Event-Intensity Stereo: Estimating Depth by the Best of Both Worlds

S. Mohammad Mostafavi I.[§]
GIST, South Korea
mostafavi@gist.ac.kr

Kuk-Jin Yoon
KAIST, South Korea
kjyoon@kaist.ac.kr

Jonghyun Choi[†]
GIST, South Korea
jhc@gist.ac.kr

Abstract

Event cameras can report scene movements as an asynchronous stream of data called the events. Unlike traditional cameras, event cameras have very low latency (microseconds vs milliseconds) very high dynamic range (140 dB vs 60 dB), and low power consumption, as they report changes of a scene and not a complete frame. As they report per pixel feature-like events and not the whole intensity frame they are immune to motion blur. However, event cameras require movement between the scene and camera to fire events, i.e.*, they have no output when the scene is relatively static. Traditional cameras, however, report the whole frame of pixels at once in fixed intervals but have lower dynamic range and are prone to motion blur in case of rapid movements. We get the best from both worlds and use events and intensity images together in our complementary design and estimate dense disparity from this combination. The proposed end-to-end design combines events and images in a sequential manner and correlates them to estimate dense depth values. Our various experimental settings in real-world and simulated scenarios exploit the superiority of our method in predicting accurate depth values with fine details. We further extend our method to extreme cases of missing the left or right event or stereo pair and also investigate stereo depth estimation with inconsistent dynamic ranges or event thresholds on the left and right pairs.*

1. Introduction

Stereo depth estimation is inspired from the human binocular vision. Estimating the depth from two or more views is one of the long-persisting topics that is tackled in many ways [24]. Early-stage stereo depth estimation methods consider matching all the pixels in a pair of stereo images to estimate the underlying 3D geometry of the scene. The camera parameters and the stereo setup are mainly available through calibration, and the task is to triangulate the matched pairs to recover the disparity or depth [33]. Stereo matching is still challenging because of the ill-posed nature of the problem, occlusions, imperfect imaging settings, blurred or low dynamic range images, repetitive patterns, and texture-less regions [24]. Recent methods estimate depth using learning-based frameworks without relying on hand-crafted parameters, and can also estimate metric depth based on the prior knowledge of the network [4, 5, 20, 41], thanks to the modern GPUs, creative architectures, and public-available large scale datasets.

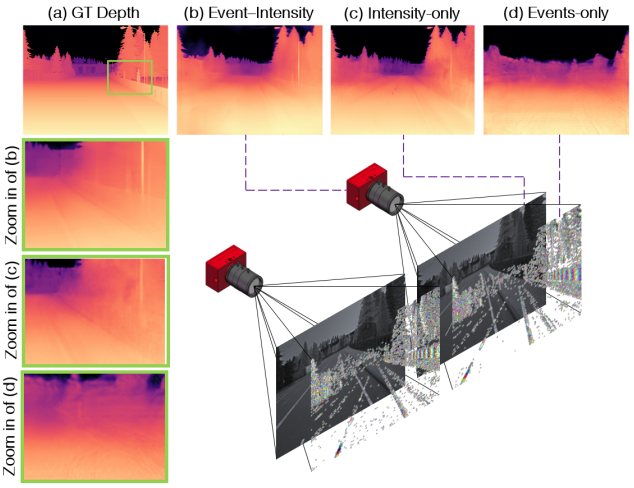

Figure 1. Estimating dense depth using our event-intensity stereo depth estimation framework. Our end-to-end network can estimate depth from the combination of Event-Intensity Stereo (b), Intensity-only stereo (c), or Event-only stereo (d) pairs. Using event-intensity stereo, we can reach higher quality depth in comparison to event-only or intensity-only inputs, as it can surpass the shortcomings of each source while gathering the best from them.

In spite of the significant progress, poor lighting conditions and complex materials properties are issues that have been less studied [24]. Infusing new sensing devices to enrich the input media is a direction worth researching. To this end we investigate in event-intensity cameras as a complementary source to enrich details captured from the scene.

Event cameras are new vision sensors that report changes of intensity individually per pixel, and asynchronous to other pixels, at the time of such changes. The output of an event camera, the events, are reported as a stream that varies based on the movement speed and direction of the

[§]: now at Lunit Inc. (lunit.io). [†]: corresponding author.

camera and the scene. Event cameras are intrinsically immune to motion blur, thus an ideal candidate for tasks involving rapid movements such as driving scenes. These cameras cover a much higher dynamic range (HDR) when compared to traditional intensity cameras, making them applicable for extreme lighting conditions.

