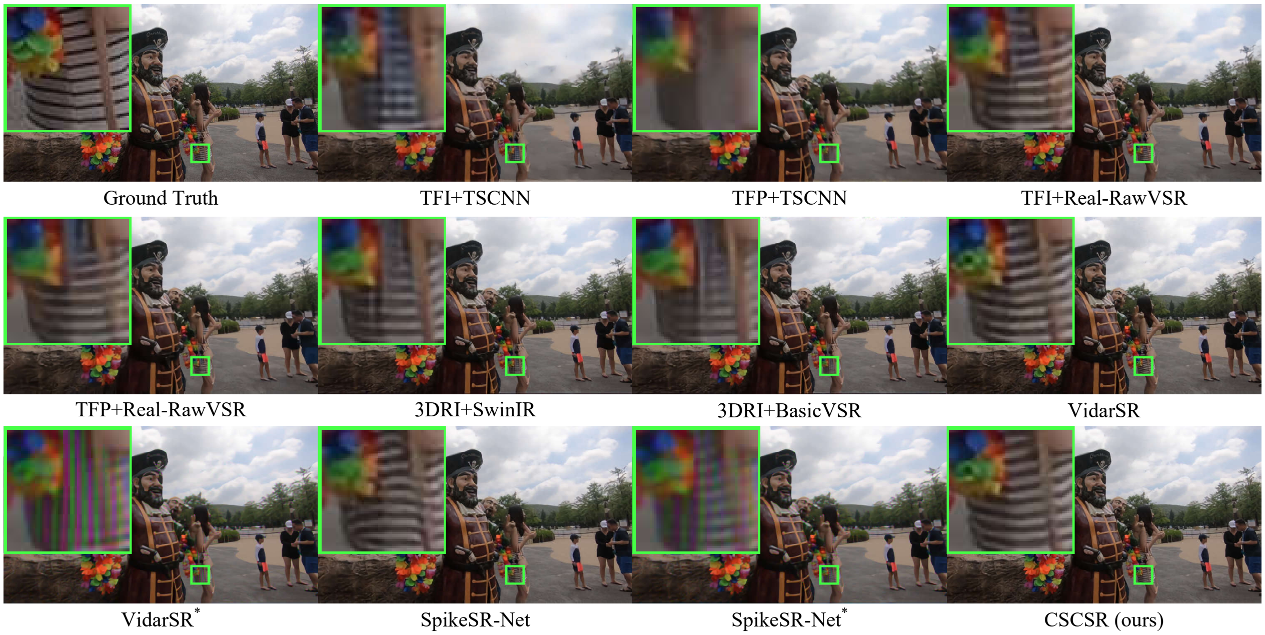
Gnecchi, Neil Calder, Bruce R Rae, Sara Pellegrini, Lindsay A Grant, and Robert K Henderson. A spad-based qvga image sensor for single-photon counting and quanta imaging. *IEEE Transactions on Electron Devices*, 63(1):189–196, 2015.

[4] NA Dutton12, Luca Parmesan12, Salvatore Gnecchi12, Istvan Gyongy, Neil Calder, Bruce R Rae, Lindsay A Grant, and Robert K Henderson. Oversampled itof imaging techniques using spad-based quanta image sensors. In *Proc. Int. Image Sensor Workshop*, pages 170–173, 2015.

[5] Abhiram Gnanasambandam, Omar Elgendy, Jiaju Ma, and Stanley H Chan. Megapixel photon-counting color imaging using quanta image sensor. *Optics express*, 27(12):17298–17310, 2019.

