Stereo Depth from Events Cameras: Concentrate and Focus on the Future

Yeongwoo Nam[1,2,*] Mohammad Mostafavi[3,*] Kuk-Jin Yoon[4] Jonghyun Choi[2,5,†]

[1]Saige Research [2]NAVER AI Lab. [3]Lunit [4]KAIST [5]Yonsei University

yw.nam@saigeresearch.ai, mostafavi@lunit.io, kjyoon@kaist.ac.kr, jc@yonsei.ac.kr

Abstract

Neuromorphic cameras or event cameras mimic human vision by reporting changes in the intensity in a scene, instead of reporting the whole scene at once in a form of an image frame as performed by conventional cameras. Events are streamed data that are often dense when either the scene changes or the camera moves rapidly. The rapid movement causes the events to be overridden or missed when creating a tensor for the machine to learn on. To alleviate the event missing or overriding issue, we propose to learn to concentrate on the dense events to produce a compact event representation with high details for depth estimation. Specifically, we learn a model with events from both past and future but infer only with past data with the predicted future. We initially estimate depth in an event-only setting but also propose to further incorporate images and events by a hierarchical event and intensity combination network for better depth estimation. By experiments in challenging real-world scenarios, we validate that our method outperforms prior arts even with low computational cost. Code is available at: https://github.com/yonseivnl/se-cff.

1. Introduction

A common practice to tackle design challenges is learning from nature. Mimicking natural strategies by copying their form, shape, process, or even ecosystem for specific applications is called biomimicry [2]. Stereo depth estimation mimics the human visual ability to understand depth from a pair of cameras. The computer vision community has shown significant interest in stereo vision, while it has remained a challenging task. The ill-posed nature of stereo depth estimation, shortcomings from RGB sensors (*e.g.*, low dynamic range, motion blur and *etc.*), and algorithmic limitations make stereo vision very challenging. Examples of imperfect sensing the scene include low dynamic range, blurry, or noisy images. Special cases that the algorithms cannot handle include repeating patterns, reflective (*i.e.*, shiny) objects, and low-texture areas [23, 27].

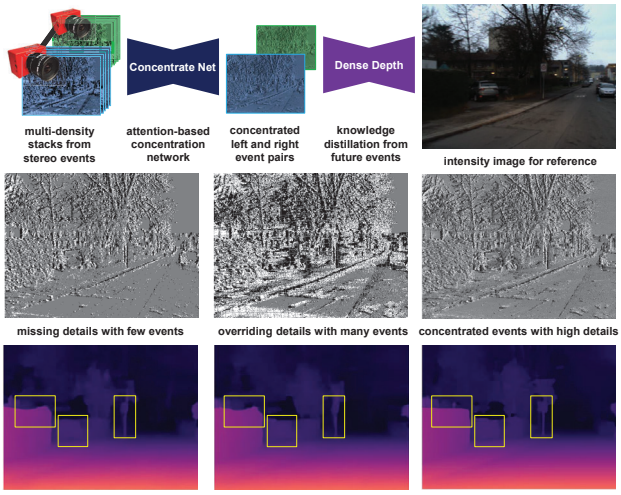

Figure 1. **Overview of our stereo depth estimation from events.** We predict dense depth with sharp edges by 'concentrating' the event representation tensors to preserve details. We further transfer the knowledge of future events by training with past and future events while keeping our system causal at inference.

Event cameras which are also referred to as neuromorphic cameras follow the same concept of biomimicry as they mimic human vision. Similar to the human eye, an event camera captures only pixel-wise intensity differences and reports them as a stream instead of the whole scene at once as a frame. Although event cameras bring new and unique specifications, they require a paradigm shift in the algorithms that use these devices. This paper falls within both mentioned trends coined as neuromorphic stereo vision and mimics the human eye to estimate depth from a stereo pair of event cameras. This line of research has gathered interest within the event camera community and has advanced in many aspects [23] including novel algorithms and attempts to generalize to real-world situations.

The stream of events is sparse in nature and does not follow any predefined pattern in terms of the density of events either in time or space, and relies purely on the scene and camera movements. We use this unpredictable sparse-dense stream and divide it into a sequence of stacked events that

*: equal contribution. †: corresponding author. This work is done while YN and JC are an intern, AI tech advisor at NAVER AI Lab., respectively.

each stack holds the most recent event location and their time information. The sequence is made with multiple numbers of events per stack, and we call it the *multi-density series of stacks*. As depicted in Fig 1, we feed this collection of event stacks with event slices that end at the GT timestamp to an *event concentration network*. As the name suggests, it concentrates all events into a clear edge-like tensor without any blur-like artifacts or omitting any details.

Event concentration helps create further details and sharp edges. It only depends on previous information, the events, thus it is a causal system. However, some details may also be omitted from the scene as we only use previously fired events. As a remedy, we further take into account the future events, but not directly as inputs. Specifically, we teach our network to distill from future events by feeding the past and future events at training time which may help the network to understand the scene contents and produce better predictions. Intuitively, we implicitly teach the network to contemplate or distillate from the knowledge of previously observed future events. Our experiments support this intuition by showing a significant increase in performance when training using the past and future events in comparison to training only with the past events.

Furthermore, unlike image frames that the conventional camera takes, events are sparse, which may lead to larger depth estimation errors in comparison to the depth from image frames. We supplement the sparse information by the combination of concentrating events and using the transferred knowledge from the future events. We evaluate our method on the challenging outdoor stereo events from the public benchmark dataset of DSEC [13], and use their metrics to compare with the state-of-the-art event stereo depth estimation methods. We present qualitative and quantitative comparisons to show how we outperform previous arts.