Events are mainly fired at object edges, as intensity changes usually happen on edges, making event cameras an ideal tool for estimating sharp depth values on such boundaries. However, event cameras do not directly report intensity values, as they only sense the changes in intensity. Event cameras remain silent when the scene is static, *e.g.*, when we stop behind a traffic junction in which parts of the scene become invisible to an event camera. Other sample scenarios that contrast the pros and cons of using event or intensity cameras are presented in Fig. 2.

Considering the pros and cons of traditional and event cameras, we investigate in finding a way to go beyond the trade-off between events and images and utilize the benefits of these widely differing sensors at the same time. In our settings, we use a stereo pair of event-intensity sensors to estimate depth in different movement and lighting conditions. We propose a network that unifies events and intensity images through a recycling unit and applies deformable aggregations and multiscale refinements to estimate precise depth. Our system can work with all combination of available sensors so it is robust to failures of either modality.

To our knowledge, we are the first to investigate in the combination of events and intensity images to estimate dense stereo depth as in Fig. 1. We show the practical advantages of this combination in estimating depth by contrasting with event-only and image-only stereo depth estimation methods on synthetic and real-world datasets.

2. Preliminary

An event camera reports the scene as a tuple of: x and y locations, timestamp of the change in intensity (t) and sign of the change (σ) which indicates whether the sensed intensity is higher (positive event) or lower (negative event) than a predefined intensity threshold (τ). This stream of asynchronous events can reach near microseconds resolution of latency making it suitable for fast movement scenes.

As each pixel location of the event stream only holds the timestamp and sign information, an edge-like representation is produced when visualizing the event stream in short periods of time. Unlike ordinary cameras, event cameras report changes of a scene instead of the whole frame, thus, they require lower storage, bandwidth, and power. Recent event sensors report the event stream and intensity image, the active pixel sensor (APS), on a single device. They share a common grid-line of pixels for the events and intensity images, making them free from requiring further transformations when matching events to the image locations.

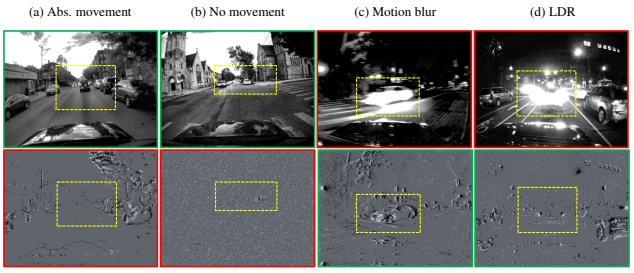

Figure 2. Expressing the pros (green) and cons (red) of intensity cameras (top) and event cameras (bot) on different scenes. Unlike the intensity camera, the event camera cannot capture scenes when absolute changes are zero such as two cars moving with the same speed ((a) Abs. movement), or a car stopped at a junction ((b) No movement). Unlike the event camera, the intensity camera creates blurred edges when the camera or objects are moving ((c) Motion blur), and has a lower dynamic range ((d) LDR). The data belong to the MVSEC dataset [49] as explained in sec. 5.1.

3. Related Work

3.1. Stereo depth estimation on images

Stereo depth estimation methods can be divided into two main classes: Traditional stereo-matching techniques, and learning-based methods.Traditional methods learn how to correspond pixels in the input images together and estimate per-pixel disparity, and usually contain modules for feature extraction, feature matching, cost aggregation, and depth estimation sequentially [33]. Stereo matching methods, either consider global objective optimization such as belief propagation [37], and graph-cuts [22]; or local correspondence such as the adaptive support-weight approach [42], and cost-volume filtering [16]. Further refinement modules [10] can be leveraged to refine the estimated depth using prior learned knowledge or from new incoming images.

Learning-based approaches perform stereo matching, cost aggregation, disparity computation and refinements in an end-to-end fashion without handcrafted parameters by 3D convolutions [25, 20, 4]. Although further accuracy can be reached by utilizing more 3D convolutions in a pyramid design [5], recent papers have focused on increasing the accuracy while consuming less memory. The less computationally expensive methods use deformable convolutions [19] as guided or adaptive aggregating layers [43, 41].

