sensors

MDPI

Article

A Spatial-Motion-Segmentation Algorithm by Fusing EDPA and Motion Compensation

Xinghua Liu [1,*], Yunan Zhao [1], Lei Yang [1] and Shuzhi Sam Ge [2]

[1] School of Electrical Engineering, Xi'an University of Technology, Xi'an 710048, China
[2] Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077, Singapore
* Correspondence: liuxh@xaut.edu.cn

Abstract: Motion segmentation is one of the fundamental steps for detection, tracking, and recognition, and it can separate moving objects from the background. In this paper, we propose a spatial-motion-segmentation algorithm by fusing the events-dimensionality-preprocessing algorithm (EDPA) and the volume of warped events (VWE). The EDPA consists of depth estimation, linear interpolation, and coordinate normalization to obtain an extra dimension (Z) of events. The VWE is conducted by accumulating the warped events (i.e., motion compensation), and the iterative-clustering algorithm is introduced to maximize the contrast (i.e., variance) in the VWE. We established our datasets by utilizing the event-camera simulator (ESIM), which can simulate high-frame-rate videos that are decomposed into frames to generate a large amount of reliable events data. Exterior and interior scenes were segmented in the first part of the experiments. We present the sparrow search algorithm-based gradient ascent (SSA-Gradient Ascent). The SSA-Gradient Ascent, gradient ascent, and particle swarm optimization (PSO) were evaluated in the second part. In Motion Flow 1, the SSA-Gradient Ascent was 0.402% higher than the basic variance value, and 52.941% faster than the basic convergence rate. In Motion Flow 2, the SSA-Gradient Ascent still performed better than the others. The experimental results validate the feasibility of the proposed algorithm.

Keywords: event camera; motion segmentation; motion compensation; depth estimation; motion flow

Citation: Liu, X.; Zhao, Y.; Yang, L.; Ge, S.S. A Spatial-Motion-Segmentation Algorithm by Fusing EDPA and Motion Compensation. *Sensors* **2022**, *22*, 6732. https://doi.org/10.3390/s22186732

Academic Editors: Sylvain Girard and Christoph M. Friedrich

Received: 2 August 2022
Accepted: 2 September 2022
Published: 6 September 2022

1. Introduction

As a new type of sensing imaging device, event cameras, such as the dynamic vision sensor (DVS) or dynamic and active-pixel vision sensor (DAVIS) [1,2], are different from traditional cameras that shoot scenes or objects at a fixed frame rate [3]. The moving objects and light-intensity changes are focused on by event cameras. In general, the intensity change of each pixel is transmitted by a stream of asynchronous events. The information carried by an event includes the pixel coordinates (x, y), trigger time (t), and positive- or negative-event polarity (p). Compared with traditional cameras, event cameras have many advantages, for example, latency in microseconds, low motion blur, and high dynamic range (HDR) [4]. In computer vision, event cameras are often used in many fields, such as detection, tracking, recognition, and simultaneous localization and mapping (SLAM) [5,6].

In motion segmentation, independent motion-related pixels can be separated from the video sequence. Based on different motion types, the foreground objects are divided from the background by clustering these pixels [3,7]. Traditional vision-based processing algorithms usually assume that the video is shot under ideal circumstances [8–10]. Many factors are not taken into account by traditional vision tasks (e.g., moving speed, stability, and lighting conditions). In practical-application scenarios, the performance of motion-segmentation algorithms will be severely affected by fast-moving devices, such as drones and self-driving vehicles. The problem is well handled by motion segmentation based on event cameras [11]. The task of spatial-motion segmentation aims to separate the tracked feature points according to the respective rigid three-dimensional (3D) motion [12].

In this paper, we propose a framework by fusing the EDPA and 3D-motion-compensation approach to establish the spatial-motion-segmentation algorithm. The advantages of 3D information and motion segmentation are well combined by the proposed algorithm, which can not only satisfy the accuracy and stability requirements of vision works, but also separate the objects from the redundant background. The datasets of the experiments derive from the ESIM, which is an event-camera simulator [13]. After collecting the datasets, the pseudo-depth of the per-event is obtained by the EDPA in each frame of a pixel. The VWE is conducted by accumulating the warped events, and the iterative-clustering algorithm is presented to maximize the contrast in the VWE. The proposed algorithm has great segmentation effectiveness in low-speed- and high-speed-motion scenes, and in low-light and low-exposure scenes (e.g., high noon and night scenes). The main contributions of this paper are as follows:

(1) In order to inexpensively and conveniently extend 3D (2D plane (x, y) and time (t)) to 4D (3D space (x, y, z) and time (t)) for events without an RGB-depth-map (RGB-D) camera [14], or based on the light detection and ranging system (LiDAR), the EDPA is proposed for application in a variety of scenarios in this paper;
(2) Compared with the traditional frame-based and plane-based methods [15–17], the proposed algorithm combines the EDPA and 3D-motion compensation to accomplish the spatial-motion segmentation. The proposed algorithm provides a more refined segmentation model, and the segmentation accuracy can be greatly improved;
(3) The SSA-Gradient Ascent is presented to maximize the contrast in the VWE in this paper, which blends the advantages of SSA [18] and gradient ascent to obtain a better fitness value and faster convergence rate than other algorithms, such as PSO (see Section 4.2).

The remainder of this paper is organized as follows. In Section 2, we review some related works on depth estimation and motion segmentation. The mathematical model and main methods are presented in Section 3. Section 4 presents the process of the experiments. The conclusions are shown in Section 5.

2. Related Work

Depth estimation is one of the most important components of the EDPA. In [19], the authors introduced some of the high-accuracy and high-speed structured-light 3D-reconstruction methods. Deep learning enables us to accomplish visual tasks that are difficult to achieve by traditional geometry-based algorithms. Muhammad, K. et al. proposed a bidirectional long short-term memory (BiLSTM)-based attention mechanism with a dilated convolutional neural network (DCNN) to recognize the different human actions in videos [20]. A primary-prioritized recurrent deep reinforcement learning algorithm for dynamic-spectrum access based on cognitive-radio (CR) technology was proposed in [21]. In [22], a game-theory strategy was proposed to improve the energy-consumption performance of the MEC system. David et al. presented the first monocular depth estimation based on a convolutional neural network (CNN) [23]. Ibraheem et al. presented supervised learning without coupled geometric information [24]. To overcome the disadvantages of the RGB information-only-based method, an improved moving-object-detection method was proposed in [25]. Depth estimation based on event cameras has been gradually developed in recent years. In [26], the authors introduced the event-based multiview stereo (EMVS) to estimate semidense 3D structures with known trajectories. A novel 3D-reconstruction method was researched by Kim et al., which can perform real-time 3D reconstruction from a single hand-held event camera with no additional sensing [27]. In [28], a unifying framework was studied to solve several computer-vision problems with event cameras: motion, depth, and optical-flow estimation.

A variety of segmentation algorithms have been proposed by researchers to recognize and separate the objects from the background in the scene. Wang et al. proposed the EvDistill to learn a student network on the unlabeled and unpaired event data (target modality) via knowledge distillation (KD) [29]. In [30], the authors introduced a framework for the

segmentation of human-motion sequences based on sensor networks. Li et al. presented a novel approach, which is the motion segmentation of 3D-active-contour registration [31]. Event cameras are also widely made use of in motion segmentation. Stoffregen et al. presented a motion-compensation-based approach to establish the image of warped events (IWE) for segmentation [10]. The dataset was proposed by Mitrokhin et al. to perform the motion segmentation of event cameras [32]. The authors developed a technique for generalized motion segmentation based on spatial statistics across timeframes [33]. In [34], Zhou et al. introduced an event-based motion-segmentation method with spatiotemporal graph cuts, which cast the problem as an energy minimization one involving the fitting of multiple motion models.

