

QR-code Reconstruction from Event Data via Optimization in Code Subspace

Jun Nagata¹ Yusuke Sekikawa² Kosuke Hara² Teppei Suzuki² Aoki Yoshimitsu¹
¹Keio University, ²Denso IT Laboratory

jnagata@aoki-medialab.jp, {ysekikawa, khara, tsuzuki}@d-itlab.co.jp, aoki@elec.keio.ac.jp

Abstract

We propose an image reconstruction method from event data, assuming the target images belong to a prespecified class like QR codes. Instead of solving the reconstruction problem in the image space, we introduce a code space that covers all the noiseless target class images and solves the reconstruction problem on it. This restriction enormously reduces the number of optimizing parameters and makes the reconstruction problem well posed and robust to noise. We demonstrate fast and robust QR-code scanning in difficult, high-speed scenes with industrial high-speed cameras and other reconstruction methods.

1. Introduction

QR codes are widely used in factory automation and production control. When a target object is moving (e.g., on a belt conveyor), motion blurring occurs and QR-code scanning often fails. A common solution for preventing motion blurring is to shorten the shutter time by preparing a light source or introducing a high-speed camera. However, it is expensive to install and maintain the facilities; moreover, the amount of redundant data increases, and frequent memory access and high computational power are required.

Event-based cameras [4] are bio-inspired sensors that asynchronously report per-pixel intensity changes instead of acquiring images at a constant rate. As shown in Fig. 1a, when a logarithmic intensity changes above a certain threshold θ , an event with precise timestamp occurs. The event also contains a binary polarity that indicates brighter or darker change. The advantages of event cameras are their high temporal resolution (μs order), high dynamic range (130 dB), and low power consumption [3, 10]. Even in a scene moving at high speed as shown in Fig. 1b, event data can be acquired without blurring. In addition, an event in which the luminance change information is encoded in binary polarity is sufficient and suitable for estimating a binary QR code.

Since the QR code is a black-and-white luminance pattern, it is necessary to restore the luminance to estimate the

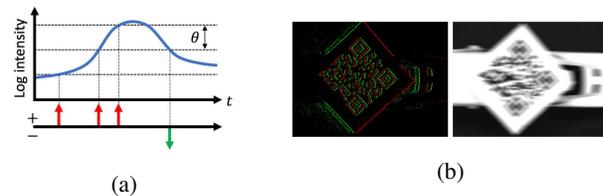


Figure 1: (a) Working principle of the event-based camera. The blue graph shows the temporal change of logarithmic intensity at one pixel. The red and green arrows indicate the positive and negative events, respectively. (b) Comparison of the event camera and industrial camera (330 fps) in the high-speed scene. Event data are integrated for 0.25 ms.

QR code from the event data. However, complete image reconstruction from spatially sparse event data is an ill-posed problem. In this paper, we propose a fast, robust reconstruction method by optimizing in a code space more constrained than the image space. The image representation of the QR code and affine transformation significantly reduce parameters, making reconstruction a well-posed problem and robust to noise. Our main contributions are summarized as follows:

- The image is modeled as generated by affine transformation the QR code, and it succeeds in significantly reducing the optimized parameters compared with pixel-wise reconstruction. The sampling described in [12] was applied to our model to formulate the gradients in terms of QR code. We proposed a method of estimating the QR code quickly and robustly from event data by optimizing with the code space more constrained than the image space.
- We captured the QR code moving at various speeds and attached at various angles with an event camera, and we constructed a new dataset called QR-code Event Camera Dataset (QRECD). We demonstrated that it is possible to read the QR code, even in high-speed moving scenes that were difficult with an industrial high-speed camera (330 fps) and other restoration methods.

2. Related Work

Image reconstruction from event data can be performed, for example, for generating video with a high dynamic range at high frame rates [7, 11, 14–16]. However, our goal is not to generate a video, but rather, to recover a QR code quickly from a short event sequence.

Barua *et al.* [2] proposed the method of patch-based image reconstruction from event data by using learned a dictionary for mapping integrated events to image gradients. Kim *et al.* [8] performed camera tracking and reconstructing two-dimensional (2D) panoramic image in the rotational motion-only scene from pure events using the extended Kalman filter. In these two methods, the image is restored from the gradient image by Poisson reconstruction. In a later work, Kim *et al.* extended their study to 6 degree of freedom (DOF) tracking and depth estimation [9]. Bardow *et al.* [1] estimated dense images and optical flow without depending on the camera motion by optimizing the cost function combining the event data term, optical flow constraints, and spatio temporal smoothness regularization. In these two methods, dense images are reconstructed from spatially sparse event data using smoothness regularization.