3.2. Stereo depth estimation on events

Stereo depth estimation emerged as one of the early applications of event cameras, as the fired events in each camera can be synchronized and matched in a short amount of time by their timestamps [35]. Early attempts utilized the low latency and power consumption of event cameras to perform fast and efficient stereo matching [21, 32]; where the matched events followed a triangulation stage in 3D to estimate the depth. However, imperfect timestamp synchro-

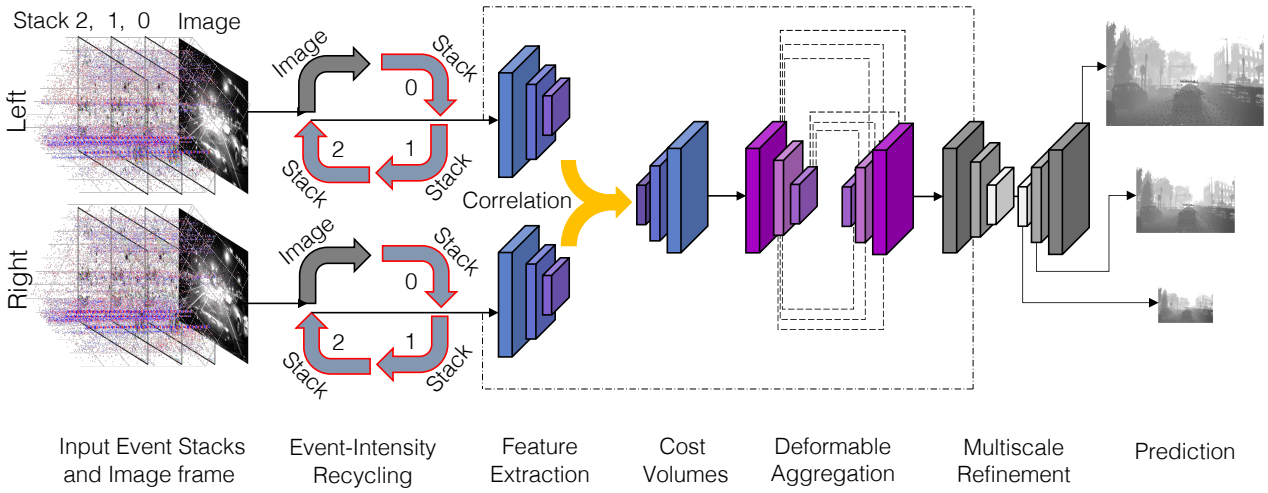

Figure 3. Architecture overview. The input to our network comes from the left and right stereo event-intensity cameras which are synchronized base on their timestamps. We use the three most recent event stacks preceding the intensity image together with the intensity image itself as the input to the event-intensity recycling network, that unifies events and images sequentially. Next, the features in each unified tensor is extracted in multiple pyramid scales. The left and right features are correlated by a multi-scale correlation layer to create the cost volumes. The cost volumes are given to the aggregation network which utilizes deformable convolutions to regress the disparity in multiple scales. Finally the intermediate predictions are refined scale-wise until reaching the original input image size using the unified tensor.

nization, real-world noise and different threshold sensitivity of the camera pairs caused ambiguous matching results.

Later methods improve the accuracy by incorporating orientation sensitive filters [3], cooperative regularization [29, 11], and spiking neural networks [28, 6, 1]. Semi-dense depth by incorporating the camera velocity for event synchronization was proposed in [48]. Depth estimation without explicit event matching was introduced in [46]. A new sequence embedding based on spatio-temporal aggregations was presented in [38]and was the first learning-based method to estimate dense depth from stereo events.

3.3. Joint Event-Intensity Applications

Recent event sensors report the event stream and intensity images within the same device. We use this already available source of images to fill in the gaps that the event camera alone cannot cover. Note that images from separate event-intensity stereo cameras, such as the DSEC dataset [12], can also be utilized after proper transformations. The event-intensity combination is studied previously in feature detection and tracking for visual SLAM [39], event to image reconstruction by high-pass filtering [34], super-resolution from a sequence of event stacks and intensity image [27, 17] and image enhancement by unifying the low dynamic range image with an event map [14].