3. Main Methods

The objectives of our research are to extend 4D events and realize spatial-motion segmentation. We assume the prior knowledge of the scene that we cannot know. Because the information carried by the per-event is limited, the events in the set $\varepsilon = \{e_k\}_{k=1}^{N_e}$ are processed so that we can obtain enough information. The depth maps of the images are obtained by depth estimation. The initial motion parameter $\boldsymbol{\theta} = \left[\theta_x, \theta_y, \theta_z\right]^T$ can be obtained by motion-flow estimation [35]. The contrast maximization is a framework that provides state-of-the-art results on several event-based computer-vision tasks [36]. A coherent motion is represented by a cluster. The problems we need to address are to classify the events into distinct clusters and maximize the contrast in the VWE.

3.1. Events from 3D to 4D by EDPA

The target of EDPA is to obtain an extra dimension (Z) for events. Firstly, the pre-pixel depths of $Frame_1$ and $Frame_2$ are derived by depth estimation as Z_{frame_1} and Z_{frame_2}, respectively. As shown in Figure 1a, we calculate the depth of the per-event by Equation (1). At the same time, the depth (Z) and events coordinates (x, y) are not in the same coordinate system. The coordinates need to be normalized by transformation. Then, we transform the pixel coordinates and depth (Z) into the same coordinate system through a global transformation, as shown in Equations (2) and (3) and Figure 1b. Finally, we have an extra dimension, as shown in Equation (4):

$$Z_k = \frac{t_k - T_{frame_1}}{T_{frame_2} - T_{frame_1}} \left| Z_{frame_2} - Z_{frame_1} \right| \tag{1}$$

where t_k is the time when the event (k) is triggered; T_{frame_j} is the time of the frame (j) in a video; Z_{frame_j} is the pixel depth on the frame (j), and Z_k is the depth of the event (k);

$$\frac{x_k}{X_k} = \frac{y_k}{Y_k} = \frac{f}{Z_k} \tag{2}$$

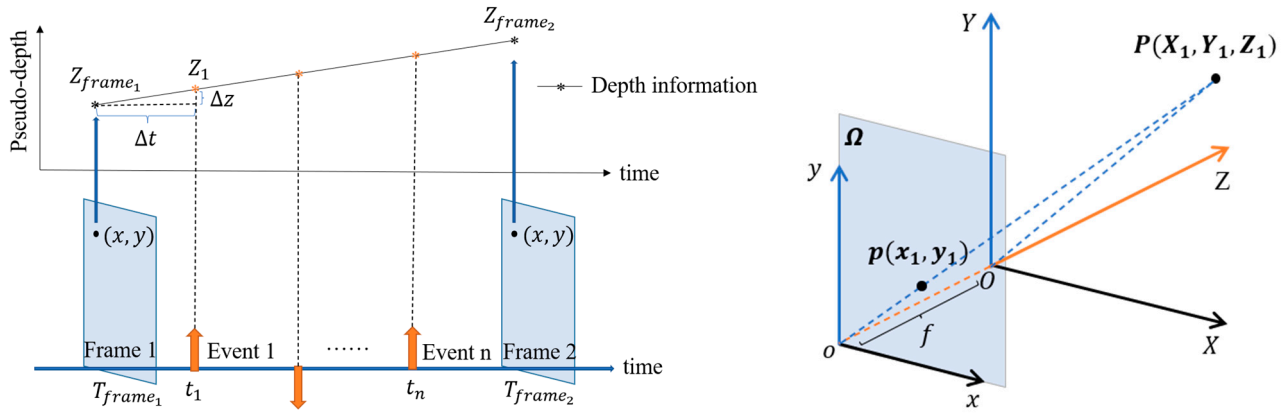
where (x_k, y_k) is the raw coordinate of the event (k); (X_k, Y_k, Z_k) are the global coordinates of the event (k); f is the scale factor;

Writing Equation (2) in matrix form, we have:

$$Z_k \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \\ Z_k \end{bmatrix} \tag{3}$$

$$\{e_k = (x_k, y_k, t_k)\}_{k=1}^{N_e} \rightarrow \{e_k = (X_k, Y_k, Z_k, t_k)\}_{k=1}^{N_e} \tag{4}$$

where e_k is the event (k) in the set, and N_e is the total number of events.



(**a**) Linear interpolation (**b**) Transformation from 2D to 3D

Figure 1. The events-dimensionality-preprocessing algorithm. (**a**) Two images are keyframes. The upward and downward arrows between the two images represent positive and negative events, respectively. (**b**) The point $p(x_1, y_1)$ is on the Ω plane, and the p is back-projected into the global coordinate system $O - X - Y - Z$ by coordinate transformation.

3.2. Motio-Flow Estimation

A method of voxel motion in space over time is described by the motion flow. The grayscale-invariance assumption is shown in Figure 2.

$$\boldsymbol{I}_{t_1}(x_1, y_1, z_1, t_1) = \boldsymbol{I}_{t_2}(x_2, y_2, z_2, t_2) \tag{5}$$

where $\boldsymbol{I}_{t_k}$ is the grayscale value at the time (t_k), and (x, y, z) are the coordinates of the voxel in space.

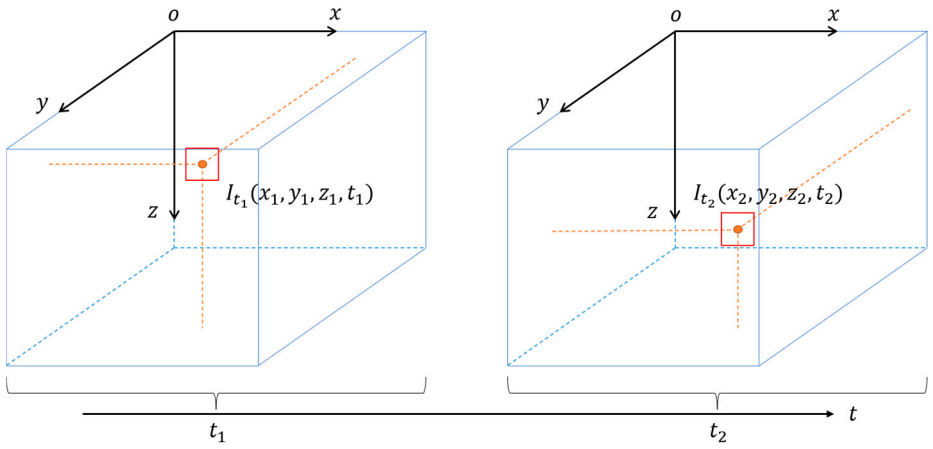


Figure 2. Grayscale-invariance assumption. According to the assumption, the grayscale value of the voxel is not changed between t_1 and t_2 in space.

As shown in Figure 2, according to the grayscale-invariance assumption, the movement of a voxel from t to $t + dt$ is shown in Equation (6):

$$\boldsymbol{I}(x + dx, y + dy, z + dz, t + dt) = \boldsymbol{I}(x, y, z, t) \tag{6}$$

When Taylor expansion is performed on the left side of the above equation and retains the first-order term, we will obtain:

$$\boldsymbol{I}(x + dx, y + dy, z + dz, t + dt) \approx \boldsymbol{I}(x, y, z, t) + \frac{\partial \boldsymbol{I}}{\partial x}dx + \frac{\partial \boldsymbol{I}}{\partial y}dy + \frac{\partial \boldsymbol{I}}{\partial z}dz + \frac{\partial \boldsymbol{I}}{\partial t}dt \tag{7}$$

Comparing the right sides of Equations (6) and (7), we have:

$$\frac{\partial \boldsymbol{I}}{\partial x}dx + \frac{\partial \boldsymbol{I}}{\partial y}dy + \frac{\partial \boldsymbol{I}}{\partial z}dz + \frac{\partial \boldsymbol{I}}{\partial t}dt = 0 \tag{8}$$

Both sides of the above equation are divided by dt, and $\frac{\partial \boldsymbol{I}}{\partial t}$ is moved to the right side of the equation:

$$\frac{\partial \boldsymbol{I}}{\partial x}\theta_x + \frac{\partial \boldsymbol{I}}{\partial y}\theta_y + \frac{\partial \boldsymbol{I}}{\partial z}\theta_z = -\frac{\partial \boldsymbol{I}}{\partial t} \tag{9}$$

where $\theta_x, \theta_y, \theta_z$ are the velocities in the x, y, z directions, respectively.

