sensors

MDPI

Article

Accuracy and Speed Improvement of Event Camera Motion Estimation Using a Bird's-Eye View Transformation

Takehiro Ozawa [1,*], Yusuke Sekikawa [2] and Hideo Saito [1]

[1] Graduate School of Science and Technology, Keio University, Yokohama 223-8522, Japan; hs@keio.jp
[2] Denso IT Laboratory, Inc., Tokyo 150-0002, Japan; ysekikawa@mail.d-itlab.co.jp
[*] Correspondence: takehiro.o@keio.jp

Abstract: Event cameras are bio-inspired sensors that have a high dynamic range and temporal resolution. This property enables motion estimation from textures with repeating patterns, which is difficult to achieve with RGB cameras. Therefore, motion estimation of an event camera is expected to be applied to vehicle position estimation. An existing method, called contrast maximization, is one of the methods that can be used for event camera motion estimation by capturing road surfaces. However, contrast maximization tends to fall into a local solution when estimating three-dimensional motion, which makes correct estimation difficult. To solve this problem, we propose a method for motion estimation by optimizing contrast in the bird's-eye view space. Instead of performing three-dimensional motion estimation, we reduced the dimensionality to two-dimensional motion estimation by transforming the event data to a bird's-eye view using homography calculated from the event camera position. This transformation mitigates the problem of the loss function becoming non-convex, which occurs in conventional methods. As a quantitative experiment, we created event data by using a car simulator and evaluated our motion estimation method, showing an improvement in accuracy and speed. In addition, we conducted estimation from real event data and evaluated the results qualitatively, showing an improvement in accuracy.

Keywords: motion estimation; event-based camera; bird's-eye view; autonomous vehicles

check for updates

Citation: Ozawa, T.; Sekikawa, Y.; Saito, H. Accuracy and Speed Improvement of Event Camera Motion Estimation Using a Bird's-Eye View Transformation. *Sensors* **2022**, *22*, 773. https://doi.org/10.3390/s22030773

Academic Editor: Omar Ait Aider

Received: 14 December 2021
Accepted: 17 January 2022
Published: 20 January 2022

1. Introduction

1.1. Backgrounds

Camera motion estimation is one of the most important technologies in many applications, such as automatic driving and assistive technologies. In particular, motion estimation by pointing a camera at a road surface has an advantage in that objects that can affect it, such as automobiles and pedestrians, are usually not easily visible [1–6]. Teshima et al. [1] proposed a method to estimate the position and orientation of a vehicle from sequential images of the ground using the known homography obtained from a normal camera. Saurer et al. [2] proposed a method to find different minimal solutions for egomotion estimation of a camera based on homography knowing the gravity vector between images. Gilles et al. [3] proposed a method based on unsupervised deep learning for motion estimation with a downward-looking camera. However, in spite of active research, camera motion estimation by using the road surface has the following problems: it is difficult to extract and match feature points robustly from noisy ground textures; the high-speed movement of cameras causes motion blur; it often involves challenging illumination conditions.

Event cameras [7] have been developed with features that can address these problems. Instead of recording an image intensity at a synchronous frame rate, event cameras record asynchronous positive and negative spikes, called events, in response to changes in the brightness of a scene. Therefore, the output is not an intensity image, but a stream of asynchronous events, where each event consists of its space–time coordinates. The advantages of event cameras include their high temporal resolution (µs order), high dynamic range

(130 dB), and low power consumption [8,9], as well as the ability to acquire data without blurring, even when the event camera itself is moving at high speed. However, they are fundamentally different in their operating characteristics from traditional frame-based cameras; thus, they require completely new algorithms.

As one of the methods for motion estimation using event cameras, Gallego et al. [10,11] and Stoffregen et al. [12] showed that the spatio-temporal trajectory of an event when capturing a plane can be modeled by a homographic transformation, and the motion can be determined by maximizing the contrast of an image of a warped event (IWE) along the trajectory (i.e., contrast maximization).

1.2. Motivations

As shown by Nunes et al. [13], contrast maximization has the problem that the loss function may have multiple extrema. This implies that the estimation results are highly dependent on the initial values. In addition, especially when all the events are warped on the same point or line, the contrast is larger, and the entropy is smaller (as compared with when they are warped with the correct value); see Figure 1. Therefore, depending on the initial values, it is possible to estimate parameters that differ significantly from the correct values. This problem can occur when estimating motion in three-dimensional Euclidean space, such as in homography. Hence, it is necessary to solve this problem when performing camera motion estimation using an event camera that is diagonally facing the ground.

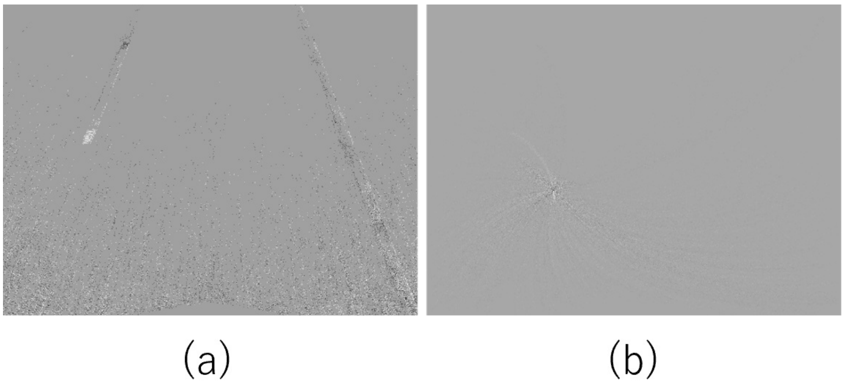

Figure 1. An example of failed estimation. (**a**) Events before estimation. (**b**) Events warped by the wrong estimation. The events have been warped onto the same point.

1.3. Objectives

Our research objectives were to solve the problem of event camera motion estimation using contrast maximization, where the loss function has multiple extrema, and to show its effectiveness experimentally. To solve this problem, we propose a method for motion estimation by maximizing the contrast in the bird's-eye view space. Figure 2 shows the overview of our method. Assuming that the event camera is mounted on a vehicle and its mounting position is known, homographic information can be obtained in advance. Furthermore, assuming that the vehicle moves parallel to the road, the motion of the event camera can be modeled in 3DoF. Using the obtained homography, a bird's-eye view transformation can reduce the 3D motion estimation problem to the 2D planar motion estimation problem. We empirically found that this transformation makes the loss function convex around the true value and improves the accuracy and speed of the estimation.