4. Approach: Event-Intensity Stereo

4.1. Event preparation

We start from a pair of rectified tuples of intensity I_l,I_r and event stacks S_l,S_r from an event-intensity stereo camera pair with the size of $W \times H$. We first transform the event stream to a machine interpretable representation called event stack following [40, 27]. Although many other event representation techniques exists [47, 38], we follow this simple representation to show that our method produces good results even with a less complicated stacking method.

We stack the events preceding the intensity frame and use "stacking based on number" (SBN) [40] to make the event tensors as shown in the upper left region of Fig. 3. In this figure, the event stream is also visualized with the positive (red) and negative events (blue) and the intensity from APS is located at the end of this stream. The size of stacked event tensors are $W \times H \times C$ and are created in a sequence which ends at the timestamp of the APS frame. We use N=3, 000 events in each stack for the MVSEC data which is adjusted linearly to other camera sizes.

We stack by SBN for $C = 3$ channels, and set the initial tensor values to 128. By each incoming event, we update its landing location with 0 (negative events) or 256 (positive events), while newer events override previous events. We fill each channel by N/C events in this type of stacking, therefore it is important to choose N and C wisely to prevent overwriting many events.

4.2. Event-Intensity Recycling

Our event-intensity recycling network is inspired from the complementary intensity reconstruction methods explained in Sec. 3.3. We unify the event stacks and intensity images in a recycling network to complementary reconstruct a blur-free image-like output that has the has high dynamic range properties of events and sensed intensity values

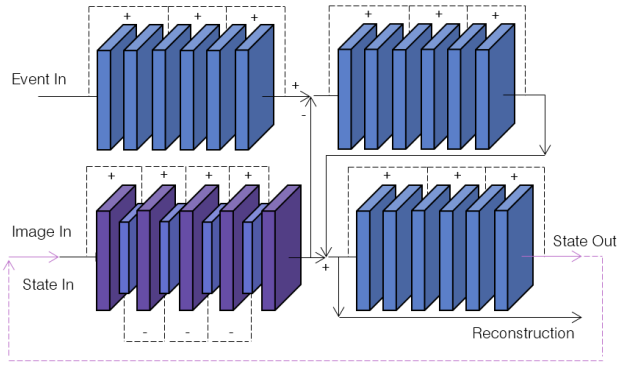

Figure 4. Event-intensity recycling network. We use an image (APS) frame and a sequence of its preceding event stacks. In the initial pass, an event stack is used ($Event\,In$) together with the intensity image ($Image\,In$) as the inputs. After that we recycle the inner state of the network ($State\,In$) and use the next stack ($Event\,In$) and continue in the same manner with the next stack and the updated state. The number of recycling cycles depends on the number of event stacks (*e.g.* 3).

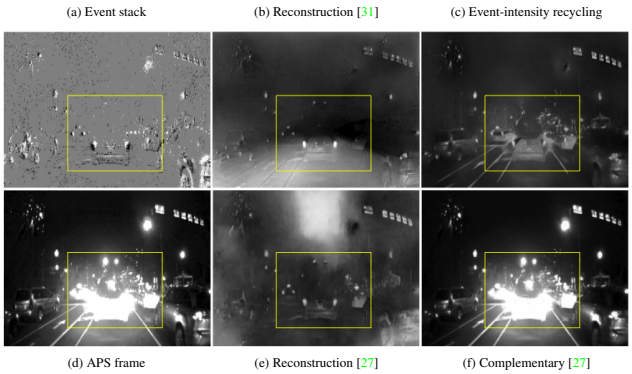

Figure 5. Comparing the structural details from event cameras, normal cameras and reconstruction methods from these inputs. Our event-intensity recycling method (c) combines the events and APS frames and shows more structural details in comparison to the (a) events (d) APS frame (b) event-only reconstruction in [31], resized event-only reconstruction in [27], (e) and also resized complementary reconstruction in [27] that mostly follows the APS.

from the ordinary camera when there is no scene changes.