[6] Istvan Gyongy, Neale Dutton, Parmesan Luca, and Robert Henderson. Bit-plane processing techniques for low-light, high speed imaging with a spad-based qis. In *International Image Sensor Workshop*, 2015. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE*

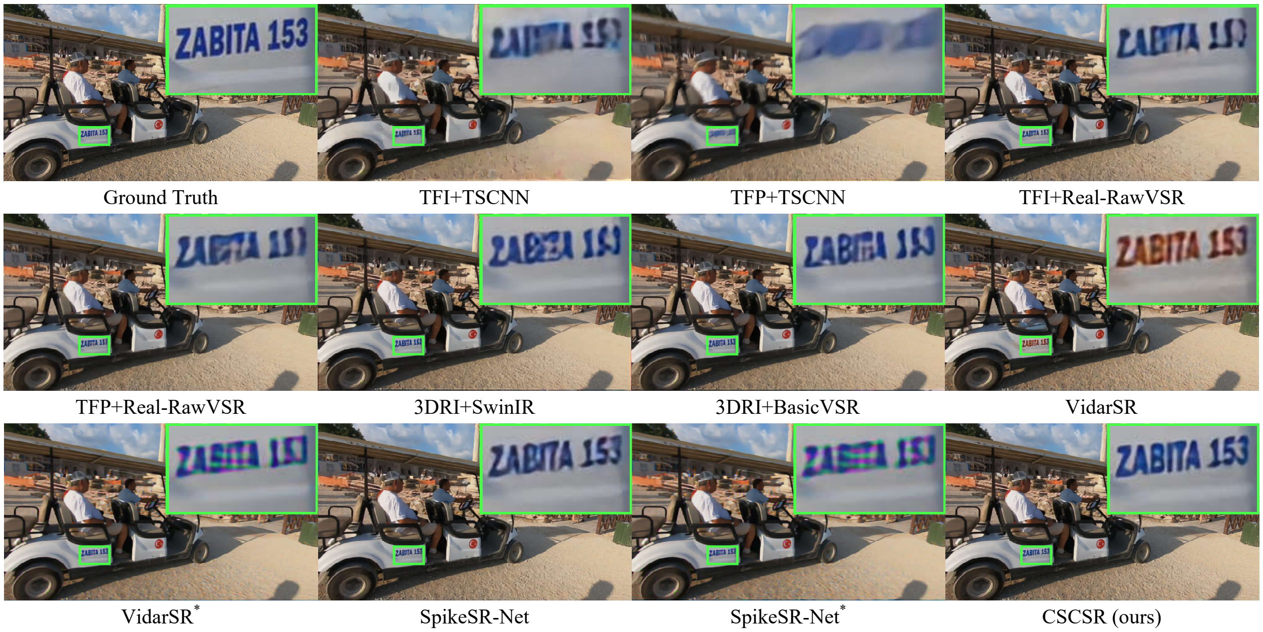


Ground Truth TFI+TSCNN TFP+TSCNN TFI+Real-RawVSR

TFP+Real-RawVSR 3DRI+SwinIR 3DRI+BasicVSR VidarSR

VidarSR* SpikeSR-Net SpikeSR-Net* CSCSR (ours)

Figure 10. Visual comparison results of color spike camera super-resolution (×4). Please enlarge the figure for better comparison.



Ground Truth TFI+TSCNN TFP+TSCNN TFI+Real-RawVSR

TFP+Real-RawVSR 3DRI+SwinIR 3DRI+BasicVSR VidarSR

VidarSR* SpikeSR-Net SpikeSR-Net* CSCSR (ours)

Figure 11. Visual comparison results of color spike camera super-resolution (×4). Please enlarge the figure for better comparison.

Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016. 1

[8] Jing Huang, Menghan Guo, and Shoushun Chen. A dynamic vision sensor with direct logarithmic output and full-frame picture-on-demand. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4. IEEE, 2017. 1

[9] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43 (2):566–576, 2008.

[10] Martin Litzenberger, Christoph Posch, D Bauer, Ahmed Nabil Belbachir, P Schon, B Kohn, and H Garn. Embedded vision system for real-time object tracking using an asynchronous transient vision sensor. In *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*, pages 173–178. IEEE, 2006.