To obtain enough equations to estimate the $\boldsymbol{\theta}$, three planes (S_1, S_2, S_3) are created as $(x, y, t), (y, z, t), (z, x, t)$, respectively, and their grayscale values are $\boldsymbol{I}_1, \boldsymbol{I}_2, \boldsymbol{I}_3$, respectively, which are shown in Figure 3. Equation (9) can be written as:

$$\frac{\partial \boldsymbol{I_k}}{\partial i}\theta_i + \frac{\partial \boldsymbol{I_k}}{\partial j}\theta_j = -\frac{\partial \boldsymbol{I_k}}{\partial t} \tag{10}$$

where (i, j) is the element that belongs to $\{(x, y), (y, z), (z, x)\}$, and k is the index of the set that belongs to $\{1, 2, 3\}$. For example, when $k = 1$, $(i, j) = (x, y)$, Equation (10) can be written as:

$$\frac{\partial \boldsymbol{I}_1}{\partial x}\theta_x + \frac{\partial \boldsymbol{I}_1}{\partial y}\theta_y = -\frac{\partial \boldsymbol{I}_1}{\partial t} \tag{11}$$

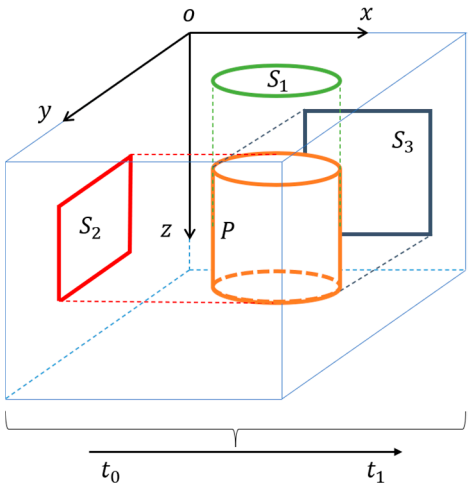


Figure 3. 2D projection of a 3D time surface (TS) of events. The 3D TS is a 3D map in which each voxel stores a single time value [37]. The object (P) is obtained by accumulating events from t_0 to t_1, and then projecting it onto three planes (S_1, S_2, S_3) to form TSs in space.

Equation (10) can be written as a matrix form:

$$\begin{pmatrix} I_{1,x} & I_{1,y} & 0 \\ 0 & I_{2,y} & I_{2,z} \\ I_{3,x} & 0 & I_{3,z} \end{pmatrix} \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = -\begin{pmatrix} \frac{\partial I_1}{\partial t} \\ \frac{\partial I_2}{\partial t} \\ \frac{\partial I_3}{\partial t} \end{pmatrix} \tag{12}$$

where $I_{1,x}$ is $k = 1$ and the partial derivative of I_1 to x.

$I_{k,(i,j)}$ are recorded as a_k and b_k, and the above equation can be written as:

$$\begin{pmatrix} a_1 & b_1 & 0 \\ 0 & a_2 & b_2 \\ b_3 & 0 & a_3 \end{pmatrix} \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = -\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \tag{13}$$

By calculating the optical flow, θ_x can be obtained such that:

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_x \\ \frac{-a_1\theta_x + c_1}{b_1} \\ \frac{-b_3\theta_x + c_3}{a_3} \end{pmatrix} = \begin{pmatrix} \theta_x \\ \theta_y \\ \theta_z \end{pmatrix} \tag{14}$$

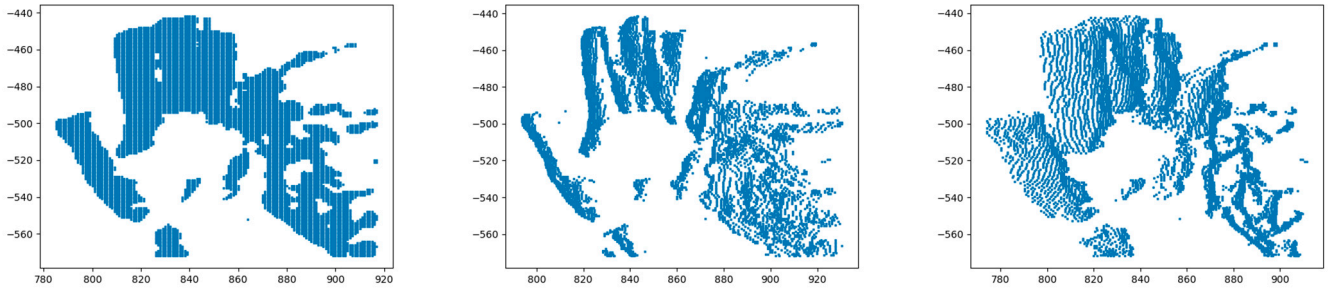
3.3. 3D-Motion-Compensation and the Iterative-Clustering Algorithm

As shown in Figure 4, the objects are warped in the direction of the optical flow, and the image becomes sharp; in the contrary case, the image becomes a blur. The warped events are given by:

$$\mathbf{x}_k = (x_k, y_k, z_k)^T \tag{15}$$

$$\mathbf{x}'_k = \mathbf{x}_k - \left(t_k - t_{ref}\right)\boldsymbol{\theta} = \mathbf{x}_k - \Delta t_k \boldsymbol{\theta} \tag{16}$$

where t_k is the trigger time of the event (k), and $\mathbf{x}'_k$ is the position where the event ($\mathbf{x}_k$) is warped.



(**a**) (**b**) (**c**)

Figure 4. Images of warped events (IWEs): (**a**) TS when the events are not warped; (**b**) events are warped in the direction of the "hand", and so the image of the "hand" becomes sharp, and the image of the "face" becomes a blur; (**c**) the condition presented by the image is counter to (**b**).

Similar to the IWEs, we accumulate the warped events according to the weight values. The definition of the VWE is:

$$V_j(\mathbf{x}) = \sum_{k=1}^{N_e} p_{kj}\delta\left(\mathbf{x} - \mathbf{x}'_{kj}\right) \tag{17}$$

where p_{kj} represents the probability that the event (k) belongs to the optical flow (j), and $\delta = \begin{cases} 1, \mathbf{x} - \mathbf{x}'_{kj} = 0 \\ 0, otherwise \end{cases}$ represents the Dirac function. $\mathbf{x}'_{kj}$ is the location, where the event (k) is warped along the optical flow (j).

Variance is employed to evaluate the contrast in this paper. To make the VWE sharper, the contrast is maximized as:

$$\mathrm{Var}(V_j) = \frac{1}{|\Omega|}\int_{\Omega}\left(V_j(\mathbf{x}) - \mu_{V_j}\right)^2 d\mathbf{x} \tag{18}$$

where Ω is the image volume, and μ_{V_j} denotes the mean of the VWE over the image volume (Ω).