2. Preliminary: Event Cameras

Unlike traditional cameras, an event camera reports the scene as a stream of sparse and disconnected events, *i.e.*, per-pixel intensity changes larger than a predefined threshold. Each event is fired when it happens with very low latency, in the order of microseconds. The asynchronous nature of events brings the unique capability of being less adversely impacted by motion blur under rapid scene changes and camera movements but not completely immune to it [17]. Event cameras have higher dynamic ranges that reveals scene details that ordinary cameras may miss. We discuss more details in supplementary material for space sake.

3. Related Work

3.1. Stereo Depth Estimation on Images

Frame-based depth estimation is heavily studied by the computer vision community [36]. It traditionally involves stereo matching [42], optimized stereo matching using graph-cuts [22], and cost-volume filtering [16]. Recently, learning-based approaches improved accuracy drastically [4, 19, 26] which was also further improved by utilizing more 3D convolutions [5], deformable convolutions [18] and adaptive aggregations [45, 46].

3.2. Stereo Depth Estimation on Events

With the rise of event cameras, event-based stereo depth estimation emerged rapidly as the events already held timestamps and position details that may be efficiently utilized for synchronization and stereo matching [39]. However, imperfections such as real-world noise and different event threshold values among stereo pairs make the problem non-trivial [21, 34]. Such problems were addressed by utilizing orientation-sensitive filters [3], and cooperative regularization [10, 33]. Spiking neural networks were also the main study direction to address event-based stereo depth [1, 8, 31].

Other proposals include utilizing camera velocity for event synchronization [49], or estimating depth without explicit event matching [47]. Deep learning solutions considered combining a novel sequence embedding [43], or fusing depth and intensity images to cover the best from both worlds [27] to create highly detailed depth estimates. Both are capable of estimating dense depth from stereo events.

3.3. Event Alignment and Maximization

Events are reported as sparse points, thus adding additional appearance information about the scene may help reveal the underlying structure. In the varying sparse-dense stream, the events may not align easily over small time periods or counts of the number of events. Specifically in the presence of rapid camera trajectories in 6 degrees of freedom. An early attempt, considered estimating the 'lifetime' of events computed from their velocity on the image plane [30] for constructing sharp gradient images. Contrast maximization [12], warped events along motion trajectories, with which its parameters rely on the number of events in relation to a reference time. Its usages include motion, depth, and optical flow estimation applications. Contrast maximization is extended by analyzing possible rewards [41] and showing how a robust reward to noise and aperture uncertainty may be created. The objective functions [11] were also studied as 'focus loss' as they resemble the loss functions in shape-from-focus applications. Event segmentation [40] utilized an iterative clustering algorithm to distinguish between events fired from the camera movement and events created by moving objects in the scene.

These methods so far, require a good initialization to prevent bad local minimum convergence. The rotational motion estimation on event streams based on the branch-and-bound method is presented in [24], which aims for applications such as video stabilization and attitude estimation

without perfect initialization. A shallow convolutional sequence was utilized [28] for rectifying events with the aid of optical flow from events to keep more details in the event stacks and was used for events to reconstruct super-resolved images. As spatiotemporal registration further produces feature tracks, it can be utilized in visual odometry. A simple yet very fast visual odometry using graph-based optimizations was presented [25] for motion averaging and was verified by the motion of a high-precision robot arm. Unlike all of the aforementioned optimization-based methods that are usually limited due to their setting assumptions, we propose an event concentration method that produces a sharp edge-like tensor that holds scene details with high precision regardless of the scene or camera speed, movement directions, or degrees of freedom, and is aimed for real-world applications such as stereo depth estimation.

4. Approach

4.1. Event Preparation

We begin with the left and right rectified event sequences, E_L and E_R. Each event sequence $E = \{(x_i, y_i, t_i, p_i) \mid t_{i+1} > t_i, i = 1...N\}$ consists of N events sorted by time, where x and y present the pixel location, while t, and p present the timestamp, and polarity respectively. Given the left and right event sequences, our goal is to predict the disparity map D at time t_N.

We initially represent the event stream using the simple stacking methods based on the number of events (SBN), *i.e.*, by reversely counting the number of fired events from the depth timestamp until a pre-defined number, *e.g.*, $5,000$ events following [27–29, 44]. Although stacking based on time, *i.e.*, including all event in a short period of time, *e.g.*, 10 milliseconds can be also used, w e only use SBN throughout this paper. Although more complicated stacking methods exist [43, 48], we show that this simple representation is adequate to estimate depth with high accuracy, thanks to the attentive event usage by the 'concentration network' of our model. In our experiments, we use a single channel SBN stack unless otherwise stated. Following SBN, the single-channel tensor is initialized with intensity value 128. New incoming events per pixel location, update previous values. The value is updated to 256 when there is a positive event and set to 0 when there is a negative event.

4.2. Mixed-density Event Stacking

Event cameras generate different amounts of events depending on the movement of the camera or the objects in the scene. Faster movements create further events and *vice versa*. While stacking the events based on time or the number of events, if the pre-defined number of events or time period to include the event sequence or stacks is small (short stack), information on objects with low movement may be omitted. Conversely, if the number of events included in the event sequence is large (large stack), the excessive events from rapid-moving objects may overwrite previous events.