Despite its advantages, smoothness regularization requires many iterations due to the event’s sparsity. In addition, code boundaries are not clearly estimated with smoothness regularization due to the high noise level of the event camera. Due to the nature of event camera, no event occurs when the motion and luminance gradients are orthogonal. At this time, the codes parallel to the motion direction are uniformly estimated in surrounding pixels with smoothness regularization.

To reconstruct the intensity from a spatially sparse and temporally short event sequence, it is necessary to apply constraints in accordance with the physical model. In the QR code scene we focused on, there is a prespecified base of a QR code in the image, and it is only necessary that the luminance information be restored for that part. Therefore, in our model that the image is generated by the QR code and affine transformation, and it is optimized in the restricted code space instead of the image space.

3. Proposed Method

3.1. Event data representation

An event-based camera outputs an event $e_i \doteq (t_i, \mathbf{x}_i, \rho_i)$ consisting of the microsecond timestamp t_i , pixel coordinates \mathbf{x}_i , and polarity $\rho_i \in \{+1, -1\}$, indicating brighter or darker changes when the logarithmic intensity changes above a threshold θ at a certain pixel $\mathbf{x} = (x, y)$. The input data comprise event $\mathcal{E} \doteq \{e_i\}$, where $t_i \in [t_{\text{ref}}, t_{\text{ref}} + \Delta t]$ over a duration of Δt from reference time t_{ref} .

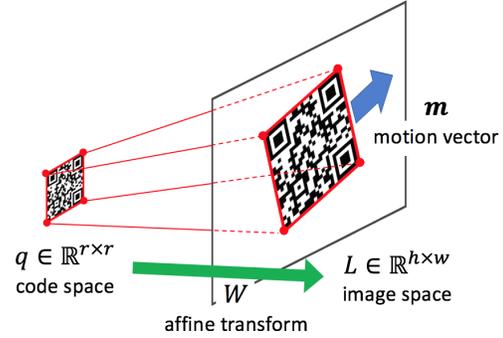


Figure 2: Parameterization. The intensity image $L \in \mathbb{R}^{h \times w}$ is generated by affine transforming the QR code $q \in \mathbb{R}^{r \times r}$. The optimization parameters are drastically reduced from image size (e.g., 480×640) to QR-code size (e.g., 25×25), with six affine parameters. The motion vector is common to all pixels, and the number of parameters is 2.

3.2. Code space and parameterization

As shown in Fig. 2, there are three parameters to be estimated, namely, the QR code $q \in \mathbb{R}^{r \times r}$ (r is the number of codes on one side of the QR code), motion vector common to all pixels $m \in \mathbb{R}^2$, and affine parameters representing the state of the QR code on the image $W \in \mathbb{R}^6$.

$\mathbb{R}^{r \times r}$ is the code space we defined. The intensity image $L \in \mathbb{R}^{h \times w}$ is modeled to be generated by affine transforming the QR code $q \in \mathbb{R}^{r \times r}$, and it allows the optimization parameters to be drastically reduced from the image size (e.g., 480×640) to the QR-code size (e.g., 25×25), with six affine parameters. The number of parameters of the QR code is actually smaller than $r \times r$ because there are known patterns in the specification.¹ By optimizing in strongly constrained code space while maintaining expressiveness, the QR code can be estimated robustly from event data obtained in the image domain. Although the QR code is a binary code, the code space is defined in the real number space, since it is evaluated for consistency with event data in the image space. The QR code is optimized with continuous values in the real number space, and binarized with threshold 0 to obtain the final QR code. In addition, assuming a situation where the QR code on the belt conveyor is photographed from above, the geometrical transformation is the affine transformation and the motion vector is common to all pixels.