For the discrete VWE, the above equation can be written as:

$$\mathrm{Var}(V_j) = \sum_{a,b,c\in\Omega}\left(V_{abc} - \mu_{V_j}\right)^2 \tag{19}$$

where a, b, c is the index of the voxel in spacetime.

Now, the $\boldsymbol{\theta}$ and $\mathbf{P}$ are needed to initialize and iterate. The elements of the $\mathbf{P}$ are the event-cluster membership probabilities, and the elements of the $\boldsymbol{\theta}$ are the motion parameters. Because our clustering type is motion flow, the k-means algorithm was used to find the cluster center to initialize the $\boldsymbol{\theta}$. The loss function of the k-means algorithm is defined as follows:

$$J(c, \lambda) = \min \sum_{i=1}^{M} \|x_i - \lambda_{c_i}\|^2 \tag{20}$$

where x_i represents the sample point (i), c is the cluster to which x_i belongs, λ_{c_i} represents the center point corresponding to the cluster, and M is the total number of samples.

To make the per-event have the same probability of being classified into the cluster, we initialize the $\mathbf{P}$:

$$\mathbf{P} = \frac{1}{N_l} \tag{21}$$

where N_l is the clusters number.

The above process can be summarized as the need to optimize the $\boldsymbol{\theta}$ and $\mathbf{P}$ that maximize the variance, and the $\boldsymbol{\theta}$ and $\mathbf{P}$ are updated by the iterative-clustering algorithm. Each iteration of the iterative-clustering algorithm has two steps: a fixed $\mathbf{P}$ to update the $\boldsymbol{\theta}$, and a fixed $\boldsymbol{\theta}$ to update the $\mathbf{P}$:

$$(\boldsymbol{\theta}^*, \mathbf{P}^*) = \underset{(\boldsymbol{\theta}, \mathbf{P})}{\operatorname{argmax}} \sum_{j=1}^{N_l} \operatorname{Var}(V_j) \tag{22}$$

The iteration of the $\boldsymbol{\theta}$ makes use of the SSA-Gradient Ascent. In SSA, the producers' location update is given by:

$$\theta_{i,j}^{t+1} = \begin{cases} \theta_{i,j}^{t} \cdot e^{\left(\frac{-i}{\alpha \cdot iter_{\max}}\right)} & if\ R_2 < ST \\ \theta_{i,j}^{t} + q \cdot l & if\ R_2 \geq ST \end{cases} \tag{23}$$

where t is the current iteration number, and $\theta_{i,j}^{t}$ represents the value of the dimension (j) of the sparrow (i) at iteration (t). The largest number of iterations is represented by $iter_{\max}$.$\alpha \in (0, 1]$, and q is a random number obeying a normal distribution. $R_2 \in [0, 1]$ represents the alarm value, and $ST \in [0.5, 1]$ represents the safety threshold ($l \in \mathbb{R}^{1 \times d}$).

The scroungers are updated by:

$$\theta_{i,j}^{t+1} = \begin{cases} q \cdot e^{\left(\frac{\theta_{worst}^{t} - \theta_{i,j}^{t}}{i^2}\right)} & if\ \mathrm{i} > \frac{n}{2} \\ \theta_{p}^{t+1} + \left|\theta_{i,j}^{t} - \theta_{p}^{t+1}\right| \cdot n^{+} \cdot l & otherwise \end{cases} \tag{24}$$

where θ_p is the optimal position currently occupied by the producers; θ_{worst} represents the present global worst position; $n \in \mathbb{R}^{1 \times d}$, and $n^{+} = n^{T}\left(nn^{T}\right)^{-1}$.

The sparrows that are aware of the danger are given by:

$$\theta_{i,j}^{t+1} = \begin{cases} \theta_{best}^{t} + \beta \cdot \left|\theta_{i,j}^{t} - \theta_{best}^{t}\right| & if\ f_i > f_g \\ \theta_{i,j}^{t} + K \cdot \left(\frac{\left|\theta_{i,j}^{t} - \theta_{worst}^{t}\right|}{(f_i - f_w) + \varepsilon}\right) & if\ f_i = f_g \end{cases} \tag{25}$$

where θ_{best} represents the present global best position; β is the step-size-controlled parameter; $K \in [-1, 1]$; f_g and f_W are the current global fitness values, which represent the best and worst fitness values, respectively; ε is a constant that avoids zero in the denominator.

The SSA is used to find the better $\boldsymbol{\theta}$ in the global area. Gradient ascent is used to obtain the best in the local area, as shown in (25).

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mu \nabla_{\boldsymbol{\theta}} \left(\sum_{j=1}^{N_l} \mathrm{Var}(V_j) \right) \tag{26}$$

where μ is the iteration step size.

The iteration of the **P** is given by:

$$p_{kj} = \frac{c_j(\mathbf{x}'_k(\boldsymbol{\theta}_j))}{\sum_{i=1}^{N_l} c_i(\mathbf{x}'_k(\boldsymbol{\theta}_i))} \tag{27}$$

where c_i is the VWE of the i-th; $\mathbf{x}'_k(\boldsymbol{\theta}_j)$ is the position where the event (k) is warped along the 3D-motion flow (j); p_{kj} is the probability that the event (k) belongs to the j-th.

The algorithm flowchart is shown in Figure 5. The Algorithm 1 and Figure 5 show the overall process of our method. Variance convergence can usually be judged by the following conditions:

(1) The variance curves tend to be horizontal, or the fluctuation changes a little;
(2) In the variance values, more than a certain constant can be considered variance convergence;
(3) The iterations will be interrupted when the number of iterations reaches a certain value.

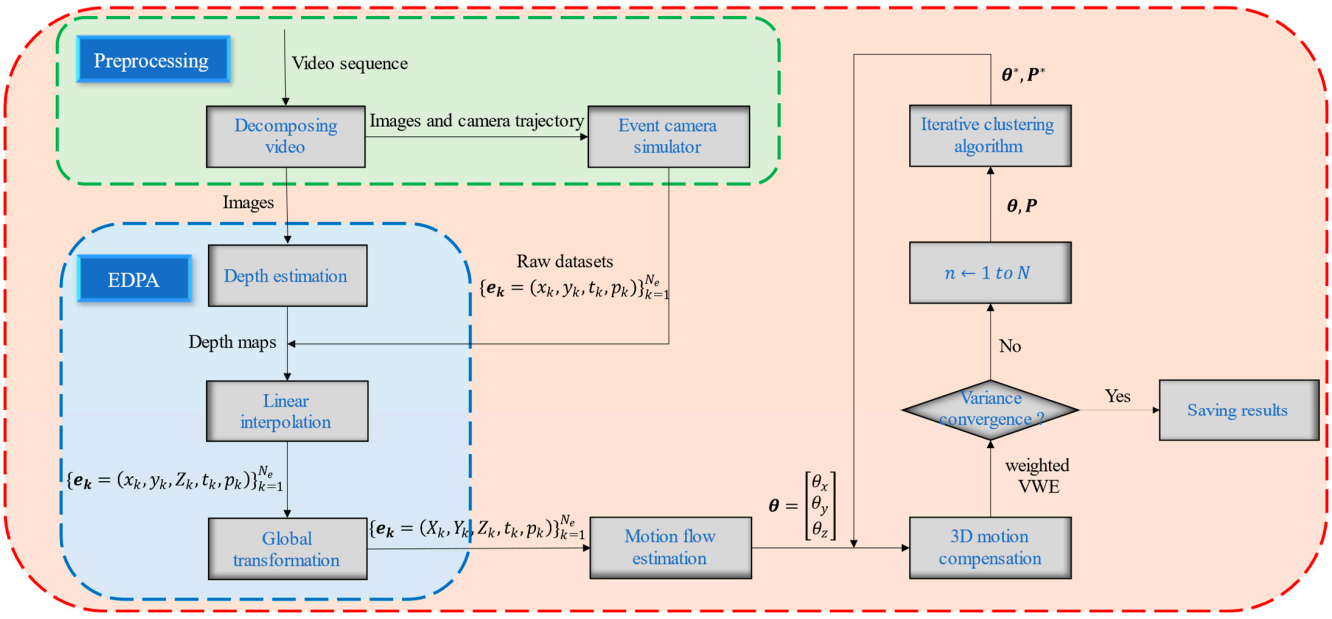


Figure 5. The spatial motion segmentation algorithm flowchart. The n is the current number of iterations. The N represents the total number of iterations required. In the whole algorithm, the dataset we need is obtained by the preprocessing part, and the extension of events from 3D to 4D is realized by the EDPA. The VWE is conducted by accumulating the warped events, and the iterative-clustering algorithm is used to maximize the contrast in the VWE.