This is depicted in Fig. 2 for both the long and short stack cases, together with a temporally nearby intensity image for reference. This problem occurs more frequently in real-world situations, such as driving cars or flying drones, because scene objects are moving at different speeds while the camera may also be moving. In such situations, it is very

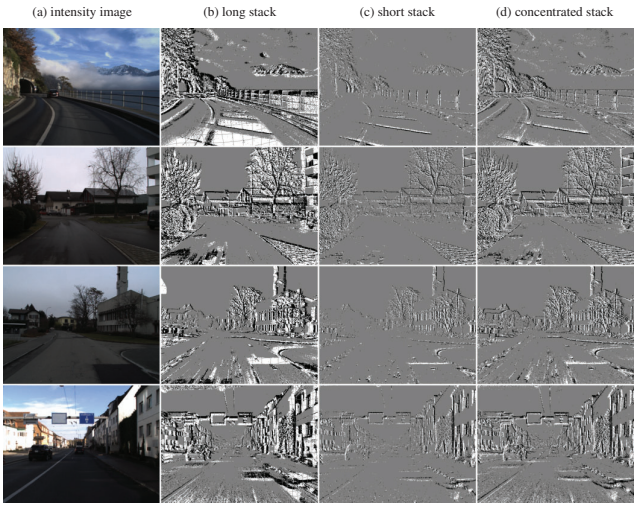

Figure 2. **Shortcomings of conventional event stacking method by a number of stacks (SBN)**. Representing the event stream by a predefined number of events or time intervals either overwrites previous events when the number of events is large, ((b) long stack) or omits scene details when the number of events is small ((c) short stack). (d) Our concentration network produces meaningful stacks (c) without overriding or missing events. The intensity image (a) is presented for reference.

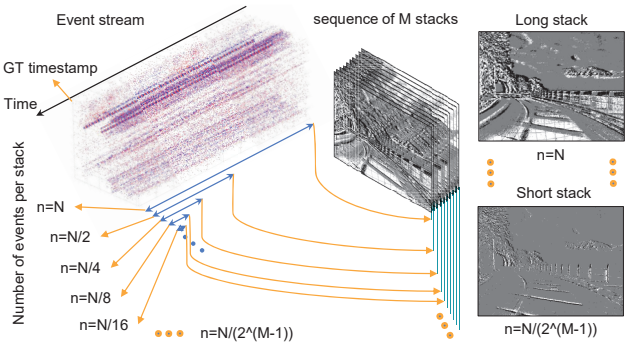

Figure 3. **Event sequence creation for the 'concentration network'.** We utilize a sequence of stacks created with different numbers of events per stack. Moving backward in time from the ground-truth (GT) depth timestamp, we first stack a very large number of events that include all movements from a long time span. For the next stack, we use half of the events used in the previous slice of the event stream and use the chunk that is connected to the GT location. We continue slicing the events in half until we reach the final stack M that we choose based on the number of stacks we want to include in our attention network.

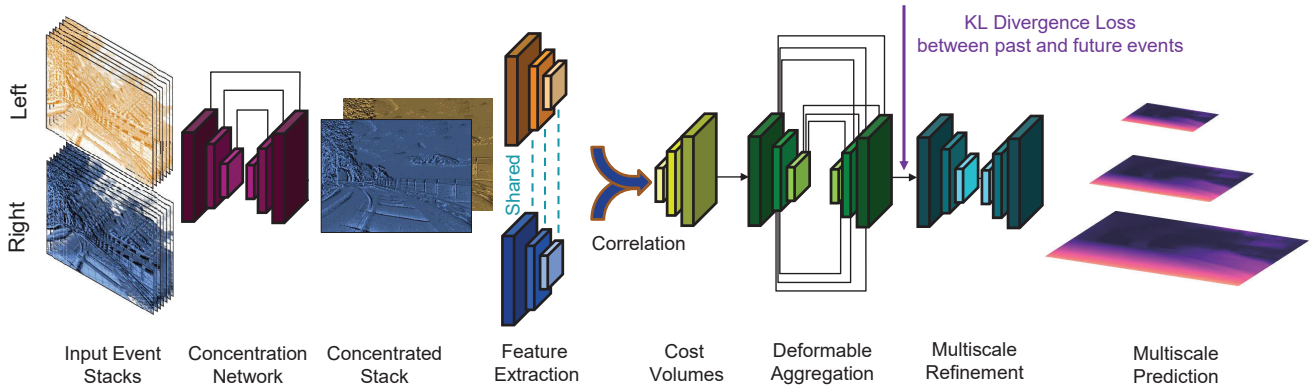

Figure 4. **Architecture overview**. We create a series of multi-density event stacks (Sec. 4.2) and use the concentration network (Sec. 5.1) to create the detailed event-based tensor called concentrated stack. We use the multiscale encoder to extract features (with shared weights) and correlate them to create cost volumes (Sec. 5.2). By 'deformable aggregation' and 'multiscale refinement' of the prediction we create our output dense depth estimation. Utilizing past and future events at training (Sec. 5.3) we can distill the knowledge from future events through the KL-divergence loss and reach higher quality depth estimations. The learning objectives are described in Sec. 5.4.

difficult to determine how many events should be included in the sequence of events which in return heavily distorts the quality of the downstream application, *i.e.*, depth estimation. As a remedy, we propose an event 'concentration' method that utilizes multiple event sequences with different event counts and learns to create an event tensor that is highly detailed without overriding previous details or missing the structural information.

As presented in Fig 3, we start by creating the first stack in our event sequence E_1 with an exaggeratedly large number of events $n{=}N$ to contain all possible event information necessary for stereo matching. The length of our event sequence has $M{=}10$ stacks and we set N to five million events for our experiments which linearly depends on the resolution of our event camera, *e.g.*, $640{\times}480$ for the dataset we utilized. We continue creating the event sequence with E_2, which ends at the same timestamp of E_1, however, with only half of the events in the previous event stack. We continue this, *i.e.*, stacking half of the events in the previous stack for the next stack with $E_3{-}E_M$, until reaching the final stack M that we choose; *i.e.* E_M has $n{=}N/2^{(M-1)}$ events. Note that n is rounded to the nearest integer to remain valid.We remove the first half when creating subsequent stacks since the first stack (E_1) that already has N events, has less relevant information from already moved objects and is far from the GT timestamp which in return may reduce the accuracy under rapid movements.