[11] Diederik Paul Moeys, Federico Corradi, Chenghan Li,

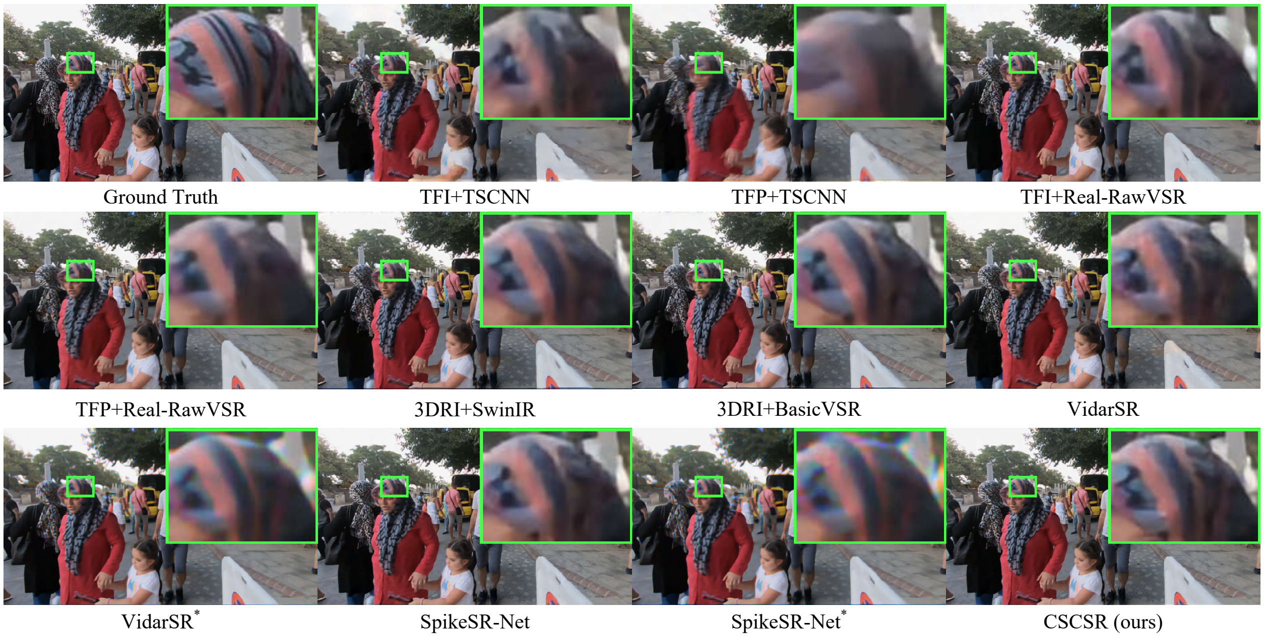


Figure 12. Visual comparison results of color spike camera super-resolution (×4). Please enlarge the figure for better comparison.