¹QR-code Standardization: <https://www.qrcode.com/en/about/standards.html>

3.3. Cost function

The proposed cost function is as follows:

$$\operatorname{argmin}_{q, \mathbf{m}, W} \sum_{\mathbf{x}} \left\| (\nabla L(\mathbf{x}) \cdot \mathbf{m}) \Delta t + \sum_i \theta \rho_i \delta(\mathbf{x} - \mathbf{x}'_i) \right\|^2, \quad (1)$$

$$\begin{aligned} \text{where } \mathbf{x}'_i &= \mathbf{x}_i - \mathbf{m}(t_i - t_{\text{ref}}), & (2) \\ L &= f_W(q), \quad f_W: \mathbb{R}^{r \times r} \rightarrow \mathbb{R}^{h \times w}. & (3) \end{aligned}$$

This cost function is based on the occurrence model of the event when the object moves in a uniform linear motion. Due to the μs -order high temporal resolution of the event data, the time width of the input sequence can be set to be arbitrarily small, and the assumption of uniform linear motion is sufficiently satisfied. Events that occur when an edge moves via uniform linear motion \mathbf{m} are schematically described in Fig. 3. Events occur by drawing a linear trajectory, and the integration of events on the trajectory is equivalent to the temporal change of luminance due to the movement \mathbf{m} of the edge. The events trajectory integration can be expressed by warping events with a motion vector \mathbf{m} and summing it up in the time direction [5, 6]. Then, the delta function $\delta(\mathbf{x} - \mathbf{x}'_i)$ is smoothly approximated by the Gaussian function $\mathcal{N}(\mathbf{x}; \mathbf{x}'_i, \epsilon^2)$ so that the events occurring at the same edge are associated.

3.4. Formulation of image generation model from QR code

The main feature of the proposed cost function is that the image L is generated by affine transforming f_W the QR code q to the image domain. Compared with the pixel-wise restoration, a large number of parameters can be reduced while maintaining expressiveness, thereby enabling robust restoration.

The point (u^q, v^q) on the QR code corresponding to the image coordinates (u, v) after affine transformation are expressed as follows:

$$\begin{pmatrix} u^q \\ v^q \end{pmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \end{bmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad (4)$$

The image L is represented by sampling the QR code q by bilinear interpolation as follows:

$$L_{uv} = \sum_j^r \sum_i^r q_{ij} \max(0, 1 - |u^q - i|) \max(0, 1 - |v^q - j|) \quad (5)$$

Here, q_{ij} is a luminance value of a QR code on 2D grid coordinates (i, j) . This sampling is differentiable, and partial derivatives of code q and affine parameters W for luminance L can be calculated [12].

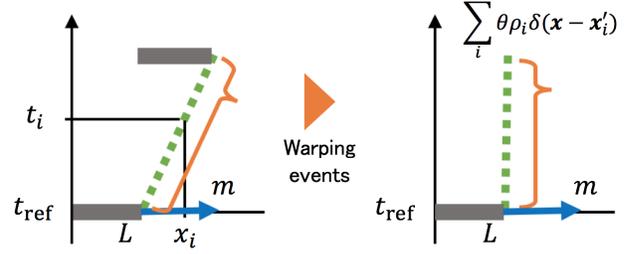


Figure 3: Accumulation of events. The events are warped with motion vector \mathbf{m} and summed in the time direction to obtain the time change of luminance.

3.5. Optimization scheme

The system flow is shown in Fig. 4. If the affine parameter W is not determined correctly, the code space can not correspond to the image space, making it difficult to estimate the QR code q . Thus, optimization was performed in two steps, first \mathbf{m} and W , then q . In the first step, for the fixed value of q , the part determined by the specifications of the QR code as in the lower left of Fig. 4 was used.

The initial value of W is estimated before optimization because the cost function is not convex when the positional correspondence between the QR code and image shifts by one code. In the QR code, the symbols shown in Fig. 5 are arranged at three corners. For luminance images, the position and orientation are determined by detecting where the code has a 1:1:3:1:1 ratio, shown on the left in Fig. 5. In the case of events, the distance between events is a 1:1:3:1:1 ratio as shown on the right of Fig. 5. The symbol was detected by finding the peak via convolving the one-dimensional kernel of this pattern while changing the scale. For the initial value of \mathbf{m} , a velocity vector is obtained by principle components analysis (PCA) in spacetime. After W converges in the first step, optimization is performed for q in the second step.

The convergence of the parameter \mathbf{p} was determined based on the update amount as follows:

$$\|\mathbf{p}^{(i)} - \mathbf{p}^{(i-1)}\| < \epsilon_p \quad (6)$$

The superscript i indicates the number of iterations, and ϵ_p indicates the criteria for convergence judgment.

After convergence, q is binarized with a threshold of 0 for obtaining the QR code. All optimizations were simply performed using the first-order gradient descent method.