5. Network Design

For the depth estimation, we design an end-to-end neural network model depicted in Fig. 4. We first concentrate the sequence of events and transfer them to a highly detailed tensor (Sec. 5.1). Following the event concentration network, we introduce our depth estimation backbone design (Sec. 5.2). We present how to utilize past and future events at training in Sec. 5.3, to reach high-quality depth estimates from past-only events at inference as a causal system. Our learning objective is defined in Sec. 5.4, and we further show how to incorporate intensity images with events in our design in Sec. 5.5.

5.1. Event Concentration Network

Our concentration network handles mixed-density event stacks to create a detailed representation from events. The mixed-density event stacks contain a lot of detailed information, although transferring the event stream to stacks has shortcomings as described in Sec. 4.2. To reduce the negative effect of stacking, such as overriding previous events and also missing fine details, we design our event concentration network following the U-Net architecture [35] that can focus only on important information from M mixed-density event stacks $E_{1...M}$ using an attention mechanism.

We concatenate the mixed-density event stack in the channel dimension and use it as input to the event concentration network. This network receives image-like tensors $E_{1...M} \in \mathbb{R}^{H\times W\times M}$ as the input and outputs an attention score $z \in \mathbb{R}^{H\times W\times M}$, where H and W denote height, width of an image, respectively. The output of this network generates the weight $W \in \mathbb{R}^{H\times W\times M}$ that is utilized for assigning a weight to each event stack through a pixel-wise softmax operation formulated:

$$W(y,x,m) = \frac{e^{z(y,x,m)}}{\sum_{i=1}^{M} e^{z(y,x,i)}}, \tag{1}$$

where y and x denote pixel positions, and m is a layer index of mixed density event input. We then perform a weighted sum with the output weights W on the mixed-density event stacks $E_{1...M}$ to obtain our concentrated event stack tensor

$E_{con} \in \mathbb{R}^{H \times W}$ as:

$$E_{con}(y,x) = \sum_{i=1}^{M} W(y,x,i) \cdot E_i(y,x). \quad (2)$$

Our experiments show the effect of using an event concentration network by comparing the depth results from the concentrated event stack to the depth estimates from a randomly assigned low number of events and a high number of events per stack. Please refer to Sec. 6.3, Table 2 for quantitative analysis, and Fig 6 for qualitative comparisons, and also the supplementary material for further experiments.

5.2. Depth Estimation Network

Following well-performing stereo depth estimation networks [27], we design our model using some of their subnetworks. Note that we do not use their initial modules such as the event representation in [43] or the parts for merging event and intensity images in [27] and rather focus on the stereo matching modules. Our stereo matching network consists of four main modules as presented in Fig. 4, that are namely (1) the feature extraction module, (2) the cost volume module, (3) the deformable aggregation module, and (4) the multiscale refinement module. These modules are commonly used in stereo depth estimation networks, thus we follow the stereo matching design from [27], which in turn is also inspired from previous arts.

Specifically, we use ResNet for the feature extraction module [14] for its widely proved functionality and simplicity. We use feature pyramid networks [5] to recover details from each layer in a coarse to fine manner in multiple resolutions. Feature correlation [9] is used instead of concatenating features as the inner product convolves data from the left pair with data from the correct pair instead of convolving data with filters, resulting in a light-weighted network. We use deformable convolutions [6] on our cost volume for aggregation as they have non-fixed receptive fields which help matching sparse events better. To estimate accuracy depth at edges, we hierarchically upsample the predicted low-resolution disparity to higher intermediate scales by refinement [4]. More details can be found in [23, 27, 43].

5.3. Knowledge Transfer from Future Events

Even though we created the concentrated event stack as described in Sec. 5.1, problems such as occlusion, repetitive patterns, and incomplete sensing of the scene may still prevent from reconstructing high fidelity depth due to insufficient information from the past. Here, we propose to use the event information from the future to further enhance the quality of depth estimation. However, it is not viable at inference as the system is causal. As a remedy, we propose a novel scheme to predict the latent representation of future events by a loss function even if it receives past events only. Our empirical validations back up the validity of the approach by showing that we can better estimate the depth with the predicted future.

Specifically, we prepare two different stacking schemes as described in Sec. 5.1; an event stack focusing only on past events denoted as $E_{con,past}$ and an event stack focusing on both past and future events denoted as $E_{con,both}$. We incorporate a loss function in which the intermediate representation in the latent space of the network, *i.e.*, b_{past} and b_{both}, are enforced to be similar although the inputs are different, *i.e.*, using past-only $E_{con,past}$, and past with future $E_{con,both}$. For $E_{con,both}$, we use $2M$ mixed-density event stacks, M from the past up to the GT timestamp, and another M from the GT timestamp towards the future events. The $E_{con,both}$ has $2\times$ more events in comparison to $E_{con,past}$, and completely overlaps with $E_{con,past}$. To this end, we utilize the output of the deformable aggregation module, that holds the disparity probability for each pixel, and enforce the similarity loss. We use KL-Divergence between the two latent space representations for the similarity loss as:

$$\mathcal{L}_{sim}(b_{both}, b_{past}) = \sum b_{both} \log\left(\frac{b_{both}}{b_{past}}\right). \quad (3)$$

As we aim to enforce the intermediate latent space representation of b_{past} to be similar to b_{both}, thus, the gradient for similarity loss is only backpropagated through the past event stack path.

Regarding the choice of KL-Divergence, we first considered direct alignments (*e.g.*, $L2$, $L1$) but they may underperform as the location of future events may not land on corresponding GT depth edges due to the movement of objects in the scence in the future. Instead, we choose to align the future information to current 'softly' [38], computing the discrepancy between past-only and past and future events by the probability distance through the relative entropy (the well-known and widely used KL divergence for knowledge distillation [15, 38] and recent successful application in the event literature [7].