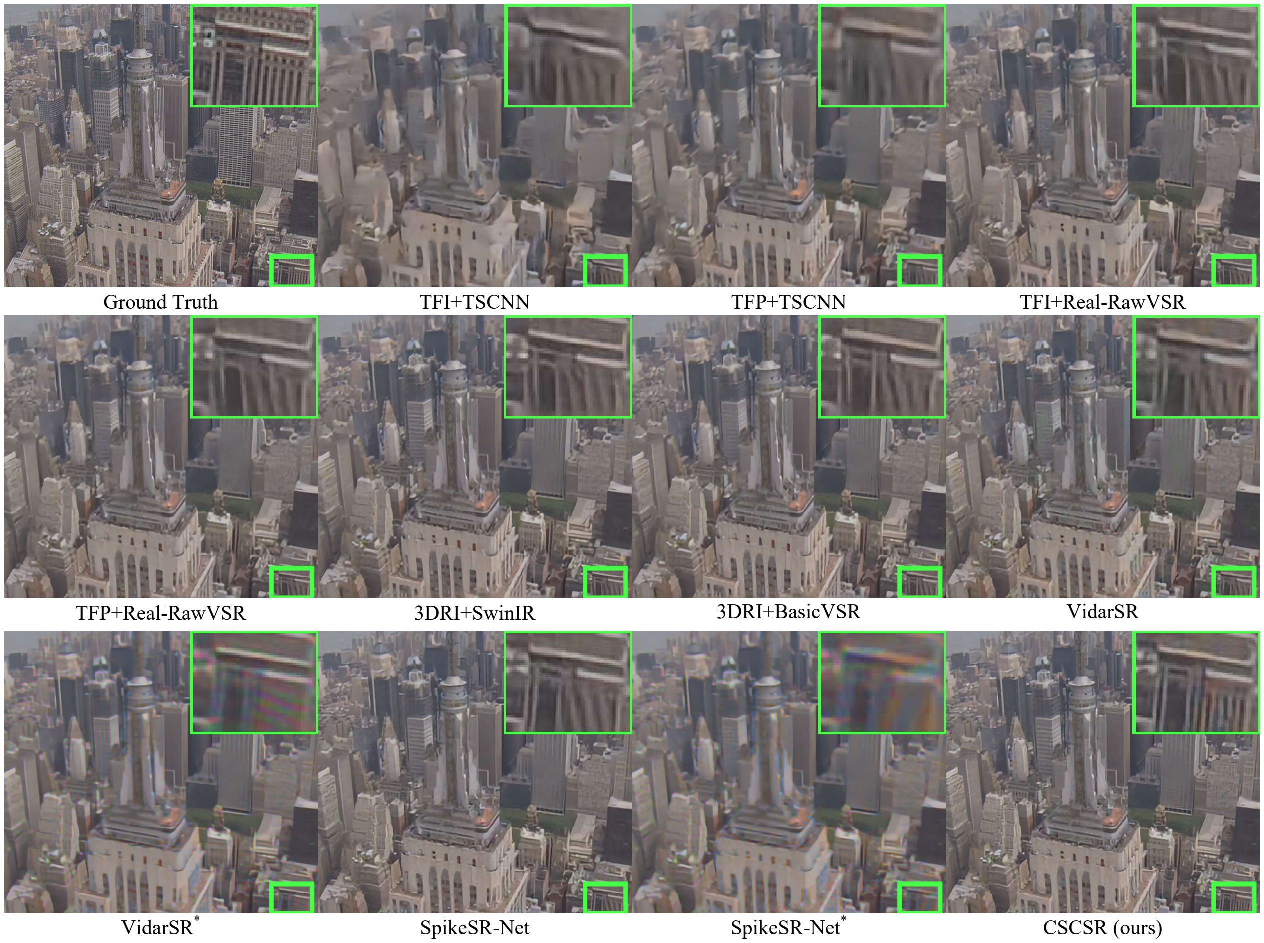
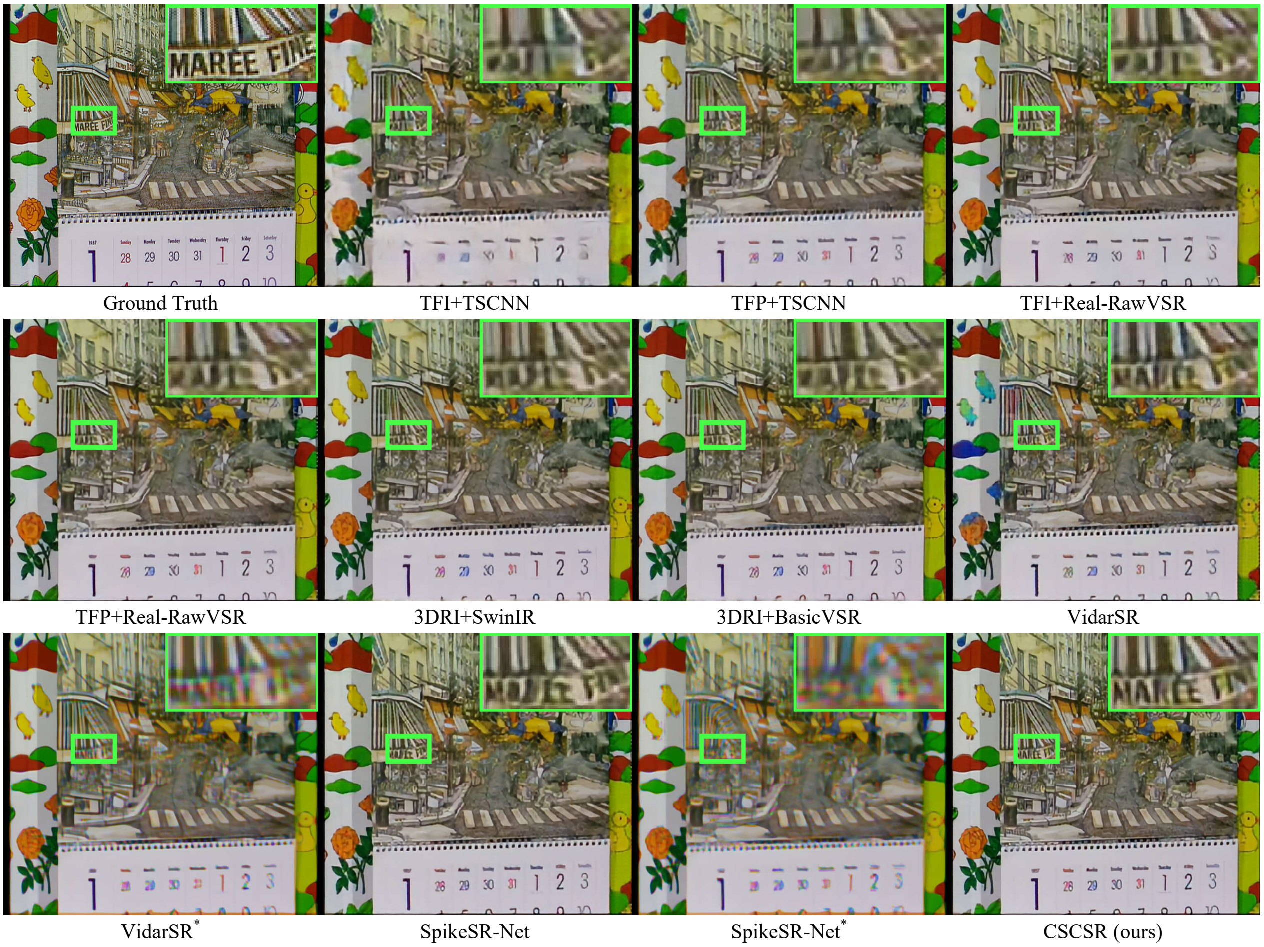