Algorithm 1. Spatial-Motion Segmentation

Start

Input: Raw events, set $\varepsilon_1 = \{e_k = (x_k, y_k, t_k, p_k)\}_{k=1}^{N_e}$; original images; the number of clusters (N_l).

Output: Event-cluster membership probabilities (**P**), and motion parameters ($\boldsymbol{\theta}$).

Procedure:

1: For each image, **do** depth estimation.

End for;

2: For $k \leftarrow 1$ to N_e, **do:**

$$Z_k = \frac{t_k - T_{frame_1}}{T_{frame_2} - T_{frame_1}} \left| Z_{frame_2} - Z_{frame_1} \right|$$

$$Z_k \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \\ Z_k \end{bmatrix}$$

End for:

$\varepsilon_2 = \{e_k = (X_k, Y_k, Z_k, t_k, p_k)\}_{k=1}^{N_e}$

3: For e_k, **do:**

computing $\boldsymbol{\theta}_k$ according to $\boldsymbol{\theta} = \left(\theta_x, \frac{-a_1\theta_x + c_1}{b_1}, \frac{-b_3\theta_x + c_3}{a_3}\right)^T$

End for:

4: For $k \leftarrow 1$ to N_e, **do:**

warping e_k according to $\mathbf{x}'_k = \mathbf{x}_k - \left(t_k - t_{ref}\right)\boldsymbol{\theta} = \mathbf{x}_k - \Delta t_k \boldsymbol{\theta}$

End for:

5: For each warped e_k, **do** weighted accumulate:

$$V_j(\mathbf{x}) = \sum_{k=1}^{N_e} p_{kj}\delta\left(\mathbf{x} - \mathbf{x}'_{kj}\right)$$

6: For $i \leftarrow 1$ to ∞, **do:**

If the variance is not converged, **then:**

update the **P** using $p_{kj} = \frac{c_j(\mathbf{x}'_k(\boldsymbol{\theta}_j))}{\sum_{i=1}^{N_l} c_i(\mathbf{x}'_k(\boldsymbol{\theta}_i))}$

update the $\boldsymbol{\theta}$ according to SSA-Gradient Ascent:

computing $Var\left(V_j\right) = \sum_{a,b,c \in \Omega} \left(V_{abc} - \mu_{V_j}\right)^2$

End if:

$i \leftarrow i + 1$

End for:

End

4. Experiments

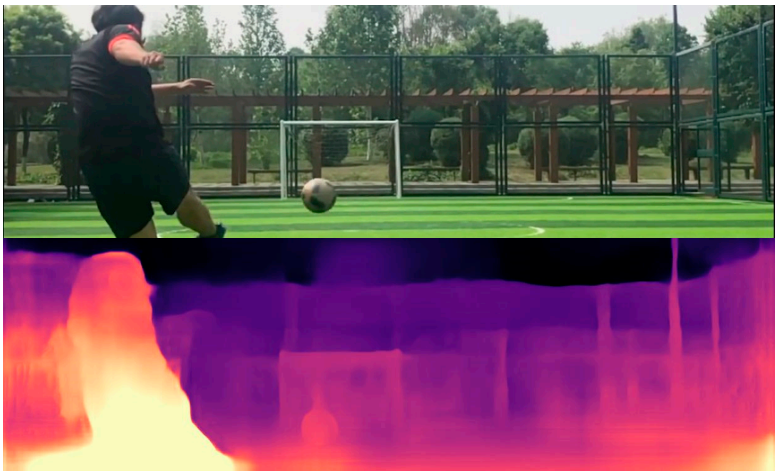
The performance of the proposed algorithm was evaluated by two experiments. The following experiments were conducted to verify the feasibility of the proposed algorithm. We utilized our datasets, which were derived from the event-camera simulator (ESIM). Our experimental process consisted of two parts: in the first part, the spatial-motion segmentation was implemented in the exterior scene and interior scene, and in the second part, the influence of the optimization-algorithm performance on the variance in the VWEs was estimated by SSA-Gradient Ascent, PSO, and gradient ascent.

4.1. Experiment 1: Spatial-Motion Segmentation of Exterior and Interior Scenes

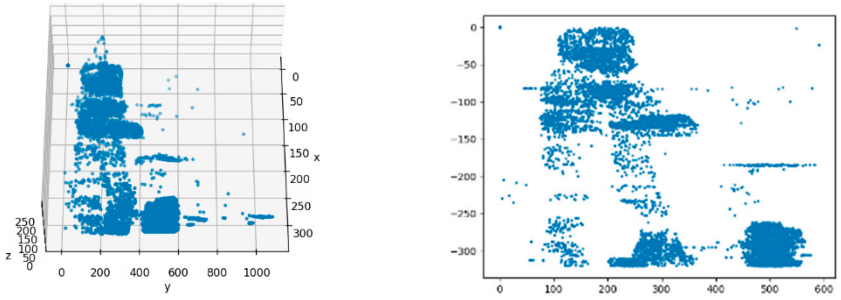
The Monodepth2 [38] depth-estimation network was utilized to predict the exterior scene in this paper, which was in a situation where the camera was stationary and only had moving objects. The predicted result is shown in Figure 6a, which displays the depth map of the first keyframe. The EDPA needs to utilize two images to expand the dimension of the events. After we obtained the depth map of the first key frame, the second key frame was selected for the depth estimation based on the trigger time of the 15,000th event. The spatial-motion segmentation was performed as shown in Figure 6b,c. Figure 6b is warped in the wrong direction, which is not the motion-flow direction of the objects. Figure 6c is the segmentation result by warping in the correct direction. When the algorithm iterates

to around the 10th round, the variance has been maximized, and we can consider that the variance has converged. The iterative plots of variance are shown in Figure 6d. The changes in the $\boldsymbol{\theta}$ are shown in Figure 6e–g. Six curves represent lateral-velocity vectors ($\boldsymbol{\theta_x}$), column-velocity vectors ($\boldsymbol{\theta_y}$) and velocity vectors on the z-axis ($\boldsymbol{\theta_z}$).

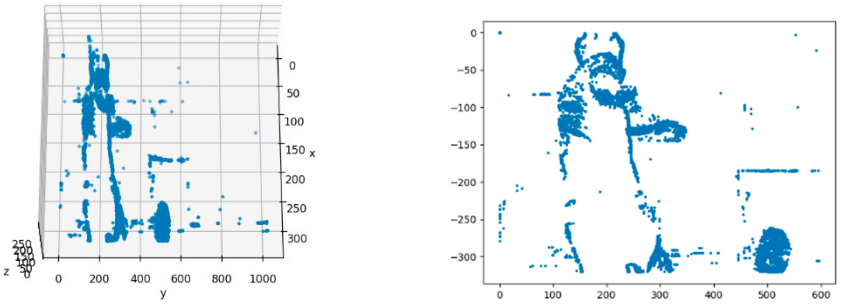
The fully convolutional residual network (FCRN) [39] was applied to predict the interior scene, as shown in Figure 7a. In Figure 7b,c, two carts along the upper-left and lower-right corners of the table with different directions and speeds were successfully separated into different classes in 3D space, and little redundant background information was segmented into the foreground. Experimenting with the objects moving in different directions demonstrated the feasibility of the proposed algorithm in the interior scene.