Our event-intensity recycling network is demonstrated in details in Fig. 4, which is also presented abstractly as four sequential arrows in Fig. 3, that actually shows how data is provided, and how the hidden state is passed to the next stage. The four sequential arrows are the APS frame, followed by three stacks. In Fig. 4, we start from the APS frame (image), *e.g.*, for the left camera, I_{l0}, which goes to the image input $Image\,In$. At the same time, the most nearby stack to the APS frame that is synchronized by its timestamp, S_{l0} ($Stack\,0$), goes to the event input, $Event\,In$. This creates the first hidden state $State_{l0}$ at $State\,Out$. In the next step, the next stack ($Stack\,1$) S_{l1} goes to the the event input, $Event\,In$. At the same time, instead of any other image, the previous state $State_{l0}$ gets recycled and goes to the $State\,In$ (which was called $Image\,In$ in the previous step) and creates state $State_{l1}$ at $State\,Out$. In the last step, the final stack ($Stack\,2$) S_{l2} goes to the the event input, $Event\,In$, while the previous state $State_{l1}$ gets recycled and goes to the $State\,In$ and creates the unified event-intensity output $Reconstruction$.

The sequence order is not important, *e.g.*, we can first provide Stack 2, then the next stacks until reaching the APS frame, however the same order used in training should be used in inference as the network adapts itself to the sequence order. Please note that we do not aim for intensity reconstruction (Sec. 4.5), and rather unify the events and APS frames in a manner that can keep distinctive details from each camera to perform stereo matching. We call the unified outputs of our recycling network, U_l and U_r.

Our sequential design for unifying the event stacks and intensity images, the recycling network, is adopted from $e2sri$ [27, 17]. We unify the events-intensity information of the left and right pair separate to each other, but at the same time in separate threads. However, unlike e2sri, we utilize much less trainable parameters (almost ten times smaller than e2sri) for faster inference. Moreover, our event-intensity recycling is aimed for same-size reconstruction directly without any super-resolving components.

We do not aim for event only image reconstruction as events alone (top left) may miss static scene details. Furthermore, unlike the training scheme of e2sri, we do directly utilize clear intensity frames in training and further use blurred intensity images with altered dynamic ranges to dynamically unify the event and intensity images. We do this so that the network learns to reconstruct structural details from both events and images, even if such detail is completely missing, partially available or imperfect in one of the inputs.

We compare our results to e2sri visually in Fig. 5 and show event-intensity recycling method can create HDR details that the APS frame cannot capture. This sample is a night driving scene of a car when stopping at a junction. Unlike the event camera, the rear stop lights are not captured as the APS frame has limited dynamic range. However, the event camera also misses some parts as the car is stopping and overall scene-camera movement is not available to trigger the event camera. We visually compare to the event-to-image reconstruction method e2vid [31] and e2sri [27]. Both of the event-only reconstruction methods did not capture parts of the car and its surrounding area shown in the yellow region. The complementary method proposed in e2sri also creates white regions around the lights as it follows the APS frame in reconstruction. However, our event-intensity recycling method can unify details captured from the event camera and intensity camera adaptively and exhibits further scene details.

4.3. Deformable Aggregation

The feature extraction and cost volume creations are adopted from the well-designed depth estimation networks. Our feature extractor follows the Res-Net architecture [15], and we utilize feature pyramids [5] to create our cost volumes through feature correlation [7] instead of concatenation. Once we have our cost volumes, we can aggregate them using intra-scale and cross-scale aggregation modules, inspired from [41], for our three pyramid levels.

As explained in Sec. 2, event cameras mainly sense edges while missing other details, and intensity cameras sense multiple areas of the scene, while edge details may be missed as a result of lower dynamic range, occlusions or motion blur. Thus, we utilize deformable convolutions as they aim to go further than the fixed geometric structures of ordinary convolutional networks, and learn dense spatial transformations with additional offsets [19]. We utilizes deformable convolutions to adaptively aggregate the cost volumes using local and global aggregations[43]. We also utilize intra-scale aggregations [41], as higher-quality results can be obtained on object boundaries and thin structures.

Another fact that encourages us to utilize this design is that all edges are not sensed at once in an event camera, and it depends on the direction of movement between the event camera and the scene. Thus, discontinuities on the location of such edges may appear in the event stack. Downsampling the event stack can help connect this kind of gap on the edges. In stereo-depth estimation, cross-scale aggregation, aims for searching the correspondence in the downsampled images as low-texture or textureless regions will be more discriminative at a coarse scale [44].

4.4. Disparity Estimation

Our last stage utilizes stereo depth refinement modules [4] to upsample lower scales to higher intermediate scales and then to our final output scale in our framework. We use the left and right unified event-intensity images (U_l and U_r), that are presented in Fig. 3 by the dash-lines between our event-intensity recycling stage, and the refinement stage. We utilize soft $argmin$ disparity estimation [20], to regress the per-pixel disparity from the final cost volume.