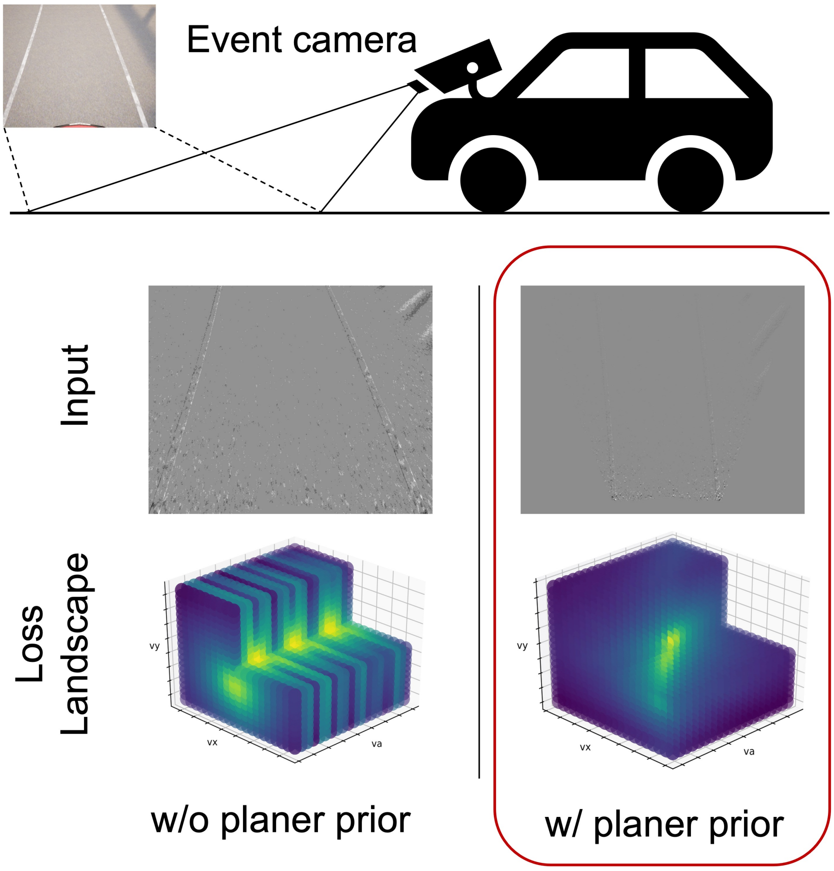

Figure 2. Overview of our method. When capturing a road surface at an angle for motion estimation, it can be difficult to track using an RGB camera. Even with an event camera, homographic motion estimation can have multiple extrema in the loss function. By performing a bird's-eye view transformation, the loss function becomes convex around the true value, which allows for improvements in accuracy and speed.

The proposed method was quantitatively evaluated using event sequences generated using the CARLA simulator [14]. We compared the accuracy and speed of the motion estimation results of the proposed method and the existing methods and show that the proposed method was better in both cases. In the above experiment, we also confirmed that the loss function was convex around the true value by performing a bird's-eye view transformation. We also qualitatively evaluated the proposed method by estimating the motion of an actual event camera installed on a dolly.

1.4. Related Works

As it is a similar problem to camera motion estimation by using the road surface, optical flow estimation with event cameras can be mentioned. In early research, a method using intensity image reconstruction by an accumulation of events was proposed [15]. Furthermore, several methods specific to event cameras using local plane fitting in space–time were proposed [16–20]. Almatrafi et al. [21] introduced the concept of the distance surface and used it for a successful estimation of optical flow with high accuracy. Recently, methods for optical flow estimation using machine learning have been successfully applied [22–27].

Motion estimation using event cameras has also been studied. Weikersdorfer et al. [28,29] captured images of a ceiling using an event camera mounted upward and used particle filters to estimate the position of the camera, which was moving parallel to the plane. However, this method requires capturing a plane in parallel and a black-and-white line pattern in the scene. Kim et al. [30,31] proposed a method to jointly estimate six degrees of freedom (DoFs) of the camera, depth, and intensity gradient of the scene. The results were limited to small-scale environments and motions. Gallego et al. [32] proposed to estimate rotational motion by accumulating events over a small time window into an image and maximizing the variance of the image. Gallego et al. [10,11] proposed a contrast maximization framework, based on the method of [32]. This framework maximizes the contrast of the image by warping events according to the model parameters and accumulating the warped events onto the IWE. Nunes et al. [13] proposed a method called entropy minimization, based on contrast maximization, that performs motion estimation without projecting onto the IWE. However, as shown by Nunes et al. [13], both contrast maximization and entropy minimization have the problem that the loss function may have multiple extrema.

In response to this problem, Liu et al. [33] proposed a method for finding the global maximum solution of a rotational motion using a branch and bound (BnB) approach for contrast maximization. Peng et al. [34] also proposed a method for finding the global maximum solution using BnB and achieved planar motion estimation from an event camera mounted downward on an automatic guided vehicle (AGV). However, when mounting an event camera on a car, it is difficult to capture the road surface vertically.

1.5. Contributions

The main contributions of this paper are as follows:

- We empirically found that the loss function of contrast maximization becomes convex around the true value by performing a bird's-eye transformation to the event data. In doing this, we accomplished the following:
  - The initial value of the estimation can be taken widely;
  - The global extremum will be the correct value.
- The effectiveness of the proposed method was demonstrated using both synthetic and real data.

Note that the proposed method is based on empirical results and lacks mathematical analysis. However, the effectiveness of our method was clearly demonstrated through experiments.

The rest of this paper is organized as follows: Section 2 describes our proposed method. In Section 3, we discuss the results of quantitative experiments conducted with event data generated by the CARLA simulator and qualitative experiments conducted with real event data. Section 4 presents a discussion of the results, the limitations, and the future works. Finally, Section 5 summarizes the overall results of this study.

2. Materials and Methods

2.1. Event Data Representation

An event camera asynchronously outputs the luminance change of each pixel, called an event. Suppose there exists a given set of events $E = e_k$ in the time window $[0, T]$, where $e_k = (\mathbf{x}_k, t_k, p_k)$. Here, $\mathbf{x}_k$ is the coordinate where the event triggered, t_k, is the timestamp where the event occurs with a precision of microseconds and $p_k \in \{-1, +1\}$ is the polarity of the event, which indicates brighter or darker changes when the logarithmic intensity changes above a threshold at a certain pixel $\mathbf{x} = (x, y)$.

2.2. Contrast Maximization

Gallego et al. [10] proposed a unified framework called contrast maximization, which can be applied to various problems in event-based vision. Because the proposed method relies on contrast maximization, it is briefly reviewed here.

Events originate from edges with high luminance gradients, as indicated by their properties. The goal of contrast maximization is finding the trajectory of events that originate from the same edge in XYT space caused by the camera or scene motion. Given their coordinates and timestamps, events originating from the same edge can be aggregated to a single point by warping along the trajectory. The warped events projected onto the XY image plane are called the image of warped events (IWEs). Once the correct trajectory is found, the IWE becomes a sharp image as an edge map. Gallego et al. [10] estimated the trajectory so that the contrast of the IWE is maximized.