(**a**) Original image and depth map of the first keyframe

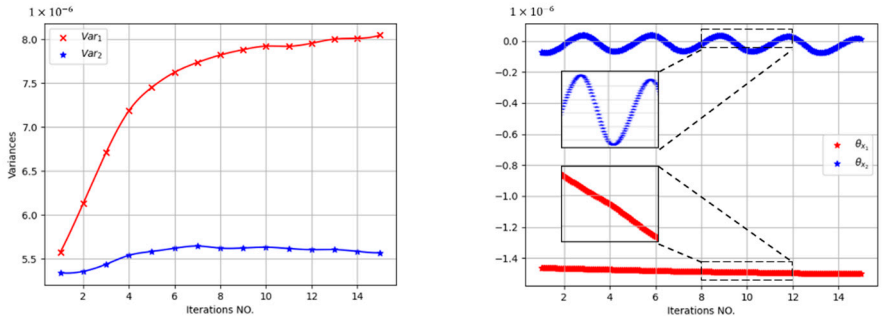


(**b**) VWE of warping in the wrong direction

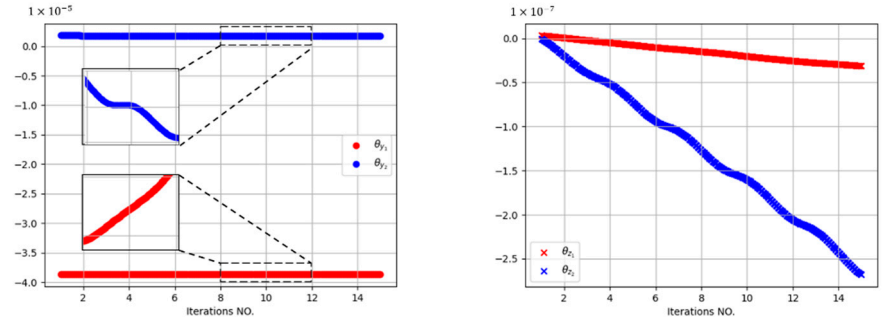


(**c**) VWE of warping in the correct direction

Figure 6. *Cont.*

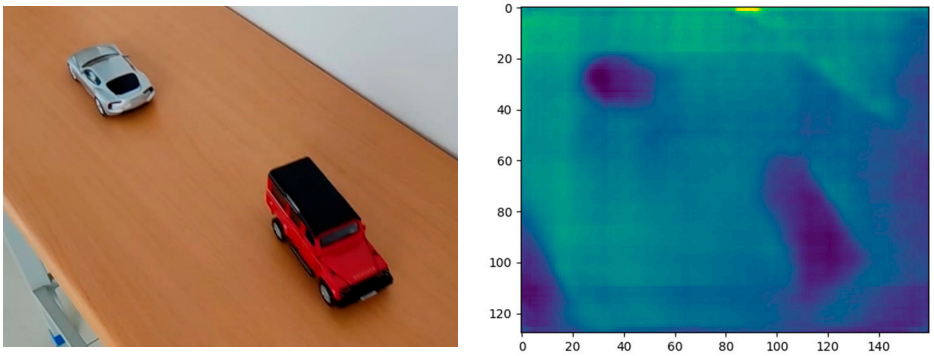


(**d**) Variance-convergence curves (**e**) $\boldsymbol{\theta_x}$ curves

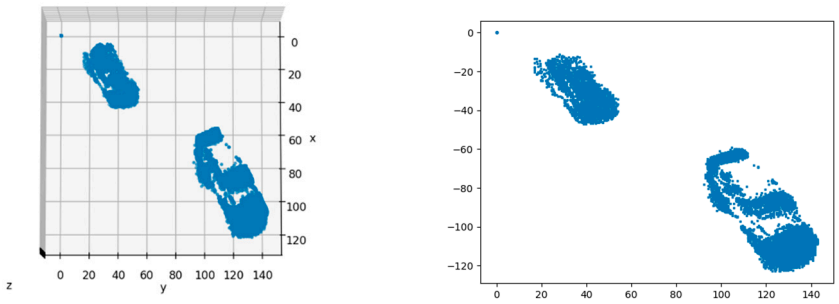


(**f**) $\boldsymbol{\theta_y}$ curves (**g**) $\boldsymbol{\theta_z}$ curves

Figure 6. Results of exterior-scene experiment. (**a**) The left is an image of the original scene. The right is the predicted depth map. The time of the first frame comes from the first event's trigger time. (**b**,**c**) Warping events to obtain the VWE. The initial value of the velocity vectors of warping events comes from the k-means algorithm. (**d**) Variance 1 and Variance 2 represent the variance change in the VWE after events are warped along with different motion flows. (**e**–**g**) Iterating process of motion parameters ($\boldsymbol{\theta}$).