4. Experiments

4.1. Dataset

Since there was no publicly available dataset that had QR codes scenes using an event-based camera, we constructed

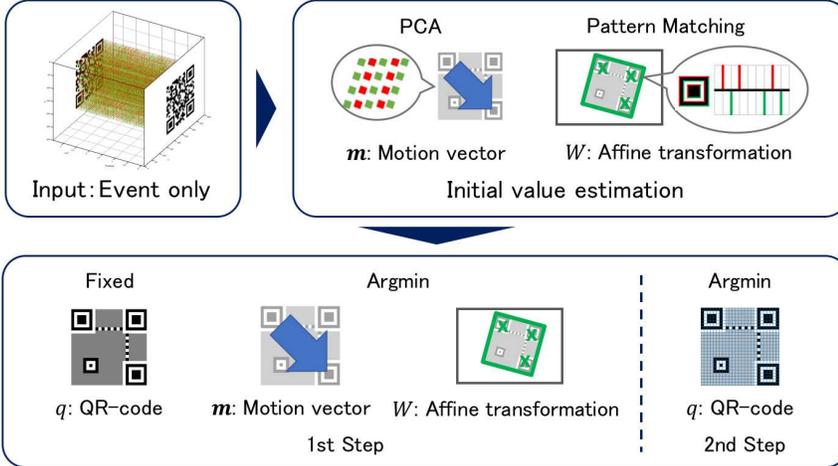


Figure 4: System flow. The input is only events over a duration of Δt . The initial values of motion vector \mathbf{m} and affine parameter W are calculated before optimization. In the first step, \mathbf{m} and W are optimized using known parts of the QR code, and after W converges, q is optimized in the second step.

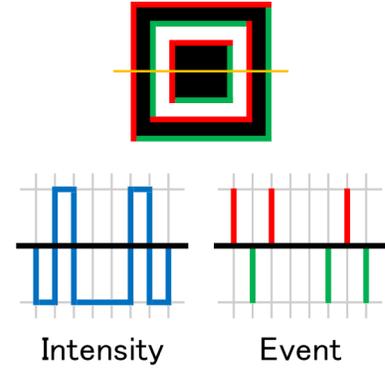


Figure 5: Symbol detection for affine transformation's initial value estimation. The affine transformation is calculated from the three symbols location detected by matching the patterns in the 1:1:3:1:1 ratio.

a new dataset called the QRECD. The dataset includes real data acquired by the Samsung DVS Gen3 [17] event camera and artificial data simulated by ESIM [13], with a resolution of 480×640 .

As shown in Fig. 6a, we created a situation where the plane of the QR code the optical axis of the camera are perpendicular and the QR code is moving at a constant linear motion. The QR code was attached at an angle of $\{0^\circ, 30^\circ, 45^\circ\}$ in the traveling direction. Two train toys with different speeds (faster train, approx. 2m/s; slower train, approx. 0.35m/s) were prepared to obtain real data. The 25×25 QR codes of two binary patterns encoding different character strings at the error correction level H (approx 30%) are used. The error correction rate is defined as the ratio of recoverable data codewords out of all codewords, including codewords for error correction.² In this case, it is possible to decode the character strings if approximately five codewords consisting of eight bits are mistaken. A dataset containing 18 scenes in all will be released when the paper is published.

4.2. Setup

The time duration Δt of the event sequence used in each estimation was 0.25 ms for faster train scenes and simulator scenes, while it was 0.5 ms for slower scenes. For utilizing events while the QR code appeared on the screen, we estimated N times (see Fig. 6b). The final QR-code was the mode for each bit of the QR code estimated N times. In the experiment, only one type of QR code binary pattern

²Error Correction Feature: https://www.qrcode.com/en/about/error_correction.html

was used because it has nothing to do with the algorithm. The threshold θ for event occurrence was set to 0.2, while the update rate was set to 0.001 for the first step, and 1 for the second step. Constants for convergence judgment are ϵ_W as 0.001 and ϵ_q as 5. MATLAB was used for the implementation and experiments were performed on a system consisting of an Intel $\text{\textcircled{R}}$ CORE TM i9 7900X CPU and system memory of 64 GB.

4.3. Comparison

The following comparisons of methods were carried out to prove the usefulness of our method.

Image For faster train scenes, the Lumenera industrial high-speed camera Lt225 (330 fps) was used, and for slower train scenes, the Logicool web camera C615 (30 fps) was used.