Given N events, $\mathcal{E} = \{e_k\}_{k=1}^{N}$, let $\mathbf{x}'_k = \mathbf{W}(\mathbf{x}'_k, t_k; \theta)$ be the function that warps e_k along the trajectory to $\mathbf{x}'_k$. θ is the parameter of the warping, and the IWE is obtained by accumulating the events warped by $\mathbf{W}$ as follows:

$$I(\mathbf{x};\boldsymbol{\theta}) \doteq \sum_{k=1}^{N} \delta\big(\mathbf{x} - \mathbf{W}(\mathbf{x}'_k, t_k; \theta)\big), \tag{1}$$

where $\delta(\cdot)$ is a delta function, approximated by a Gaussian. Now, the variance of the IWE is defined as follows:

$$\mathrm{Var}(I;\theta) = \frac{1}{N}\sum_{\mathbf{x}} (I(\mathbf{x};\theta) - \mu(I;\theta))^2 \tag{2}$$

$$\mu(I;\theta) = \frac{1}{N}\sum_{\mathbf{x}} I(\mathbf{x};\theta). \tag{3}$$

The goal of contrast maximization is to find the θ^* that maximizes the variance of the IWE,

$$\theta^* = \arg\max_{\theta} \mathrm{Var}(I;\theta). \tag{4}$$

Even surfaces with few robust features, such as many roads, can be tracked using the contrast maximization described above.

Camera motion estimation in planar scenes such as a road surface can be achieved by using the homography model. This model has eight degrees of freedom (DoFs) and can be parameterized by the 3D angular velocity, $\mathbf{w} = (w_x, w_y, w_z)$; the 3D velocity, $\mathbf{v} = (v_x, v_y, v_z)$; and the normalized normal vector of the target plane, $\mathbf{n} = (n_x, n_y, n_z)$, expressed as:

$$\mathbf{W}(\mathbf{x}_k, t_k; \boldsymbol{\theta}) \propto \mathrm{H}^{-1}(t_k; \boldsymbol{\theta}) \begin{pmatrix} \mathbf{x}_k \\ 1 \end{pmatrix}, \tag{5}$$

where $\boldsymbol{\theta} = \left(\mathbf{w}^\top, \mathbf{v}^\top, \mathbf{n}^\top\right)^\top$ are the model parameters and H is the homography matrix.

2.3. Bird's-Eye View Transformation

Figure 3 shows the flow of our method. Generally, a vehicle moves parallel to a flat road surface without changing its height. Therefore, the motion of the onboard event camera is limited to a plane parallel to the road surface because the camera also moves parallel to the road surface. Therefore, we propose a method to reduce the number of estimation parameters by performing a bird's-eye view transformation on the event data to convert them to the viewpoint of a virtual camera pointed perpendicular to the road surface, resulting in a two-dimensional problem of motion estimation.

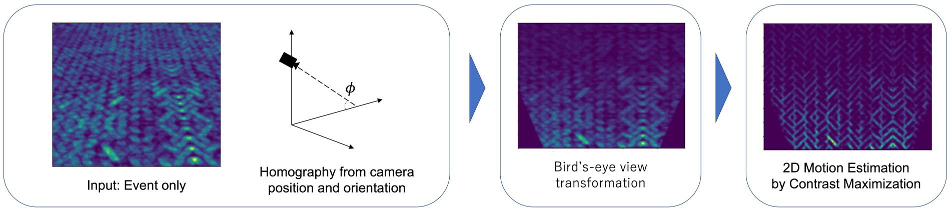
Input: Event only
ϕ
Homography from camera position and orientation
Bird's-eye view transformation
2D Motion Estimation by Contrast Maximization

Figure 3. System flow. The input is only events. The events are transformed into a bird's-eye view using the homography computed from the camera position and angle. The 2D motion estimation is then performed using contrast maximization.

A bird's-eye view conversion of images from conventional cameras (with the concept of frames) requires interpolation processing. However, since the pixels of an event camera react individually to changes in intensity, the event data are sparse, unlike a conventional image. Therefore, it can be converted more efficiently.

Suppose that the installation angle of the camera is known when the event camera is mounted on the vehicle; the homography for the bird's-eye view transformation can then be calculated as follows:

$$\mathrm{H}_{ab} = \mathrm{K}_b \cdot (\mathrm{R}_{ab} - \mathbf{t}_{ab}\mathbf{n}^T) \cdot \mathrm{K}_a^{-1}, \tag{6}$$

where a is the event camera and b is a virtual camera oriented perpendicular to the road surface. R_{ab} and $\mathbf{t}_{ab}$ are the rotation matrix and translation vector between a and b, respectively. $\mathbf{n}$ is the normal vector to the road surface; K_a and K_b are the internal parameters of each camera. Let ϕ be the angle between the normal of the road surface and the optical axis of the event camera, which is obtained as $\mathbf{n} = [0, -\sin\phi, -\cos\phi]^\top$.

With this homography, H_{ab}, we can transform the event as follows:

$$\hat{\mathbf{x}}_a = \mathrm{H}_{ab}\mathbf{x}_b. \tag{7}$$

The events transformed by (7) are converted to the viewpoint of a virtual camera moving parallel to the road surface. With this transformation, the motion estimation of the event camera is reduced to a two-dimensional translation and rotation problem. This model has three DoFs and can be parameterized by the angular velocity, ω, and the 2D velocity, $\mathbf{v} = (v_x, v_y)$, expressed as:

$$\mathbf{W}(\mathbf{x}_k, t_k; \boldsymbol{\theta}) = \begin{pmatrix} \mathrm{R}^{-1} & \mathbf{t} \\ \mathbf{0}^\mathbf{T} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ 1 \end{pmatrix}, \tag{8}$$

where $\boldsymbol{\theta} = (\omega, v_x, v_y)$ are the model parameters shared by all the pixels , $\mathrm{R} = \exp(\Delta t_k \omega)$ is the 2×2 rotation matrix where exp is the exponential map, and $\mathbf{t} = \Delta t_k \mathbf{v}$ is the translation 2D vector.

3. Results

The novelty of this method is that we empirically found that the bird's-eye view transformation makes the loss function convex around the true value, which leads to fast and accurate motion estimation. To demonstrate this, we performed the following three experiments using both synthetic and real datasets.

In the first experiment, to demonstrate the improvement in accuracy and speed of our method, we generated event data acquired from a car driving on a road using the CARLA simulator [14] and compared the accuracy and speed of the proposed method and existing method. In the second experiment, we compared the loss landscape near the ground truth using the same data as in the first experiment and showed that the non-convex loss function becomes convex by performing a bird's-eye view transformation. In the third experiment, to demonstrate the applicability of our method, we applied our method to real data captured by an event camera and evaluated the results. All experiments were conducted on a system consisting of an Intel Core i7-6800K CPU and 64 GB of system memory, using Python for the implementation.