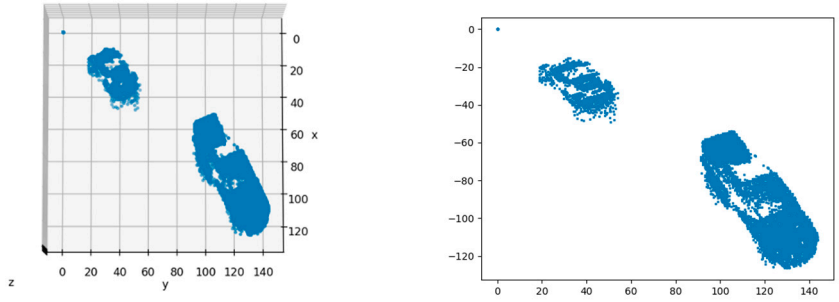


(**a**) Original image and depth map of the first frame

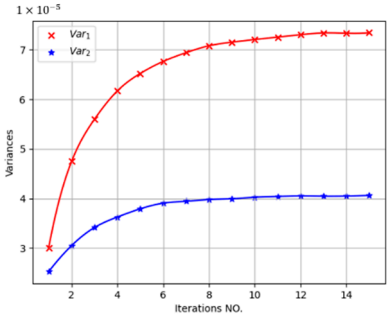


(**b**) VWE of warping in Motion Flow 1

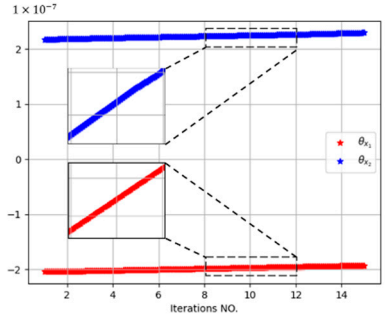
Figure 7. *Cont.*



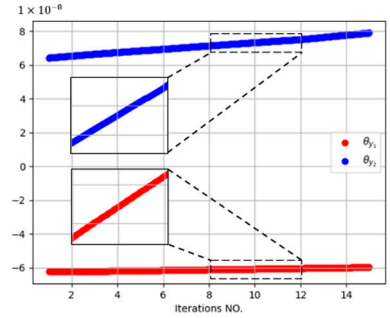
(**c**) VWE of warping in Motion Flow 2



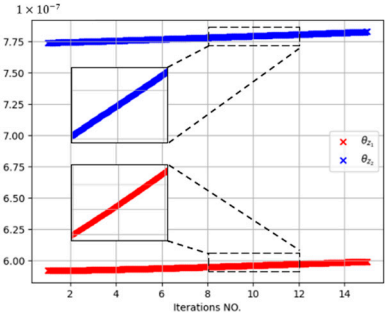
(**d**) Variance-convergence curves



(**e**) $\boldsymbol{\theta_x}$ curves



(**f**) $\boldsymbol{\theta_y}$ curves



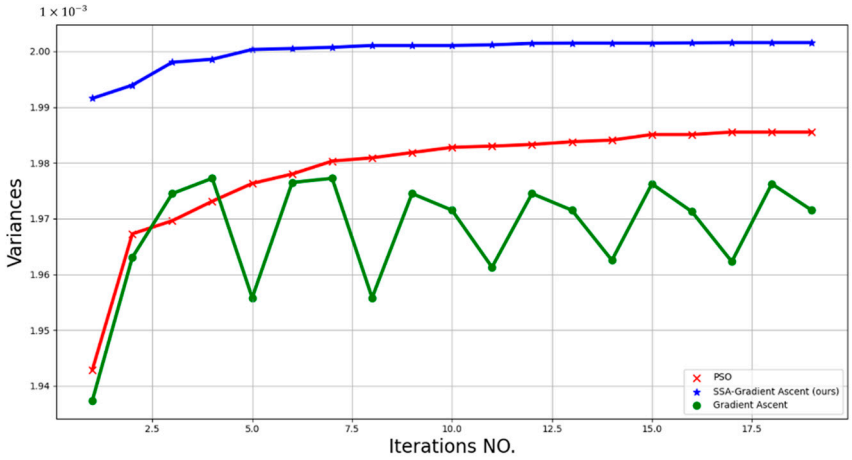
(**g**) $\boldsymbol{\theta_z}$ curves

Figure 7. Results of interior-scene experiment. (**a**) Original image and predicted depth map. (**b**) Events are warped by the motion flow towards the lower-right corner. The red car is sharpened, and the silver car is blurred. (**c**) Events are warped by the motion flow towards the upper-left corner, the silver car being sharpened, and the red car being blurred. (**d**) Variance 1 and Variance 2 represent the variance change in the VWE after events are warped, along with different motion flows. (**e**–**g**) Iterating process of motion parameters ($\boldsymbol{\theta}$).

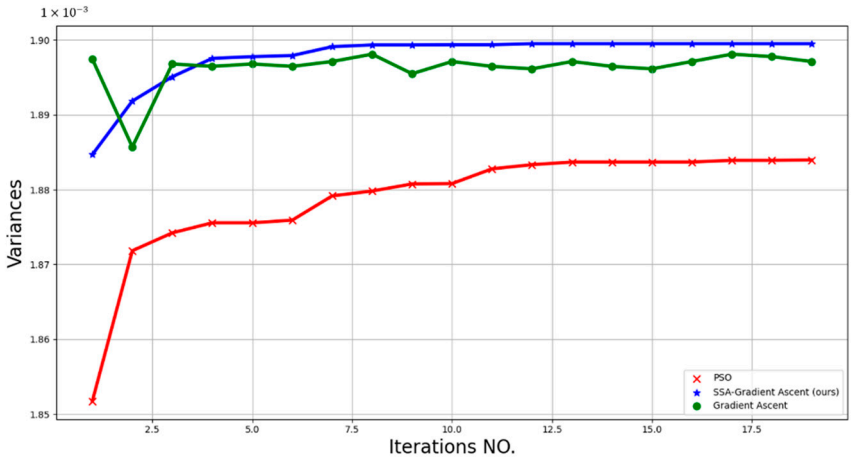
4.2. Experiment 2: Comparison of the Effects of Different Optimization Algorithms

The SSA has a preferable ability in global optimization, while it is weak in local searches. The gradient ascent has an excellent performance in local optimization, but it easily falls into the local extreme value. The SSA-Gradient Ascent combines the advantages of SSA and gradient ascent. The updates of the $\boldsymbol{\theta}$ by the SSA-Gradient Ascent, PSO, and gradient ascent were evaluated in this part, as shown in Figure 8. The variance-convergence rate of the algorithms and the value of the variance after convergence are listed in Table 1. As presented by the data in Table 1 and Figure 8, we chose the PSO method as the benchmark value. In Motion Flow 1, the proposed algorithm outperforms the basic value, with the gradient ascent 0.402% lower than the baseline. In Motion Flow 2, the gradient ascent is 0.731% higher than the basic value, while the proposed algorithm is 0.819% higher than the basic value. The same initial value of the $\boldsymbol{\theta}$ is used by the above algorithms, and the maximum-variance value of the SSA-Gradient Ascent indicates that the proposed algorithm is less dependent on initialization. The gradient-ascent method

does not converge until 19 iterative rounds. Compared with PSO, the proposed algorithm's convergence rate was 52.941% higher in Motion Flow 1, and 46.154% higher in Motion Flow 2. The better performance of the proposed algorithm was confirmed according to this experiment.



(**a**)



(**b**)

Figure 8. The variance convergence curves of VWEs when the $\boldsymbol{\theta}$ is optimized by SSA-Gradient Ascent, PSO, and gradient ascent. (**a**) Variance curves along the direction of Motion Flow 1 (**b**) Variance curves along the direction of Motion Flow 2.

Table 1. Comparison of algorithm performances.

		Motion Flow 1		Motion Flow 2	
Variance value	**SSA-Gradient Ascent (ours)**	**0.00200104**	**+0.800%**	**0.00189911**	**+0.819%**
	PSO	0.00198522	0%	0.00188369	0%
	Gradient Ascent	0.00197723	−0.402%	0.00189746	+0.731%
Convergence speed (iteration no.)	**SSA-Gradient Ascent (ours)**	**8**	**+52.941%**	**7**	**+46.154%**
	PSO	17	0%	13	0%
	Gradient Ascent	None		None	

5. Conclusions

A spatial-motion-segmentation algorithm is proposed in this paper, which has fused the EDPA and 3D-motion-compensation approach. The accuracy for tasks such as feature

detection in complex environments is addressed by the proposed algorithm, while the advantages of 3D information and motion segmentation are well combined. The pseudo-depth of the per-event was obtained by the EDPA in each frame of a pixel. The VWE was conducted by accumulating the warped events, and a spatial-motion-segmentation algorithm is presented to maximize the contrast in the VWE. Interior and exterior scenes were segmented in the first part of the experiments. In the second part, the effect on the variance in the VWEs was estimated by SSA-Gradient Ascent, PSO, and gradient ascent. In Motion Flow 1, the SSA-Gradient Ascent was 0.402% higher than the basic variance value, and 52.941% faster than the basic convergence rate. In Motion Flow 2, the gradient ascent was 0.731% higher than the basic value, and 46.154% faster than the basic convergence rate. As a result, the experimental results validate the feasibility of the proposed algorithm. For future research, more complex scenes and conspicuous-spatial-motion segmentation will be studied by us.

Author Contributions: Conceptualization, L.Y. and S.S.G.; Investigation, X.L., Y.Z., L.Y. and S.S.G.; Methodology, X.L., Y.Z. and S.S.G.; Project administration, X.L.; Supervision, X.L.; Validation, Y.Z.; Writing—original draft, Y.Z.; Writing—review & editing, X.L. and L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China under Grant U2003110, and in part by the Key Laboratory Project of Shaanxi Provincial Department of Education (No. 20JS110).