5.4. Learning Objectives

We use the smooth $L1$ loss between the ground truth disparity and the predicted disparity as our main objective term to train our network. Smooth $L1$ loss is widely used in image-based stereo matching because of its robustness in disparity discontinuities and low sensitivity to outliers or noise in comparison to $L2$ loss [5, 37, 45, 46]. The loss function is defined as:

$$\mathcal{L}_{sl1}(D, \hat{D}) = \frac{1}{V}\sum_{v=0}^{V} smooth_{L_1}(d_v - \hat{d}_v),$$
$$smooth_{L_1}(x) = \begin{cases} x^2, & \text{if } |x| < 1, \\ |x| - 0.5, & \text{otherwise,} \end{cases} \quad (4)$$

where D is the ground truth disparity, $\hat{D}$ is the disparity predicted by the model, and V is the number of valid pixels with ground truths for training.

By combining the loss of predicted disparity by using past event information only and using both past and future event information with the predicted future by the similarity loss (Eq. 3), we define the final loss as:

$$\mathcal{L} = \mathcal{L}_{sl1}(D, \hat{D}_{past}) + \mathcal{L}_{sl1}(D, \hat{D}_{both}) + \mathcal{L}_{sim}(b_{both}, b_{past}). \quad (5)$$

Although the loss of using both past and future implicitly include information about the past, the past only loss is necessary to take the input with only the past at inference.

5.5. Events and Intensity Image Fusion

In an early work of estimating stereo depth in a complementary setting by utilizing both events and intensity images [27], they unify the event stacks and intensity images in a 'recycling network', a recurrent module that iterates over events and images to reconstruct a blur-free image-like tensor that has high dynamic range properties of events. They utilize the intensity values from the ordinary camera when there are no scene changes as the events do not fire in that condition and shows better performance in comparison to the setting that uses only the event or intensity camera.

Following the intuition, we also report the performance of using images and events together with a simple method of fusing both sensors. Note that our method does not use a heavy sub-network to fuse the events and images. Instead, we use our concentrated event stack and the intensity image as inputs to two separate feature extractor modules.

Specifically, we concatenate the two feature maps by channel dimension, then further fuse the features of the two sensors by a 1×1 convolution. This fusing method is very simple yet effective. As shown in Table 1, the fusion performs better in all metrics. In Fig. 5, we qualitatively compare the output depth predictions of our method to the state-of-the-art event-intensity stereo depth estimation method [27]. Our depth predictions are either *on par* or slightly better than the prior arts. Furthermore, our method is computationally efficient as it does not use recurrent elements (see Table 1).

6. Experiments

We implement our network using the PyTorch [32] and initialize the network with random values and train from scratch end-to-end. We trained our model for 100 epochs with a batch size of 16. The maximum disparity is set to 192. We use Adam [20] with beta of (0.9, 0.999) for the optimizer, and weight decay to 1e-4. The learning rate starts at 5e-4 and decays with a cosine annealing. We use the DSEC dataset [13] for empirical validation. We describe the details of the dataset in the supplementary material.

Table 1. **Comparison of our method with state-of-the-art depth estimation methods on the DSEC dataset**. 'E': Events-only, 'E+I': Events plus intensity. Lower values are preferred (↓) in all metrics except FPS (frames per second). FPS is reported for two input resolutions '346×260 / 640×480.' Note: we obtain the FPS of [43] by authors' public code. The best is in **bold** and second best is in underline.

Method	Modality	MAE(↓)	1PE(↓)	2PE(↓)	RMSE(↓)	FPS (↑)
Baseline [43]	E	0.576	10.915	2.905	1.386	17.4 / 7.4
E-Stereo [27]	E	0.529	9.958	2.645	**1.222**	- / -
Ours on E	E	**0.519**	**9.583**	**2.620**	1.231	**23.2 / 11.3**
EI-Stereo [27]	E+I	0.396	5.814	1.055	0.905	10 / -
Ours on E+I	E+I	**0.364**	**4.844**	**0.840**	**0.818**	**18.2 / 9.3**

Evaluation Metrics. To quantitatively evaluate the quality of predicted disparity maps, following the standard metrics for the DSEC disparity benchmark, we use mean absolute error (MAE), root-mean-square disparity error (RMSE), and also the 1-pixel error (1PE) and 2-pixel error (2PE) that are the percentage of ground truth pixels with disparity error bigger than 1 and 2 respectively.

6.1. Quantitative Analysis

We first present the performance of our stereo depth estimation method using the DSEC disparity benchmark website in Table 1. As shown in the table, our method that only utilizes events (Ours on E) has much lower errors in comparison to the DSEC event-only baseline [43] in all metrics and to the state of the art [27] in MAE (the main metric), 1PE and 2PE but only worse in RMSE. When we utilize event and intensity images using our fusion scheme, ours (Ours on E+I) clearly outperforms the state-of-the-art method [27] by large margins.

Additional, our method is computationally much more efficient than the prior arts as we do not loop or iterate through recurrent modules; our method reaches more than 23 frames per seconds (FPS) using the 346×260 resolution event sensor in the event-only regime while the baseline [43] reaches 17.4 FPS. When we use events and images as input, our model reaches 18.2 FPS, while [27] only performs at 10 FPS. We compute FPS in Tab. 1 using a single NVIDIA 2080 Ti GPU, same as [27].

The pipeline of EI-Stereo [27] is arguably popular in literature [45, 46] but component determines the quality; we have a concentration network and (b) knowledge transfer from future. The benefits from event camera (*e.g.*, high dynamic range, negligible motion blur and low latency) are translated into better depth estimation by recovering missing details. As we focus on quality depth estimation, operational latency was beyond our scope; model compression with special neural engine hardware may help and are a great future research avenue. Note that despite the computational cost, ours is more than $2\times$ faster with better accuracy than the arts as presented in Table 1.