3.1. Accuracy and Speed Evaluation

In this section, we describe and report on the evaluation of the accuracy and speed improvement provided by the bird's-eye view transformation of the event data. In order to compare the proposed method with bird's-eye view transformation and the existing method without bird's-eye view transformation, we conducted an experiment to estimate the parameters on the data generated by CARLA, which simulates a vehicle-mounted event camera (see Figure 4). The event camera was placed at a height of 1 m from the ground at an angle of 45°, and the event generation algorithm implemented in the CARLA simulator was used to create the data (similar to ESIM [35]).

For parameter estimation, we used the entropy minimization method [13] as a comparison, whose source code is publicly available. This method defines entropy instead of contrast and minimizes it to estimate the parameters. For the created data, we performed homographic motion estimation using the entropy minimization method and isometric motion estimation using the bird's-eye view transformation and then compared the speeds and accuracies of the estimations. The motion to be estimated is the angular velocity (w) and velocities (v_x, v_y) of the event camera.

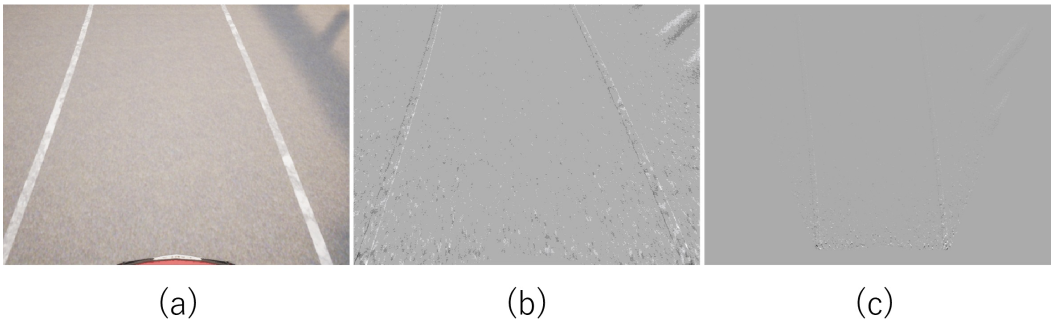

Figure 4. Road images taken at an angle as created by CARLA. (**a**) RGB image (not used). (**b**) Accumulated events. (**c**) Bird's-eye view transformation of events.

For the accuracy comparison experiments, we prepared three types of initial values for estimation: (a) using the previous estimation result, (b) using zero, and (c) using the previous estimation result, but with the exception of using zero if the distance from the true value was greater than the threshold.

The results of the accuracy comparison experiment are shown in Figure 5. As can be seen from the figure, the homographic motion estimation in Condition (a) estimated a value that was far from the ground truth. Furthermore, because the previous estimation result was used as the initial value of the next estimation, the estimates continued on incorrectly. Figure 6 shows the events after the warp in the XYT space, and we can see that there was a problem of events being warped to a straight line, as described in Section 1. This problem occurs because when events are warped on a straight line, the entropy is smaller than when it is warped on a correct value. This problem can occur when estimating velocities in three dimensions. In the proposed method, this problem is avoided by reducing the problem to a 2D velocity estimation by performing the bird's-eye view transformation; see Figure 7. In Condition (b), each time we used the initial value of zero, which was close to the true value, the aforementioned problem was avoided. However, it can be seen that even in this case, the proposed method successfully improved the accuracy. This can be attributed to the fact that the loss function had multiple extrema around the true value and thus fell into a local solution when performing homographic motion estimation, as shown in the following experiment. In the proposed method, the bird's-eye view transformation is used to make the loss function convex around the true value, which enables relatively accurate estimation. In Condition (c), it was shown that both methods could estimate relatively accurately by using the ground truth information. This means that if the initial values can be set appropriately, homographic motion estimation can be relatively accurate, but it is difficult to set the initial values properly in practice.

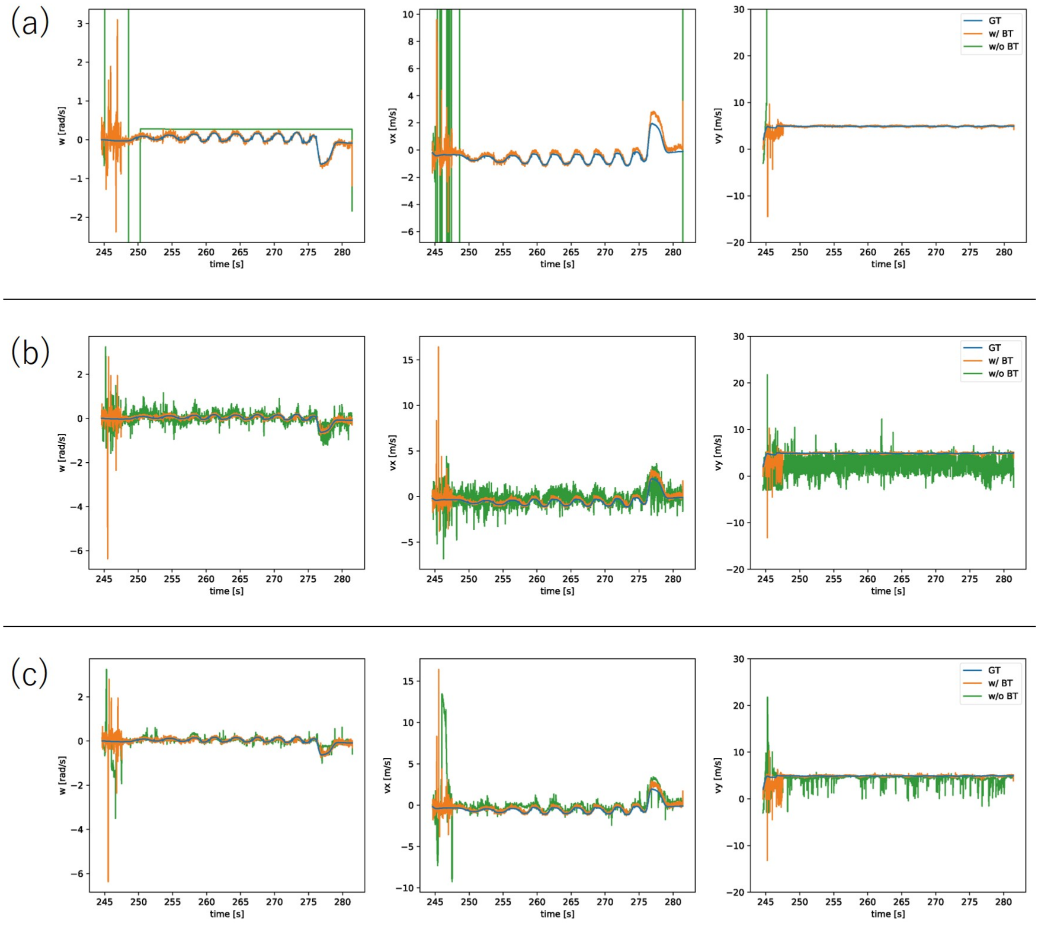

Figure 5. Results of the quantitative experiments. Blue: ground truth. Orange: proposed method with the bird's-eye view transformation (w/BT). Green: entropy minimization method without the bird's-eye view transformation (w/o BT). (**a**) Condition (a). (**b**) Condition (b). (**c**) Condition (c).