Bardow et al. [1] The regularization parameters used were those in their paper. The number of images was set to be optimized K to 128, and the time duration δt was the same as that in our method, denoted by Δt . In this method, the Nvidia QuadroP6000 GPU was used.

w/o QR The method, in which the code space was not introduced, and smoothness regularization was added as shown in the cost function below:

$$\operatorname{argmin}_{L, \mathbf{m}} \sum_{\mathbf{x}} \left(\left\| (\nabla L(\mathbf{x}) \cdot \mathbf{m}) \Delta t + \sum_i \theta \rho_i \delta(\mathbf{x} - \mathbf{x}'_i) \right\|^2 + \lambda \|\nabla L(\mathbf{x})\|^2 \right). \quad (7)$$

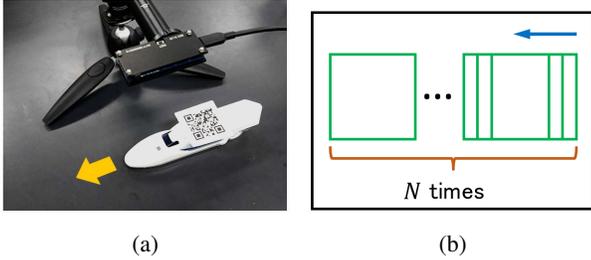


Figure 6: (a) Experimental setup. A toy train with a QR code attached is taken from above with an event camera. (b) QR code is estimated using the event sequence N times for one scene over a duration of Δt .

The optimization parameters are the intensity image L and motion vector m . The regularization parameter λ was set to 0.2.

Ours Our approach, in which the code space has been introduced.

4.4. Evaluation metrics

The five evaluation metrics were delineated below were used.

Mode acc. This is the bit accuracy of the QR code, which is finally determined as the mode when converging in N trials. The known part of the QR code is excluded when the accuracy is calculated.

Time/iteration to read Average time and iterations to read when the reading is successful, regardless of convergence.

Max acc. This is the max bit accuracy of the QR code, regardless of convergence, in N trials. The accuracy is calculated without the known part of the QR code. This metric can not be calculated when the true QR is unknown in real operation, but in the experiments, it is calculated for comparing the limits of the model.

Read rate Success rate of reading the QR code when convergence occurs.

ZXing read rate from the image Success rate of reading from estimated images using the most famous open source QR code library, ZXing.³

In the image-based reconstruction method, four indexes were calculated for the QR code detected using the pattern matching; this was similar to the initial value estimation in our method. This is because the judgment of detection is severe in ZXing and the detection rate is lowered; thus, comparison cannot be performed to the same standard as in our

³ZXing: <https://github.com/zxing/zxing>

method. As shown on the left side of the Fig. 5, the one-dimensional kernel that becomes 1:1:3:1:1 was convolved in the vertical and horizontal directions while changing the scale, and peaks were detected in either direction. After detection, the estimated QR code was obtained by binarizing with threshold 0 in the same way as in our method.

4.5. Qualitative result

The estimated images are shown in the Fig. 7. Since the luminance is not estimated in the form of an image in our method, the displayed image is generated by the estimated QR code and affine transformation.

It can be seen that the general camera images are blurred in any scene, but the event-based camera does not suffer from motion blurring. In our method, only the QR code can be estimated without restoring redundant luminance of the whole image. In the scene of real data from the faster and slower trains, image-based restoration makes the code boundaries unsharp due to a lot of noise. When the angle of the QR code is 0° , the event does not occur from the edge where the motion and luminance gradient are orthogonal. The intensity of the code parallel to the direction of movement is not restored with Bardow *et al.* [1] or w/o QR.

The estimated QR codes are shown in Fig. 8. The magenta areas represent codes that were mistakenly estimated to be black, whereas green areas indicate the opposite. In the scene where the object moves rapidly, since the occurrence of the event is delayed and the trajectory does not become straight, QR code reconstruction tends to be more difficult. In the faster train scene, it is clear that our method has a higher QR code accuracy by an overwhelming margin.

4.6. Quantitative result

The experimental results are shown in Table 1. In the faster train scene, the QR codes can not be read at all with the 330-fps industrial camera and the method of Bardow *et al.* [1]. While the read rate after convergence is less than 10% in w/o QR, our method achieves a read rate of 90% or more. Since the mode acc. is 99% and time to read is short at any angle with our method, fast and robust QR code estimation was realized even in the most demanding fast scenes.

In the slower train and simulator scenes, the noise is less and the events are ideally generated from an edge while making a plane in spacetime. Therefore, the QR code was estimated with high accuracy in the method of Bardow *et al.* using K times longer duration events. However, to estimate a dense image from sparse events with smoothness regularization, many iterations are required for reading.