4.5. Learning objectives

We train our fully supervised model end-to-end and start from a random initialization. As the unified event-intensity reconstruction quality affects our depth results, we train our network for the first few epochs using an "image reconstruction loss" to encourage the network to create high-quality image-like tensors as side results presented in Fig. 5. However, as our final intention is to estimate depth we do not actually need our network to reconstruct images, therefore, we stop utilizing the image reconstruction loss mid way and only use the end-point-error (EPE) as or main loss.

End-point-error. We use the $L1$ loss also known as the EPE [10], *i.e.*, the mean disparity error in pixels between the ground-truth (GT) disparity d_v and estimated disparity of our model $\hat{d}_v$ for pixel v among the V valid pixels of the depth. The $L1$ loss is more robust at disparity discontinuities and is less sensitive to outliers, in comparison to $L2$.

$$\mathcal{L}_{EPE}(d_v, \hat{d}_v) = \frac{1}{V}\sum_{v=0}^{V} |d_v - \hat{d}_v| \tag{1}$$

Image reconstruction loss. For intermediate image reconstruction (first few starting epochs), we utilize both the L_1 loss and the learned perceptual similarity loss (LPIPS) [45]. It has been shown thoroughly that the combination of these two loss functions can create artifact-free reconstructions with sharp structural details [27, 26]. For LPIPS, we use the AlexNet variant [23] following [27, 36, 17].

Our final loss ($\mathcal{L}$) is created by combining all the losses. Where E is the epoch number in which we stop using the image reconstruction loss and only continue training with the EPE, and λ_1 and λ_2 are the weighting factors.

$$\mathcal{L}= \begin{cases} \mathcal{L}_{EPE} + \lambda_1 \mathcal{L}_{LPIPS} + \lambda_2 \mathcal{L}_{L_1}, & epoch < E \\ \mathcal{L}_{EPE}, & epoch \geq E \end{cases} \tag{2}$$

5. Experiments and Analysis

5.1. Datasets

We use two main datasets divided based on their origin of being from real-world cameras or simulated cameras. Our source to real-world events are the multi-vehicle stereo event camera dataset or MVSEC [49] and the stereo event camera dataset for driving scenarios (DSEC) [12]. For simulated data we generated a new dataset named ECAR.

MVSEC has two DAVIS [2] cameras that are placed in a stereo setting which provide the image frames and event streams. The sensing devices are mounted on multiple vehicles in various daytime and night lighting scenes and covers both car movements and stopping scenarios. The GT depth information in this dataset similar to many other Lidar data, does not always align with the image or event readings as it does not cover depth higher or lower than a specific elevation. Moreover some depth values of previous moving cars sometimes remains in the current GT frame.

DSEC, a recent large-scale outdoor stereo event camera dataset, changes the assumption of accurate pixel correspondence and presents a new combination that events and images are from two different camera pairs with different resolutions and baselines (distance between the two cameras). The event camera itself also has a baseline with the intensity camera, but all cameras are on the same height. DSEC covers a larger variety of illumination conditions, its

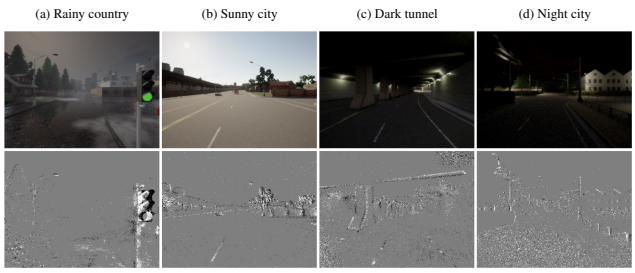

Figure 6. ECAR dataset. Intensity and events generated under various daytime, weather, city structures, and added noise conditions.

training splits are clear and the GT evaluation depth is withheld at their submission website that reports the evaluations. As the calibration parameters and rectified images are provided, we warp the images to the event locations.