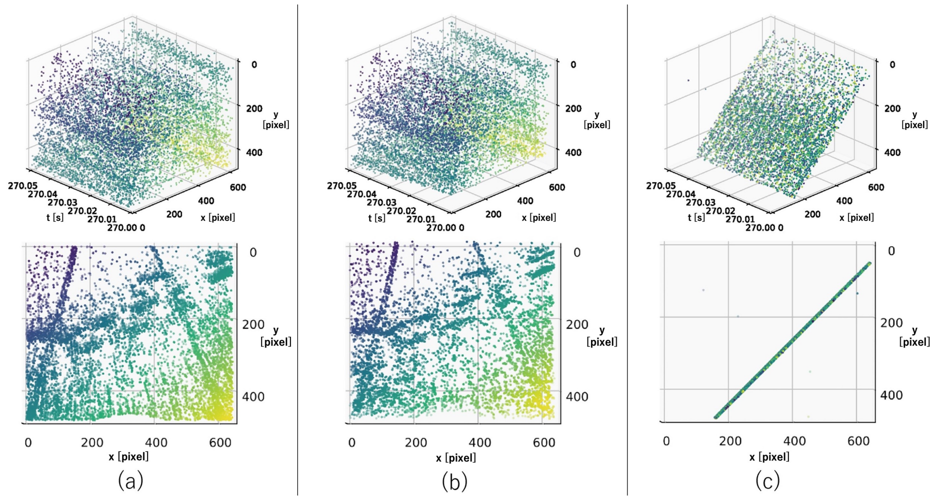

Figure 6. (**a**) Events before warp. (**b**) Events warped by the true value. (**c**) Events warped by a failed homographic motion estimation. The top row shows how the events were warped by homographic motion estimation in the XYT space. The bottom row shows how they are projected to the IWE.

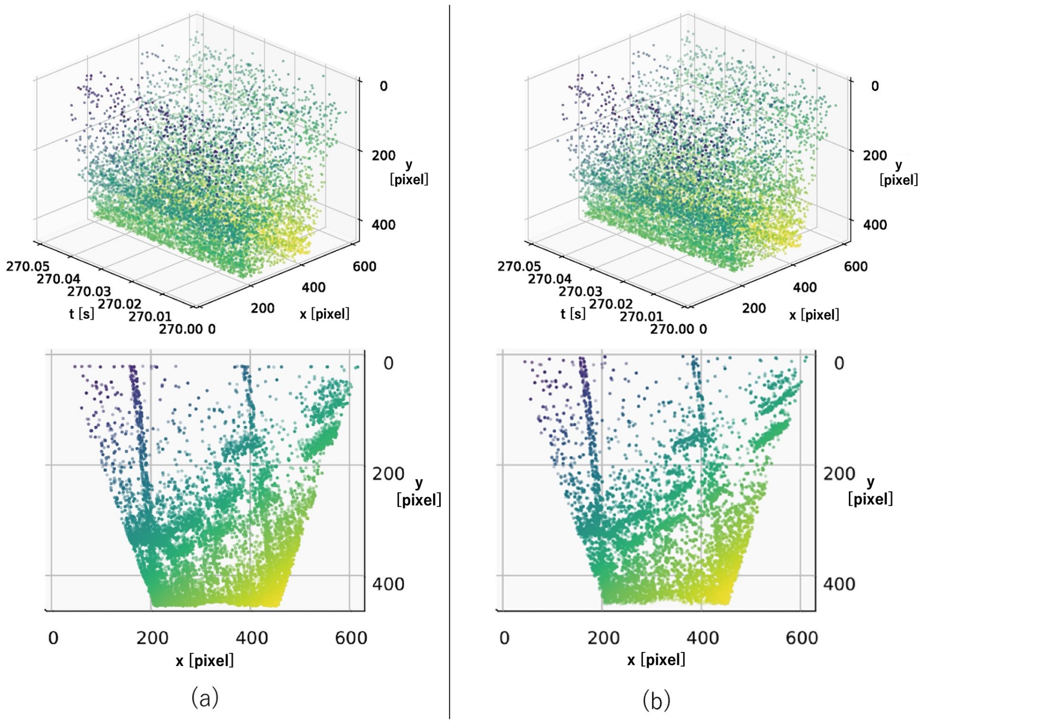

Figure 7. (**a**) Events before the warp. (**b**) Events warped by the isometric motion estimation. The top row shows how the events were warped by the isometric motion estimation in the XYT space. The bottom row shows how they were projected to the IWE.

For the computational complexity comparison experiments, we extracted four scenes from our dataset (see Figure 8) and measured the number of iterations and computation time until the estimation was completed.

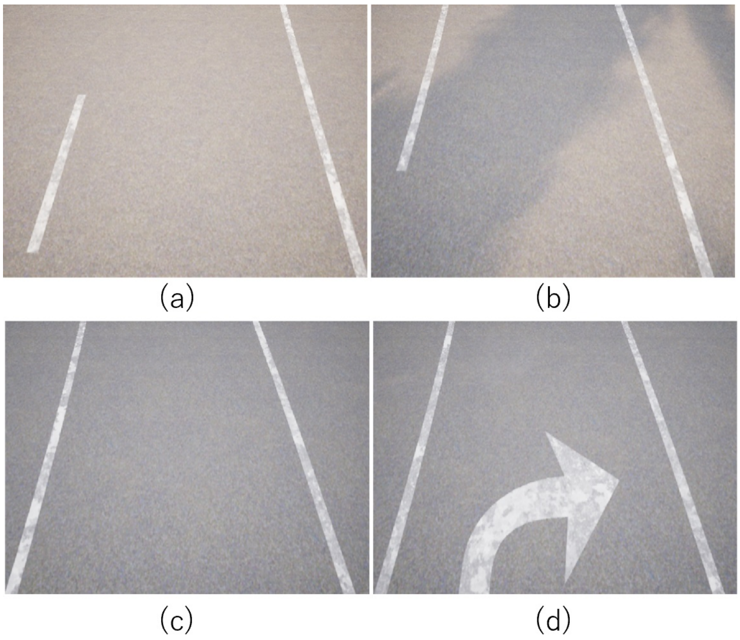

Figure 8. RGB images of scenes for the computational complexity experiments. (**a**) Scene 1. (**b**) Scene 2. (**c**) Scene 3. (**d**) Scene 4.

The results of the computational complexity comparison experiment are shown in Table 1. As can be seen from the table, the bird's-eye view transformation succeeded in reducing the computational complexity, both in terms of the number of iterations and the time required for each iteration. This indicates that reducing the estimated parameters by the bird's-eye view transformation not only reduced the computational complexity of each iteration, but also improved the ease of convergence.