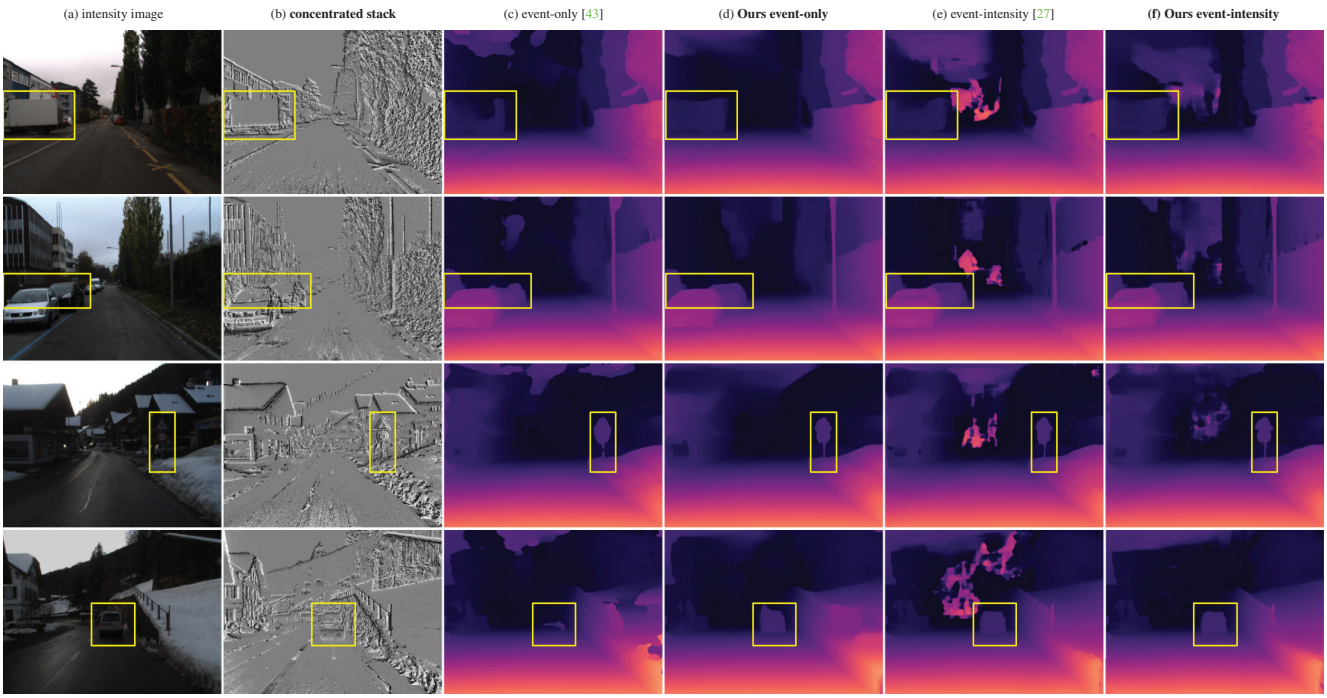

Figure 5. **Qualitative comparison on dense depth estimation**. We present our dense depth estimations using event-only (d) and events fused with intensity images (f) together with the (a) intensity image and (b) concentrated event stack for reference. We compare them to the (c) event-only [43] and (e) event-intensity [27] methods respectively. In the highlighted regions by yellow boxes, our method constructs fine details much better, *e.g.*, better clarity in cars (first, second, and fourth rows) and details of road sign with a post (third row), compared to its prior arts in both event-only and event-intensity modalities. Best viewed with the highlighted regions for detailed comparison.

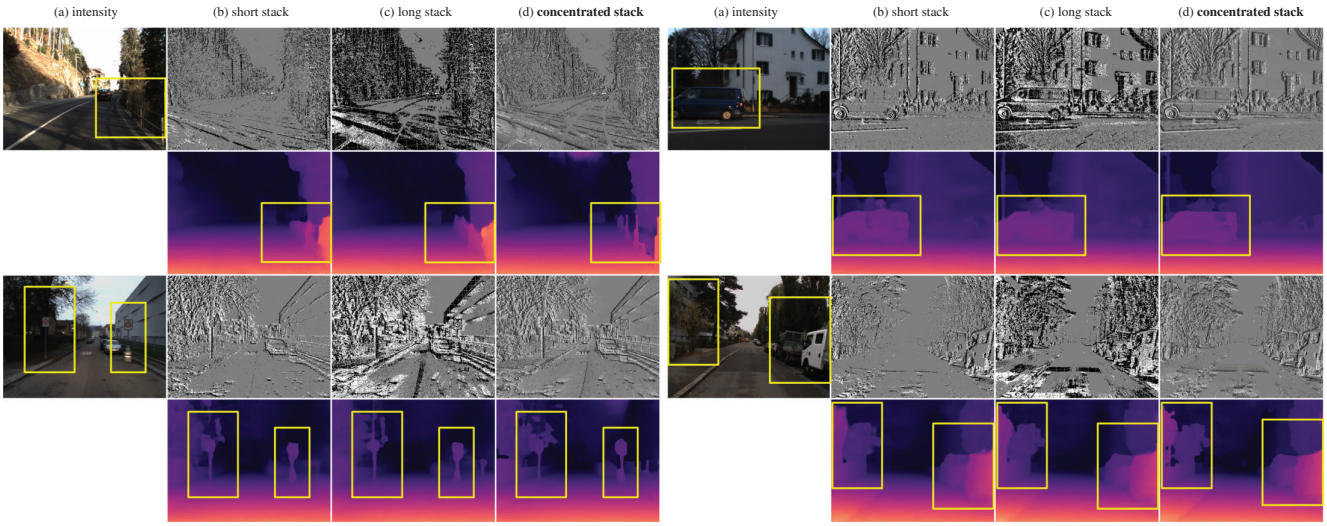

Figure 6. **Effect of the number of events on depth quality**. (a) intensity image, together with the depth predictions using a (b) short stack, (c) long stack, and also the (d) concentrated stack. The concentrated stack creates crisp clear boundaries and covers the scene details much better than using a fixed number of events in a stack, *i.e.*, short or long stacks.