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. 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]
2. Brandli, C.; Berner, R.; Yang, M.; Liu, S.; 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]
3. Gallego, G.; Delbruck, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A.J.; Conradt, J.; Daniilidis, K.; et al. Event-Based Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 154–180. [CrossRef] [PubMed]
4. Rebecq, H.; Ranftl, R.; Koltun, V.; Scaramuzza, D. High Speed and High Dynamic Range Video with an Event Camera. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1964–1980. [CrossRef]
5. Duo, J.; Zhao, L. An Asynchronous Real-Time Corner Extraction and Tracking Algorithm for Event Camera. *Sensors* **2021**, *21*, 1475. [CrossRef]
6. Iaboni, C.; Lobo, D.; Choi, J.-W.; Abichandani, P. Event-Based Motion Capture System for Online Multi-Quadrotor Localization and Tracking. *Sensors* **2022**, *22*, 3240. [CrossRef] [PubMed]
7. Mohamed, E.; Ewaisha, M.; Siam, M.; Rashed, H.; Yogamani, S.; Hamdy, W.; El-Dakdouky, M.; El-Sallab, A. Monocular Instance Motion Segmentation for Autonomous Driving: KITTI InstanceMotSeg Dataset and Multi-Task Baseline. In Proceedings of the IEEE Intelligent Vehicles Symposium, Nagoya, Japan, 11–17 July 2021; pp. 114–121.
8. Bradski, G.; Davis, J. Motion segmentation and pose recognition with motion history gradients. *Mach. Vis. Appl.* **2002**, *13*, 174–184. [CrossRef]
9. Zappella, L.; Lladó, X.; Salvi, J. Motion segmentation: A review. *Artif. Intell. Res. Dev.* **2008**, *184*, 398–407.
10. Mattheus, J.; Grobler, H.; Abu-Mahfouz, A.M. A Review of Motion Segmentation: Approaches and Major Challenges. In Proceedings of the International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Kimberley, South Africa, 25–27 November 2020; pp. 1–8.
11. Stoffregen, T.; Gallego, G.; Drummond, T.; Kleeman, L.; Scaramuzza, D. Event-Based Motion Segmentation by Motion Compensation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 7243–7252.
12. Xu, X.; Cheong, L.-F.; Li, Z. 3D Rigid Motion Segmentation with Mixed and Unknown Number of Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1–16. [CrossRef]
13. Rebecq, H.; Gehrig, D.; Scaramuzza, D. ESIM: An Open Event Camera Simulator. In Proceedings of the Conference on Robot Learning (CoRL), Zurich, Switzerland, 29–31 October 2018; pp. 969–982.

14. Endres, F.; Hess, J.; Sturm, J.; Cremers, D.; Burgard, W. 3-D Mapping With an RGB-D Camera. *IEEE Trans. Robot.* **2014**, *30*, 177–187. [CrossRef]
15. Lipton, A.J.; Fujiyoshi, H.; Patil, R.S. Moving target classification and tracking from real-time video. In Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV), Princeton, NJ, USA, 19–21 October 1998; pp. 8–14.
16. Chen, Y.-M.; Bajic, I.V. A Joint Approach to Global Motion Estimation and Motion Segmentation from A Coarsely Sampled Motion Vector Field. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 1316–1328. [CrossRef]
17. Jeong, S.; Lee, C.; Kim, C. Motion-Compensated Frame Interpolation Based on Multihypothesis Motion Estimation and Texture Optimization. *IEEE Trans. Image Process.* **2013**, *22*, 4497–4509. [CrossRef]
18. Xue, J.; Shen, B. A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
19. Li, B.; An, Y.; Cappelleri, D.; Xu, J.; Zhang, S. High-accuracy, high-speed 3D structured light imaging techniques and potential applications to intelligent robotics. *Int. J. Intell. Robot. Appl.* **2017**, *1*, 86–103. [CrossRef]
20. Muhammad, K.; Ullah, A.; Imran, A.S.; Sajjad, M.; Kiran, M.S.; Sannino, G.; de Albuquerque, V.H.C. Human action recognition using attention-based LSTM network with dilated CNN features. *Future Gener. Comput. Syst.* **2021**, *125*, 820–830. [CrossRef]
21. Chen, M.; Liu, A.; Liu, W.; Ota, K.; Dong, M.; Xiong, N.N. RDRL: A Recurrent Deep Reinforcement Learning Scheme for Dynamic Spectrum Access in Reconfigurable Wireless Networks. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 364–376.
22. Chen, M.; Liu, W.; Wang, T. Zhang, S.; Liu, A. A game-based deep reinforcement learning approach for energy-efficient computation in MEC systems. *Knowl.-Based Syst.* **2022**, *235*, 107660.
23. Eigen, E.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2366–2374.
24. Ibraheem, A.; Wonka, P. High Quality Monocular Depth Estimation via Transfer Learning. *arXiv* **2018**, arXiv:1812.11941.
25. Bi, X.; Yang, S.; Tong, P. Moving Object Detection Based on Fusion of Depth Information and RGB Features. *Sensors* **2022**, *22*, 4702. [CrossRef]
26. Rebecq, H.; Gallego, G.; Mueggler, E.; Scaramuzza, D. EMVS: Event-Based Multi-View Stereo-3D Reconstruction with an Event Camera in Real-Time. *Int. J. Comput. Vis.* **2018**, *126*, 1394–1414. [CrossRef]
27. Kim, H.; Leutenegger, S.; Davison, A.J. Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 349–364.
28. 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3867–3876.
29. Wang, L.; Chae, Y.; Yoon, S.H.; Kim, T.K.; Yoon, K.J. Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 608–619.
30. Liu, Y.; Feng, L.; Liu, S.; Sun, M. Sensor Network Oriented Human Motion Segmentation with Motion Change Measurement. *IEEE Access* **2018**, *6*, 9281–9291. [CrossRef]
31. Li, D.; Zhong, W.; Deh, K.M.; Nguyen, T.D.; Prince, M.R.; Wang, Y.; Spincemaille, P. Discontinuity Preserving Liver MR Registration with Three-Dimensional Active Contour Motion Segmentation. *IEEE Trans. Biomed. Eng.* **2019**, *66*, 1884–1897. [CrossRef] [PubMed]
32. Mitrokhin, A.; Ye, C.; Fermller, C.; Aloimonos, Y.; Delbruck, T. EV-IMO: Motion Segmentation Dataset and Learning Pipeline for Event Cameras. In Proceedings of the IEEE/RSJ International Coference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 6105–6112.
33. Mishra, A.; Ghosh, R.; Principe, J.C.; Thakor, N.V.; Kukreja, S.L. A Saccade Based Framework for Real-Time Motion Segmentation Using Event Based Vision Sensors. *Front. Neurosci.* **2017**, *11*, 83. [CrossRef] [PubMed]
34. Zhou, Y.; Gallego, G.; Lu, X.; Liu, S.; Shen, S. Event-based Motion Segmentation with Spatio-Temporal Graph Cuts. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–13. [CrossRef]
35. Ieng, S.-H.; Carneiro, J.; Benosman, R.B. Event-based 3D Motion Flow Estimation using 4D Spatio Temporal Subspaces Properties. *Front. Neurosci.* **2017**, *10*, 596. [CrossRef]
36. Shiba, S.; Aoki, Y.; Gallego, G. Event Collapse in Contrast Maximization Frameworks. *Sensors* **2022**, *22*, 5190. [CrossRef]
37. Lagorce, X.; Orchard, G.; Gallupi, F.; Shi, B.E.; Benosman, R.B. HOTS: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1346–1359. [CrossRef]
38. Godard, C.; Aodha, O.M.; Firman, M.; Brostow, G. Digging into Self-Supervised Monocular Depth Estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 3827–3837.
39. Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; Navab, N. Deeper Depth Prediction with Fully Convolutional Residual Networks. In Proceedings of the International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 239–248.