Table 1. Iteration and total time for the motion estimation. The column "w/o BT" represents the result of homographic motion estimation without bird's-eye view transformation, and the column "ours" represents the estimation result by the proposed method. The top performance is highlighted in bold.

Metric	**Scenes**	**w/o BT**	**Ours**
Iteration	Scene 1	200	**19**
	Scene 2	124	**19**
	Scene 3	118	**11**
	Scene 4	107	**29**
	Ave.	137.2	**19.5**
Total time (s)	Scene 1	3.98	**0.57**
	Scene 2	12.39	**3.98**
	Scene 3	15.94	**3.41**
	Scene 4	21.63	**8.61**
	Ave.	13.49	**4.14**

3.2. Loss Landscape

In this section, we show experimentally that the bird's-eye view transformation makes the loss function in the motion estimation convex around the true value. We extracted 10,000 events at $t = 270$ s from the data used in Section 3.1 and compared the distribution of the entropy when warping by homography and when warping by isometry with a bird's-eye view transformation. To compare the loss landscape, we computed the entropy around the ground truth and visualized using Matplotlib libraries [36].

Figure 9 shows the distribution of the entropy near the true value with and without the bird's-eye view transformation. In the case of homographic motion estimation, we can see that several local extrema appeared near the true value. In addition, over a wider range, local extrema with lower entropy than the true value appeared. In contrast, in the case of the isometric motion estimation with a bird's-eye view transformation, the distribution of the entropy was convex over a somewhat wide range. Therefore, it can be said that the bird's-eye view transformation helped with the relatively fast convergence of the correct solution value, while preventing falling into a local solution.

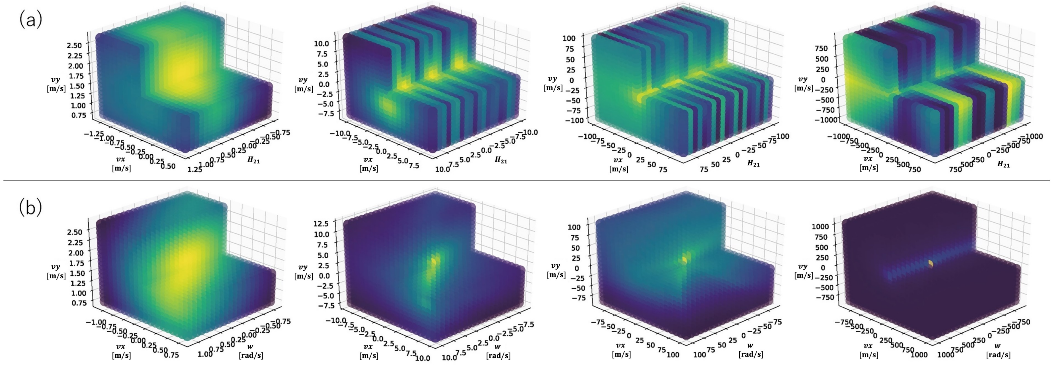

Figure 9. The landscape of the entropy near the true value, with and without the bird's-eye view transformation. (**a**) Homographic motion estimation. H_{21} refers to the (2,1) entry of the estimated homography matrix. (**b**) Proposed method.

3.3. Evaluation with Real Data

In this section, we describe the application of our method to real data captured by the event camera and the evaluation of the results in order to show the potential of our method for real-world applications. As shown in Figure 10, an event camera was mounted on a dolly angled (non-parallel) to the surface of interest, and the event camera motion estimation was performed when running on a carpeted surface. The event camera used was a Prophesee Gen3 VGA (and its supporting evaluation kit, EVK), installed at an angle of 45° to the ground. As a reference value, motion estimation with the camera facing the front was also performed at the same time using ARKit4 (iOS). In this experiment, the created IWE was saved as a map and reused for more accurate motion estimation.

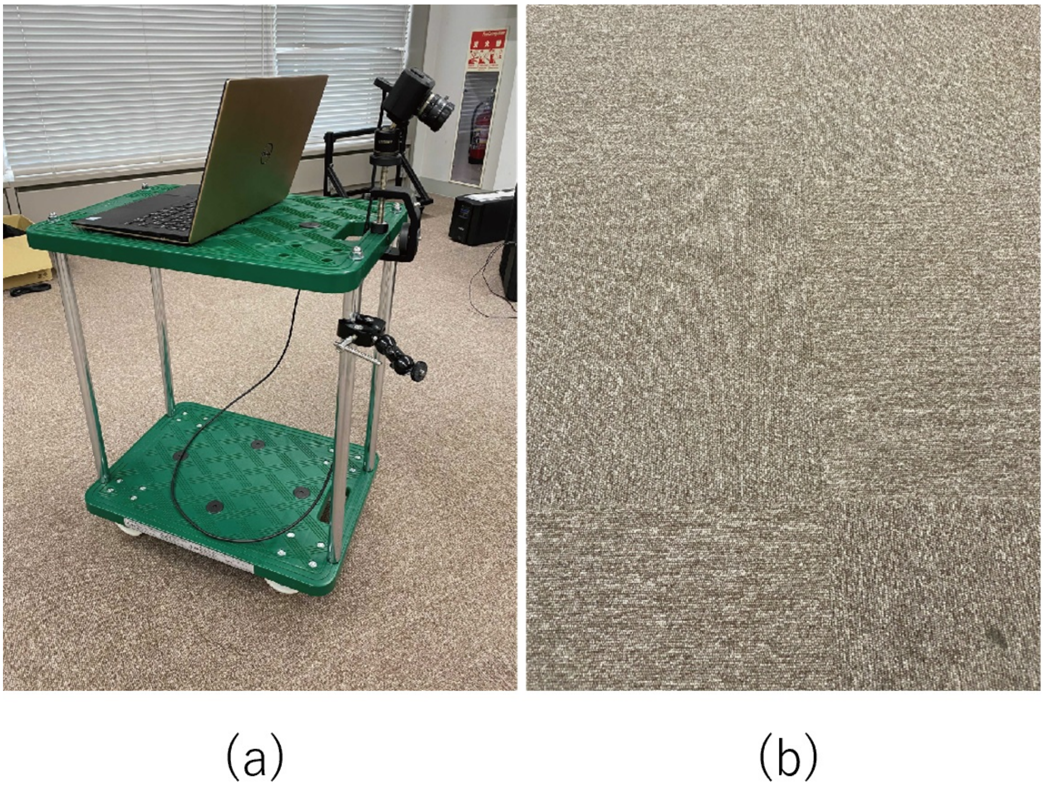

Figure 10. (**a**) Experimental setup. An event camera was mounted on a dolly at an angle (not parallel) to the surface of interest. (**b**) Carpeted surface; it would be difficult to extract its features with a normal camera.