6.2. Qualitative Analysis

We qualitatively compare our methods with prior arts in Fig. 5. Same as Table 1, we compare the results from [43] and [27]. We present randomly selected multiple scenes to showcase the performance. Our event-only method estimates more details in comparison to the event-only method [43] and our event-intensity method also predicts almost similar but with slightly sharper boundary details with less artifacts when compared to the event-intensity method [27].

Table 2. **Ablation on network components on depth**. Adding the mixed-density event stacking, event concentration network and distilling knowledge from future events all contribute meaningfully by reducing different error metrics.

Network	MAE(↓)	1PE(↓)	2PE(↓)	RMSE(↓)
Only stereo matching network	0.864	20.175	6.330	1.939
+ Mixed-density Event Stacking (MES)	0.852	19.182	6.070	1.923
+ MES + Concentration Network (CN)	0.831	18.875	5.757	1.880
+ MES + CN + Future Knowledge	**0.797**	**18.053**	**5.369**	**1.799**

Table 3. **Latent spaces to transfer future knowledge from**. We empirically investigate different possible latent spaces to transfer knowledge of future in training. Future knowledge at the deformable aggregation performs the best as argued in Sec. 5.3.

Knowledge Transfer at	MAE(↓)	1PE(↓)	2PE(↓)	RMSE(↓)
Feature extraction	0.810	18.177	5.436	1.853
Deformable aggregation	**0.797**	**18.053**	**5.369**	**1.799**
Multiscale refinement	0.833	19.143	5.880	1.867

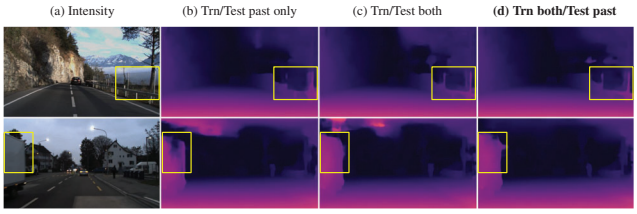

Figure 7. Qualitative results by KD from future. (both: past+future)

6.3. Further Analysis

Ablation Study. Starting from the base depth network presented in Sec. 5, with no bells and whistles, we add the proposed modules one by one. We present the results in Table 2. As presented in this table, all of the modules we proposed to lead to a performance improvement.

Latent Space To Transfer Future Knowledge From. Incorporating the future events by enforcing consistency between the past-only and past-future intermediate (latent) representations by the KL-Divergence loss improves our performance (Sec. 5.3). We argue that the location from which the future knowledge is transferred would be after the deformable convolutions. Unlike traditional convolutions, deformable convolutions learn dense spatial transformations with additional offsets by learning to expand (deform) to a 'larger receptive field' instead of fixed offsets [18]. It is likely that the future events would fall within the receptive field of past events by the deforming, transferring after the deformable aggregations would be beneficial.

We empirically verify this by experimenting with different candidate locations. As shown in Table 3, transferring the knowledge at the early stage of the network, *i.e.*, after feature extraction, exhibits the least error. We believe it is because the multiscale refinement harms the semantic of the representation space to reconstruct the detailed depths.

Impact of knowledge distillation (KD) beyond accuracy. In Fig. 7, we compare two scenarios: (1) only using past events (no KD) for train/test and (2) using past and future events (both) for train/test (non causal). Ours (Fig. 7d) estimates depths correctly at edges with less unwanted artifacts although we do not provide future events for inference.

7. Conclusion

We present a new stereo depth estimation network to estimate dense depth from stereo event cameras. Specifically, we propose to concentrate event stacks with multiple density events by an attention-based concentration network. The concentrated events shows scene details by missing less details without overriding events. We further propose to use future events in training for fine details without requiring the future at inference, but predict latent space representation of the future to maintain our system causal. Moreover, we show how to incorporate intensity images with events using a simple fusion scheme to reach higher quality depth estimates. Our method is computationally efficient, reaching more than 18 and 23 FPS for events plus images and events-only respectively, outperforming prior arts (10 and 17 FPS). We evaluate our method with challenging real-world dataset, DSEC, and show the usefulness of our method in both quantitative and qualitative analyses.

Limitations. Even with the proposed event concentration network, we still have to specify our minimum and maximum events by the size of the input image, though it is not as coarse or critical as the number of events. Although our method is computationally more efficient (18-23 FPS) than the prior arts (10 and 17 FPS), it is still far from practical. A promising research avenues include developing computationally efficient version of our method.

Potential Negative Societal Impact. Although the event camera is relatively less privacy-sensitive than conventional cameras as it largely ignores the textural details, it can still inadvertently capture unwanted private information from the human subjects, *e.g.*, the silhouette of humans on the road. Although we do not intend to allow such privacy hole, it does not have a mechanism to systematically prevent from doing so. Any privacy-preserving computer vision on event cameras is another promising research avenue.

Acknowledgement. This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2022R1A2C4002300 and No.2022R1A2B5B03002636) and Institute for Information & communications Technology Promotion (IITP) grants funded by the Korea government (MSIT) (No.2020-0-01361-003 and 2019-0-01842, Artificial Intelligence Graduate School Program (Yonsei University, GIST), and No.2021-0-02068 Artificial Intelligence Innovation Hub).