The low reading rate in our method, at 0° is considered to be the result of early convergence judgment based on the same criteria as used for other angles. It is necessary to consider when to terminate optimization when it can not be

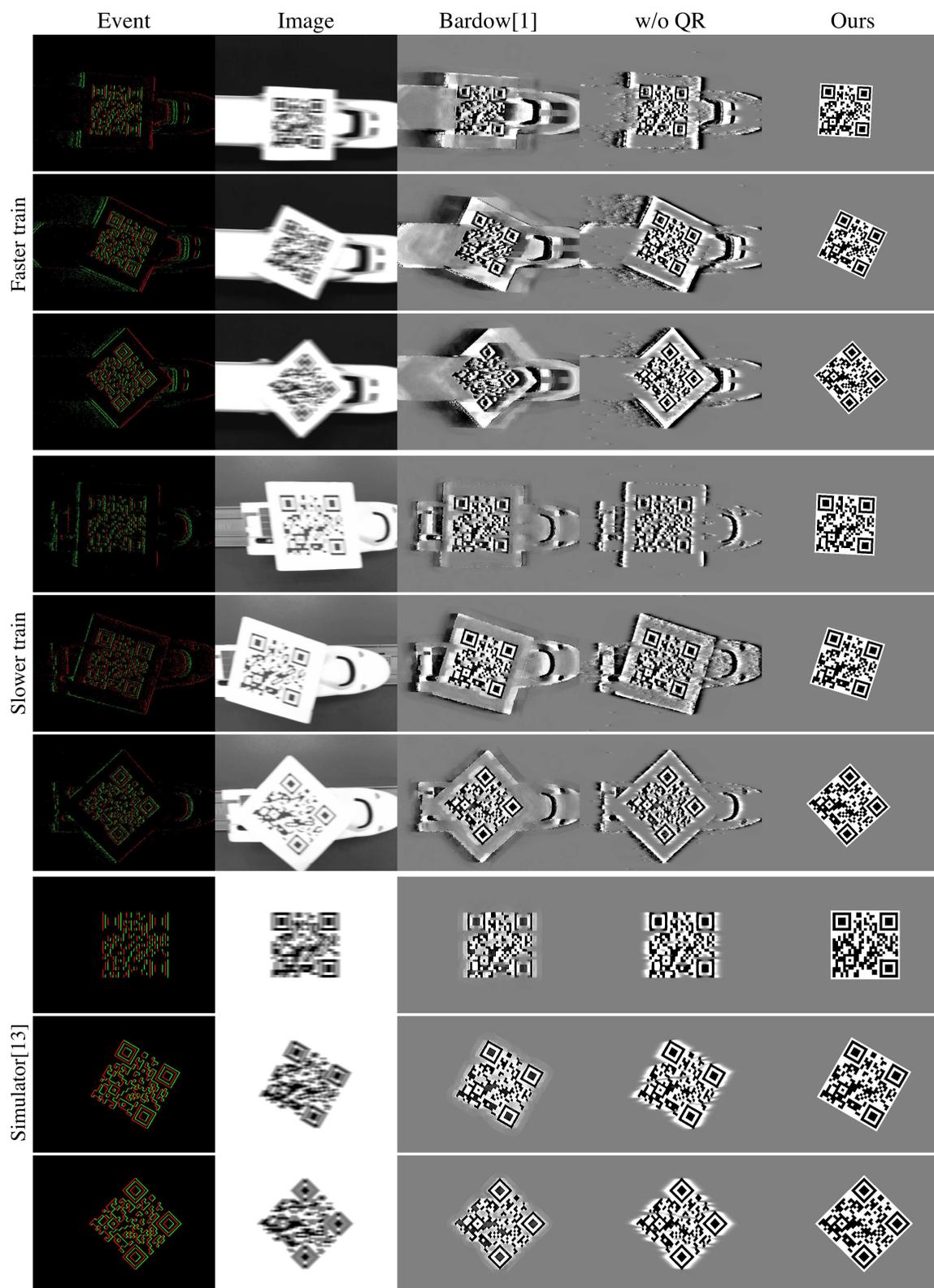


Figure 7: Estimated images. In the faster, slower train and simulator scenes, for every three columns from the upper row, the angles of the QR code are 0° , 30° , and 45° from the top.