The estimated parameters are shown in Figure 11 along with the results of the homographic motion estimation and reference values. It can be seen that while the homographic motion estimation failed, the accuracy of our estimation was comparable to that of the motion estimation by the front RGB camera and light detection and ranging (LIDAR). Figure 12 shows the location obtained from the estimated θ reference value from ARKit4 and the plotting result of the created IWE map. The IWE map shows the carpeted surface without motion blur, indicating that the motion estimation was performed correctly even on a surface where robust feature point extraction and tracking were difficult. Because it is difficult for a normal camera to perform stable tracking from a surface as the one shown in the figure, this evaluation further demonstrated the effectiveness of our motion estimation method while using an event camera.

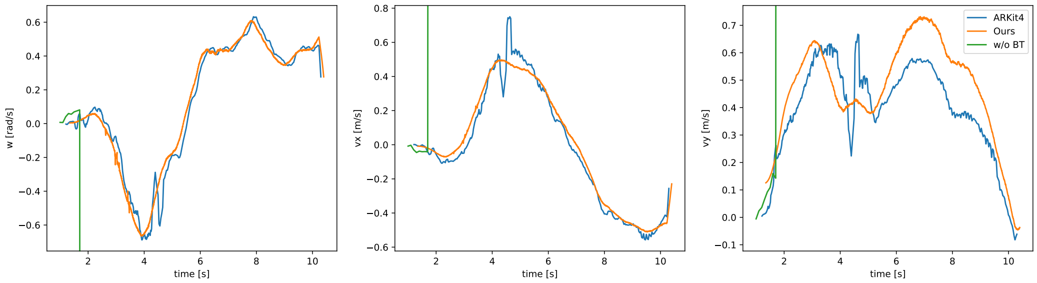

Figure 11. Results of the motion estimation experiment on a carpeted surface. Blue: ARKit4. Orange: proposed method with the bird's-eye view transformation. Green: entropy minimization method without the bird's-eye view transformation.

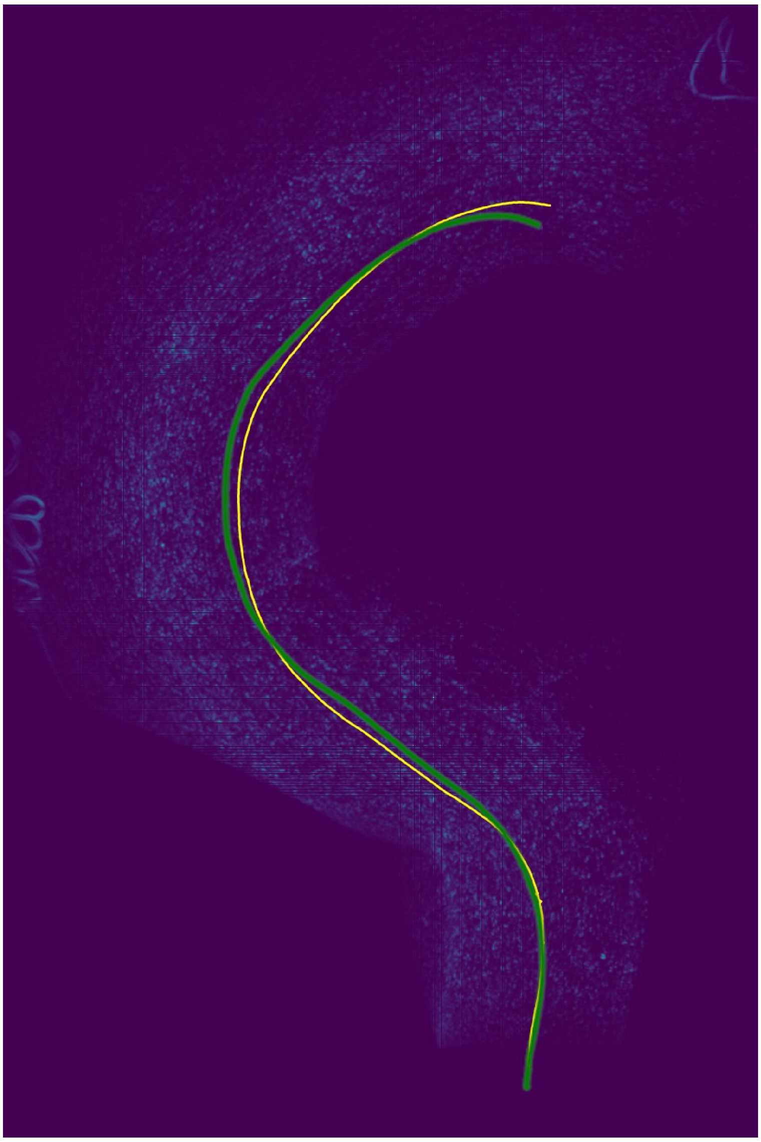

Figure 12. Result of our event camera motion estimation method on a carpeted surface (Green: location of the dolly obtained by ARKit4. Yellow: location of the dolly estimated by the proposed method). Our method was able to estimate motion even on a surface where robust feature point extraction and tracking were difficult.

4. Discussion

Previous studies have shown that contrast maximization and entropy minimization can cause the loss function to have multiple extremes in some situations [33,34]. In this study, we clearly showed that when using contrast maximization and entropy minimization to perform homographic motion estimation for a plane, the loss function has multiple extremes and the motion estimation result deviates from the true value. Such errors are considered to be a serious problem, especially when event camera motion estimation is used for autonomous driving. To solve this problem, we proposed a method of bird's-eye transformation of event data into two dimensions parallel to the road surface. As shown in the results of the loss landscape experiment, the proposed method mitigates the problem that the loss function becomes non-convex around the true value. As mentioned in the results of the quantitative experiments using the CARLA simulator, in the task of motion estimation with an event camera mounted at an angle, the proposed method outperforms the conventional method in terms of accuracy, number of computations, and computation time. Furthermore, in an experiment with an event camera attached to a dolly, we qualitatively showed that the proposed method is capable of motion estimation. From these experiments, we concluded that our method can accurately estimate the motion of an event camera mounted on a vehicle or robot by using contrast maximization, which has caused errors in the previous methods.

However, one limitation of our method is that although the computation time was reduced compared to the conventional method, real-time motion estimation is still not possible. As Gallego et al. stated, event camera algorithms can be broadly divided into two categories depending on how the events are processed: (1) those that operate on an event-by-event basis, where the arrival of a single event changes the state of the system, and (2) those that operate on a group of events [7]. Since we believe that the former method takes less time to estimate than the latter, we plan to improve our approach by modifying the contrast maximization method from processing groups of events to processing each event individually.

5. Conclusions

In this paper, we proposed a method of camera motion estimation using a road surface observed at an angle by an event camera. By using a bird's-eye view transformation, events were transformed into two dimensions parallel with the road surface, which mitigates the problem of the loss function becoming a non-convex function, which occurs in conventional methods. As a result, the loss function becomes convex around the true value, which enables a wider range of the initial value to be taken, resulting in highly accurate and fast motion estimation. Experiments using CARLA and real data demonstrated the effectiveness of the proposed method in terms of both speed and accuracy.