Figure 13. Visual comparison results of color spike camera super-resolution (×4). Please enlarge the figure for better comparison.



Ground Truth
TFI+TSCNN
TFP+TSCNN
TFI+Real-RawVSR
TFP+Real-RawVSR
3DRI+SwinIR
3DRI+BasicVSR
VidarSR
VidarSR*
SpikeSR-Net
SpikeSR-Net*
CSCSR (ours)

Figure 14. Visual comparison results of color spike camera super-resolution (×4). Please enlarge the figure for better comparison.

Simeon A Bamford, Luca Longinotti, Fabian F Voigt, Stewart Berry, Gemma Taverni, Fritjof Helmchen, and Tobi Delbruck. A sensitive dynamic and active pixel vision sensor for color or neural imaging applications. *IEEE Transactions on Biomedical Circuits and Systems*, 12(1):123–136, 2017.

[12] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2010. 1

[13] Jing Zhao, Ruiqin Xiong, Hangfan Liu, Jian Zhang, and Tiejun Huang. Spk2ImgNet: Learning to reconstruct dynamic scene from continuous spike stream. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11996–12005, 2021. 2

[14] Rui Zhao, Ruiqin Xiong, Jing Zhao, Zhaofei Yu, Xiaopeng Fan, and Tiejun Huang. Learning optical flow from continuous spike streams. *Advances in Neural Information Processing Systems*, 35:7905–7920, 2022. 2, 3

[15] Rui Zhao, Ruiqin Xiong, Jian Zhang, Zhaofei Yu, Shuyuan Zhu, Lei Ma, and Tiejun Huang. Spike camera image reconstruction using deep spiking neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 2