Figure 8: Estimated QR codes. In the faster, slower train and simulator scenes, for every three columns from the upper row, the angles of the QR code are 0° , 30° , and 45° from the top. The magenta areas are codes that were mistakenly estimated to be black, whereas green areas indicate the opposite. These QR codes are determined by the mode from the QR code converging in each of N trials.

read in real operation, such as using different convergence criteria depending on the angle of detection, and using the number of iterations as the criterion.

In this method, max acc. was 100 in every scene. This shows that a model in which an image is generated by affine transforming a QR code is consistent with the data and acts as an effective regularization.

The QR code reading rate using ZXing for the entire estimated image is low. This indicates that the image restored from the event data through smoothness regularization in the image space is low in quality, as seen in features like its unclear edges.

5. Conclusion

We worked on the task of reading a QR code using an event camera in high-speed scenes where blurs occur even with the use of an industrial high-speed camera. In the conventional method, the intensity was restored from the event data in the whole image from the event data under the smoothness assumption. In contrast, we proposed the QR code reconstruction method of optimization with the code space more constrained than the image space. By modeling such that the image was generated by affine transforming the QR code, we succeeded in significantly reducing the number of parameters while maintaining the representation of the scene. We demonstrated that the QR code can be read quickly and robustly using our dataset, including real and simulated data.

Assuming a practical application such as a belt conveyor, the geometric transformation is limited to the affine approach, and the motion vector is limited to two parameters common to all pixels. When the QR code plane is tilted in relation to the camera in a hand-held reader, some extensions are required. Geometric transformations can easily be extended to the projective transformation, but it is necessary to reconsider motion vector definitions and initial value estimation methods. However, the basic model in which the image is generated from the QR code remains unchanged, and it still seems to work powerfully.

References

- [1] P. Bardow, A. J. Davison, and S. Leutenegger. Simultaneous Optical Flow and Intensity Estimation from an Event Camera. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 884–892, 2016. [2](#), [4](#), [5](#), [8](#)
- [2] S. Barua, Y. Miyatani, and A. Veeraraghavan. Direct face detection and video reconstruction from event cameras. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016. [2](#)
- [3] C. Brandli, R. Berner, M. Yang, S. C. Liu, and T. Delbruck. A 240×180 130 dB 3 μ s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. [1](#)

| Scene | Method | Deg. | N | Mode acc. [%] | Time [s] | Iteration to read | Max acc. [%] | Read rate [%] | ZXing read rate[%] |
|----------------|--------------------------|------|-----|---------------|-------------|-------------------|--------------|---------------|--------------------|
| Faster train | Image | 0° | 12 | - | - | - | - | - | 0 |
| | | 30° | | - | - | - | - | - | 0 |
| | | 45° | | - | - | - | - | - | 0 |
| | Bardow <i>et al.</i> [1] | 0° | 22 | 50.0 | - | - | 88.7 | 0 | 0 |
| | | 30° | | 85.9 | - | - | 91.2 | 0 | 0 |
| | | 45° | | 50.0 | - | - | 82.6 | 0 | 0 |
| | w/o QR | 0° | 150 | 50.0 | 1.30 | 63.9 | 99.0 | 8.77 | 0 |
| | | 30° | | 65.6 | 2.15 | 105 | 98.5 | 4.00 | 2.00 |
| | | 45° | | 61.0 | 3.50 | 168 | 95.3 | 1.30 | 2.00 |
| | Ours | 0° | 150 | 99.2 | 0.75 | 12.6 | 100 | 92.7 | - |
| | | 30° | | 99.5 | 0.34 | 6.04 | 100 | 96.7 | - |
| | | 45° | | 99.7 | 0.20 | 4.16 | 100 | 99.2 | - |
| Slower train | Image | 0° | 4 | 66.4 | - | - | 67.9 | 0 | 0 |
| | | 30° | | 67.9 | - | - | 69.0 | 0 | 0 |
| | | 45° | | 63.8 | - | - | 64.3 | 0 | 0 |
| | Bardow <i>et al.</i> [1] | 0° | 172 | 99.5 | 40.4 | 65.2 | 100 | 100 | 16.9 |
| | | 30° | | 99.2 | 10.7 | 17.2 | 100 | 100 | 11.6 |
| | | 45° | | 99.0 | 21.1 | 33.5 | 100 | 100 | 1.16 |
| | w/o QR | 0° | 300 | 98.2 | 1.00 | 42.4 | 100 | 56.9 | 0.67 |
| | | 30° | | 99.7 | 0.78 | 28.7 | 99.7 | 91.20 | 55.3 |
| | | 45° | | 100 | 1.07 | 44.0 | 99.0 | 72.60 | 1.67 |
| | Ours | 0° | 300 | 97.9 | 1.38 | 19.6 | 100 | 73.1 | - |
| | | 30° | | 99.7 | 0.60 | 9.05 | 100 | 88.0 | - |
| | | 45° | | 100 | 0.33 | 5.51 | 100 | 96.9 | - |
| Simulator [13] | Image | 0° | 5 | 76.2 | - | - | 76.2 | 0 | 0 |
| | | 30° | | 90.0 | - | - | 90.0 | 0 | 0 |
| | | 45° | | 92.6 | - | - | 92.6 | 0 | 0 |
| | Bardow <i>et al.</i> [1] | 0° | 272 | 52.3 | 127 | 195 | 99.7 | 63.6 | 0 |
| | | 30° | | 100 | 6.11 | 9.17 | 100 | 100 | 100 |
| | | 45° | | 100 | 9.51 | 14.0 | 100 | 100 | 77.6 |
| | w/o QR | 0° | 400 | 97.9 | 0.51 | 19.6 | 99.0 | 99.0 | 0 |
| | | 30° | | 100 | 0.63 | 25.5 | 100 | 95.8 | 77.5 |
| | | 45° | | 100 | 0.60 | 19.7 | 100 | 100 | 3.25 |
| | Ours | 0° | 400 | 87.7 | 0.78 | 10.6 | 100 | 48.2 | - |
| | | 30° | | 100 | 0.15 | 3.18 | 100 | 100 | - |
| | | 45° | | 100 | 0.14 | 3.00 | 100 | 99.1 | - |