References

[1] Alexander Andreopoulos, Hirak J Kashyap, Tapan K Nayak, Arnon Amir, and Myron D Flickner. A low power, high throughput, fully event-based stereo system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7532–7542, 2018. 2

[2] Janine M Benyus. *Biomimicry: Innovation inspired by nature*. Morrow New York, 1997. 1

[3] Luis Alejandro Camunas-Mesa, Teresa Serrano-Gotarredona, Sio Hoi Ieng, Ryad Benjamin Benosman, and Bernabe Linares-Barranco. On the use of orientation filters for 3d reconstruction in event-driven stereo vision. *Frontiers in neuroscience*, 8:48, 2014. 2

[4] Rohan Chabra, Julian Straub, Christopher Sweeney, Richard Newcombe, and Henry Fuchs. Stereodrnet: Dilated residual stereonet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11786–11795, 2019. 2, 5

[5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 5

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5

[7] Yongjian Deng, Hao Chen, Huiying Chen, and Youfu Li. Learning from images: A distillation learning framework for event cameras. *IEEE Transactions on Image Processing*, 30:4919–4931, 2021. 5

[8] Georgi Dikov, Mohsen Firouzi, Florian Röhrbein, Jörg Conradt, and Christoph Richter. Spiking cooperative stereo-matching at 2 ms latency with neuromorphic hardware. In *Conference on Biomimetic and Biohybrid Systems*, pages 119–137. Springer, 2017. 2

[9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 5

[10] Mohsen Firouzi and Jörg Conradt. Asynchronous event-based cooperative stereo matching using neuromorphic silicon retinas. *Neural Processing Letters*, 43(2):311–326, 2016. 2

[11] Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza. Focus is all you need: Loss functions for event-based vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12280–12289, 2019. 2

[12] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. 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*, pages 3867–3876, 2018. 2

[13] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 2021. 2, 6

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[15] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 5

[16] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):504–511, 2012. 2

[17] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. V2e: From video frames to realistic dvs events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1312–1321, 2021. 2

[18] Dai Jifeng, Li Yi, He Kaiming, and Sun Jian. R-FCN: Object detection via region-based fully convolutional networks. In *Proceedings of the Neural Information Processing Systems Conference*, 2016. 2, 8

[19] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017. 2

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[21] Jurgen Kogler, Martin Humenberger, and Christoph Sulzbachner. Event-based stereo matching approaches for frameless address event stereo data. In *International Symposium on Visual Computing*, pages 674–685. Springer, 2011. 2

[22] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 508–515. IEEE, 2001. 2

[23] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 5

[24] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Globally optimal contrast maximisation for event-based motion estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6349–6358, 2020. 2

[25] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Spatiotemporal registration for event-based visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4937–4946, 2021. 3

[26] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity,

optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 2

[27] Mohammad Mostafavi, Kuk-Jin Yoon, and Jonghyun Choi. Event-intensity stereo: Estimating depth by the best of both worlds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4258–4267, 2021. 1, 2, 3, 5, 6, 7

[28] S. Mohammad Mostafavi I., Jonghyun Choi, and Kuk-Jin Yoon. Learning to super resolve intensity images from events. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2768–2786, June 2020. 3

[29] Sayed Mohammad Mostafaviisfahani, Yeongwoo Nam, Jonghyun Choi, and Kuk-Jin Yoon. E2sri: Learning to super-resolve intensity images from events. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1, 2021. 3

[30] Elias Mueggler, Christian Forster, Nathan Baumli, Guillermo Gallego, and Davide Scaramuzza. Lifetime estimation of events from dynamic vision sensors. In *2015 IEEE international conference on Robotics and Automation (ICRA)*, pages 4874–4881. IEEE, 2015. 2

[31] Marc Osswald, Sio-Hoi Ieng, Ryad Benosman, and Giacomo Indiveri. A spiking neural network model of 3d perception for event-based neuromorphic stereo vision systems. *Scientific reports*, 7(1):1–12, 2017. 2

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 6

[33] Ewa Piatkowska, Ahmed Belbachir, and Margrit Gelautz. Asynchronous stereo vision for event-driven dynamic stereo sensor using an adaptive cooperative approach. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 45–50, 2013. 2

[34] Paul Rogister, Ryad Benosman, Sio-Hoi Ieng, Patrick Lichtsteiner, and Tobi Delbruck. Asynchronous event-based binocular stereo matching. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2):347–353, 2011. 2

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[36] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002. 2

[37] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13906–13915, June 2021. 5

[38] Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5311–5320, 2021. 5

[39] Lea Steffen, Daniel Reichard, Jakob Weinland, Jacques Kaiser, Arne Roennau, and Rüdiger Dillmann. Neuromorphic stereo vision: A survey of bio-inspired sensors and algorithms. *Frontiers in neurorobotics*, 13:28, 2019. 2

[40] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-based motion segmentation by motion compensation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7244–7253, 2019. 2

[41] Timo Stoffregen and Lindsay Kleeman. Event cameras, contrast maximization and reward functions: an analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12300–12308, 2019. 2

[42] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, 25(7):787–800, 2003. 2

[43] Stepan Tulyakov, Francois Fleuret, Martin Kiefel, Peter Gehler, and Michael Hirsch. Learning an event sequence embedding for dense event-based deep stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1527–1537, 2019. 2, 3, 5, 6, 7

[44] Lin Wang, S. Mohammad Mostafavi I. , Yo-Sung Ho, and Kuk-Jin Yoon. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10081–10090, June 2019. 3

[45] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 5, 6

[46] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip H.S. Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 5, 6

[47] Yi Zhou, Guillermo Gallego, Henri Rebecq, Laurent Kneip, Hongdong Li, and Davide Scaramuzza. Semi-dense 3d reconstruction with a stereo event camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 235–251, 2018. 2

[48] Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. 3

[49] Alex Zihao Zhu, Yibo Chen, and Kostas Daniilidis. Real-time time synchronized event-based stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 433–447, 2018. 2