Author Contributions: Conceptualization, T.O., Y.S. and H.S.; methodology, T.O., Y.S. and H.S.; software, T.O.; validation, T.O.; formal analysis, T.O., Y.S. and H.S.; investigation, T.O.; resources, Y.S. and H.S.; data curation, T.O.; writing—original draft preparation, T.O.; writing—review and editing, T.O., Y.S. and H.S.; visualization, T.O.; supervision, Y.S. and H.S.; project administration, Y.S. and H.S.; funding acquisition, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Teshima, T.; Saito, H.; Ozawa, S.; Yamamoto, K.; Ihara, T. Vehicle Lateral Position Estimation Method Based on Matching of Top-View Images. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR), Hong Kong, China, 20–24 August 2006; Volume 4, pp. 626–629.
2. Saurer, O.; Vasseur, P.; Boutteau, R.; Demonceaux, C.; Pollefeys, M.; Fraundorfer, F. Homography Based Egomotion Estimation with a Common Direction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 327–341. [CrossRef] [PubMed]
3. Gilles, M.; Ibrahimpasic, S. Unsupervised deep learning based ego motion estimation with a downward facing camera. *Vis. Comput.* **2021**. [CrossRef]
4. Ke, Q.; Kanade, T. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Madison, WI, USA, 18–20 June 2003; Volume 1, p. I.
5. Nourani-Vatani, N.; Roberts, J.; Srinivasan, M.V. Practical visual odometry for car-like vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009; pp. 3551–3557.
6. Gao, L.; Su, J.; Cui, J.; Zeng, X.; Peng, X.; Kneip, L. Efficient Globally-Optimal Correspondence-Less Visual Odometry for Planar Ground Vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 2696–2702.
7. Gallego, G.; Delbruck, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A.; Conradt, J.; Daniilidis, K.; et al. Event-based Vision: A Survey. *arXiv* **2019**, arXiv:1904.08405.
8. Brandli, C.; Berner, R.; Yang, M.; Liu, S.C.; Delbruck, T. A 240 × 180 130 dB 3 µs Latency Global Shutter Spatiotemporal Vision Sensor. *IEEE J. Solid-State Circuits* **2014**, *49*, 2333–2341. [CrossRef]
9. Lichtsteiner, P.; Posch, C.; Delbruck, T. A 128× 128 120 dB 15 µs Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE J. Solid-State Circuits* **2008**, *43*, 566–576. [CrossRef]
10. Gallego, G.; Rebecq, H.; Scaramuzza, D. A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3867–3876.
11. Gallego, G.; Gehrig, M.; Scaramuzza, D. Focus is all you need: Loss functions for event-based vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
12. Stoffregen, T.; Kleeman, L. Event cameras, contrast maximization and reward functions: An analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 12300–12308.
13. Nunes, U.M.; Demiris, Y. Entropy Minimisation Framework for Event-Based Vision Model Estimation. In *European Conference on Computer Vision (ECCV)*; Springer International Publishing: Cham, Switzerland, 2020; pp. 161–176.
14. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; Volume 78, pp. 1–16.
15. Benosman, R.; Ieng, S.H.; Clercq, C.; Bartolozzi, C.; Srinivasan, M. Asynchronous frameless event-based optical flow. *Neural Netw.* **2012**, *27*, 32–37. [CrossRef] [PubMed]
16. Rueckauer, B.; Delbruck, T. Evaluation of Event-Based Algorithms for Optical Flow with Ground-Truth from Inertial Measurement Sensor. *Front. Neurosci.* **2016**, *10*, 176. [CrossRef] [PubMed]
17. Brosch, T.; Tschechne, S.; Neumann, H. On event-based optical flow detection. *Front. Neurosci.* **2015**, *9*, 137. [CrossRef] [PubMed]
18. Benosman, R.; Clercq, C.; Lagorce, X.; Ieng, S.H.; Bartolozzi, C. Event-based visual flow. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 407–417. [CrossRef] [PubMed]
19. Delbruck, T. Frame-free dynamic digital vision. In Proceedings of the International Symposium on Secure-Life Electronics, Advanced Electronics for Quality Life and Society, Tokyo, Japan, 6–7 March 2008; Volume 1, pp. 21–26.
20. Liu, M.; Delbruck, T. Block-matching optical flow for dynamic vision sensors: Algorithm and FPGA implementation. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017.
21. Almatrafi, M.; Baldwin, R.; Aizawa, K.; Hirakawa, K. Distance Surface for Event-Based Optical Flow. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 1547–1556. [CrossRef] [PubMed]
22. Yu, J.J.; Harley, A.W.; Derpanis, K.G. Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness. In *European Conference on Computer Vision (ECCV)*; Springer: Cham, Switaerland, 2016; pp. 3–10.
23. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T. Flownet 2.0: Evolution of optical flow estimation with deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2462–2470.
24. Meister, S.; Hur, J.; Roth, S. UnFlow: Unsupervised Learning of Optical Flow With a Bidirectional Census Loss. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
25. Zhu, A.Z.; Yuan, L.; Chaney, K.; Daniilidis, K. EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras. *arXiv* **2018**, arXiv:1802.06898.
26. Paredes-Vallés, F.; Scheper, K.Y.W.; de Croon, G.C.H.E. Unsupervised Learning of a Hierarchical Spiking Neural Network for Optical Flow Estimation: From Events to Global Motion Perception. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2051–2064. [CrossRef] [PubMed]

27. Lee, C.; Kosta, A.K.; Zhu, A.Z.; Chaney, K.; Daniilidis, K.; Roy, K. Spike-FlowNet: Event-Based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 366–382.
28. Weikersdorfer, D.; Conradt, J. Event-based particle filtering for robot self-localization. In Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China, 11–14 December 2012; pp. 866–870.
29. Weikersdorfer, D.; Hoffmann, R.; Conradt, J. Simultaneous Localization and Mapping for Event-Based Vision Systems. In *Computer Vision Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 133–142.
30. Kim, H.; Handa, A.; Benosman, R.; Ieng, S.H.; Davison, A.J. Simultaneous mosaicing and tracking with an event camera. *J. Solid State Circ.* **2008**, *43*, 566–576.
31. Kim, H.; Leutenegger, S.; Davison, A.J. Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera. In *Computer Vision—ECCV 2016*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 349–364.
32. Gallego, G.; Scaramuzza, D. Accurate Angular Velocity Estimation With an Event Camera. *IEEE Robot. Autom. Lett.* **2017**, *2*, 632–639. [CrossRef]
33. Liu, D.; Parra, A.; Chin, T.J. Globally Optimal Contrast Maximisation for Event-based Motion Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6349–6358.
34. Peng, X.; Wang, Y.; Gao, L.; Kneip, L. Globally-optimal event camera motion estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 51–67.
35. Rebecq, H.; Gehrig, D.; Scaramuzza, D. ESIM: An open event camera simulator. In Proceedings of the Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018; pp. 969–982.
36. Hunter, J.D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [CrossRef]