Table 1: Results.

- [4] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza. Event-based Vision: A Survey. *arXiv:1904.08405*, 2019. 1
- [5] G. Gallego, M. Gehrig, and D. Scaramuzza. Focus Is All You Need: Loss Functions For Event-based Vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12280–12289, 2019. 3
- [6] G. Gallego, H. Rebecq, and D. Scaramuzza. A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation. 2018. 3
- [7] S. M. M. I, L. Wang, Y.-s. Ho, and K.-j. Yoon. Event-based High Dynamic Range Image and Very High Frame Rate Video Generation using Conditional Generative Adversarial Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10081–10090, 2019. 2
- [8] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. Davison. Simultaneous Mosaicing and Tracking with an Event Camera. *Proceedings of the British Machine Vision Conference (BMVC)*, 2014. 2
- [9] H. Kim, S. Leutenegger, and A. J. Davison. Real-time 3D re-

- construction and 6-DoF tracking with an event camera. *European Conference on Computer Vision (ECCV)*, pages 349–364, 2016. [2](#)
- [10] P. Lichtsteiner, C. Posch, T. Delbruck, and S. Member. A 128×128 120 dB 15 μ s Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. [1](#)
- [11] L. Pan, R. Hartley, C. Scheerlinck, M. Liu, X. Yu, and Y. Dai. Bringing Blurry Alive at High Frame-Rate with an Event Camera. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6820–6829, 2019. [2](#)
- [12] Q. K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. *NIPS*, 2015. [1](#), [3](#)
- [13] H. Rebecq, D. Gehrig, and D. Scaramuzza. ESIM: an Open Event Camera Simulator. *Conf. on Robotics Learning (CoRL)*, 2018. [4](#), [8](#)
- [14] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. Events-to-Video: Bringing Modern Computer Vision to Event Cameras. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3857–3866, 2019. [2](#)
- [15] C. Scheerlinck, N. Barnes, and R. Mahony. Continuous-time Intensity Estimation Using Event Cameras. *Asian Conference on Computer Vision (ACCV)*, pages 308–324, 2018. [2](#)
- [16] P. A. Shedligeri, K. Shah, D. Kumar, and K. Mitra. Photorealistic Image Reconstruction from Hybrid Intensity and Event based Sensor. *arXiv:1805.06140*, 2018. [2](#)
- [17] B. Son, Y. Suh, S. Kim, H. Jung, J. S. Kim, C. Shin, K. Park, K. Lee, J. Park, J. Woo, Y. Roh, H. Lee, Y. Wang, I. Ovsianikov, and H. Ryu. A 640×480 dynamic vision sensor with a 9m pixel and 300Meps address-event representation. *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 66–67, 2017. [4](#)