ECAR is generated utilizing two open-source simulators, the CARLA simulator [8] in which we simulate many different driving scenarios and generate stereo intensity and depth pairs as videos, and the event camera simulator or ESIM [30] in which by giving a sequence of highly correlated videos event sequences are generated. This way we can generate stereo events and images together with the GT depth. The naming is the combination of ESIM and CARLA as ECAR. We used multiple lighting, weather, traffic, road and town settings in CARLA combined with different camera threshold settings in ESIM, as presented in Fig. 6. We cover most of the variations available in real-world cameras at simulation, to minimize the differences to real-world scenes as suggested in multiple places [31, 27, 36].

The ECAR dataset covers five large-size CARLA towns with almost $7,000$ pairs of APS images per town. We gradually changed the weather and daylight, so that each town includes all of the randomly chosen changes. The simulated car in ECAR, stopped when approaching other cars, pedestrians and stop lights, that prevented continuous event generation. Thus, we programmed the traffic lights to be green once the car holding the cameras reached a junction, and removed all other cars and pedestrians from the simulation. Although we utilized ESIM and CARLA separately, a recent plugin [18] includes event simulation in CARLA.

5.2. Experimental Setup

We initialize our network with random values, and train from scratch end-to-end. We set λ_1, λ_2 in Eq. 2 all to 1, and train our network for 64 epochs using 8 batches, in which the image reconstruction loss is used for the first 20 epochs for simulated data. The event-only stereo method (ES) and intensity-only stereo method (IS) share the same design with event-intensity stereo method (EIS) explained in Sec. 4. However, ES receives an extra event stack instead of the image frame of EIS, and IS receives extra intensity frames from a sequence of images instead of the event stacks in EIS. The extra event stack or intensity image is from the beginning of the sequence of stacks or images, thus all methods use past and present data *i.e*. they are causal.

Table 1. Performance evaluation using dense ground truth on the MVSEC [49] and ECAR dataset. Our event-intensity stereo (EIS) method estimates dense depth with higher quality in comparison to event-only stereo (ES), and intensity-only stereo (IS) depth estimation methods using the data split and protocols in [38].

	Mean depth error [cm]			One pixel error [%]		
Split	ES	IS	EIS	ES	IS	EIS
MVSEC Split **1**	**13.27**	14.12	13.74	80.6	71.7	**89.0**
MVSEC Split **2**	25.18	23.24	**18.43**	73.0	67.3	**85.2**
MVSEC Split **3**	25.72	23.78	**22.36**	68.3	53.8	**88.1**
ECAR	22.3	18.7	**11.8**	67.7	79.5	**81.7**

Table 2. Comparing our EIS and ES methods to the events-only state-of-the-art dense depth estimation method [38] using the DSEC [12] dataset. Both of our methods outperform the baseline, while EIS improved the baseline considerably and ranked first among all submissions of the competition hosted by the CVPR 2021 workshop on event-based vision [9] as described in Sec. 5.3.

	MAE	1PE	2PE	RMSE
Events-only Baseline [38]	0.576	10.915	2.905	1.386
Event Stereo ES	0.529	9.958	2.645	1.222
Event-Intensity Stereo EIS	**0.396**	**5.814**	**1.055**	**0.905**

5.3. Quantitative and Qualitative Analysis

We utilize the real-world and simulated datasets in Sec. 5.1 (MVSEC, DSEC and ECAR), for the qualitative and quantitative analysis. On the MVSEC dataset, we adopt the same training and validation protocols including the maximum disparity and data splits (MVSEC split 1-3) following the state-of-the-art dense stereo event depth estimation [38] to perform a comparison between the different combinations. We qualitatively compare the results using the mean depth error and one pixel error (1PE), the percentage of ground truth pixels with disparity error less than one pixel, both computed on the full dense GT disparity values.

We report the results that obtain the lowest end-point-error from the validation set in Table 1. Note that in this setting, our method reaches near 10 FPS inference time using an NVIDIA RTX 2080 Ti GPU, which can be faster with the trade-off of slightly reducing the performance by variants. Our EIS method, outperforms ES and IS on both mean depth error and 1PE on most of the data splits.

The ECAR dataset covers a variety of lighting conditions, however, split 1-3 [38] of the MVSEC dataset only covers the indoor flying scenes with acceptable scene illumination. As these sequences are captured without diverse dynamic range effects, IS usually has less error in comparison to ES. However in MVSEC split 1, ES showed lower MDE than EIS. We visually investigated that unlike its test set, the training sequences in split 1 have a lower flying height, and the flying movements in it are uniform

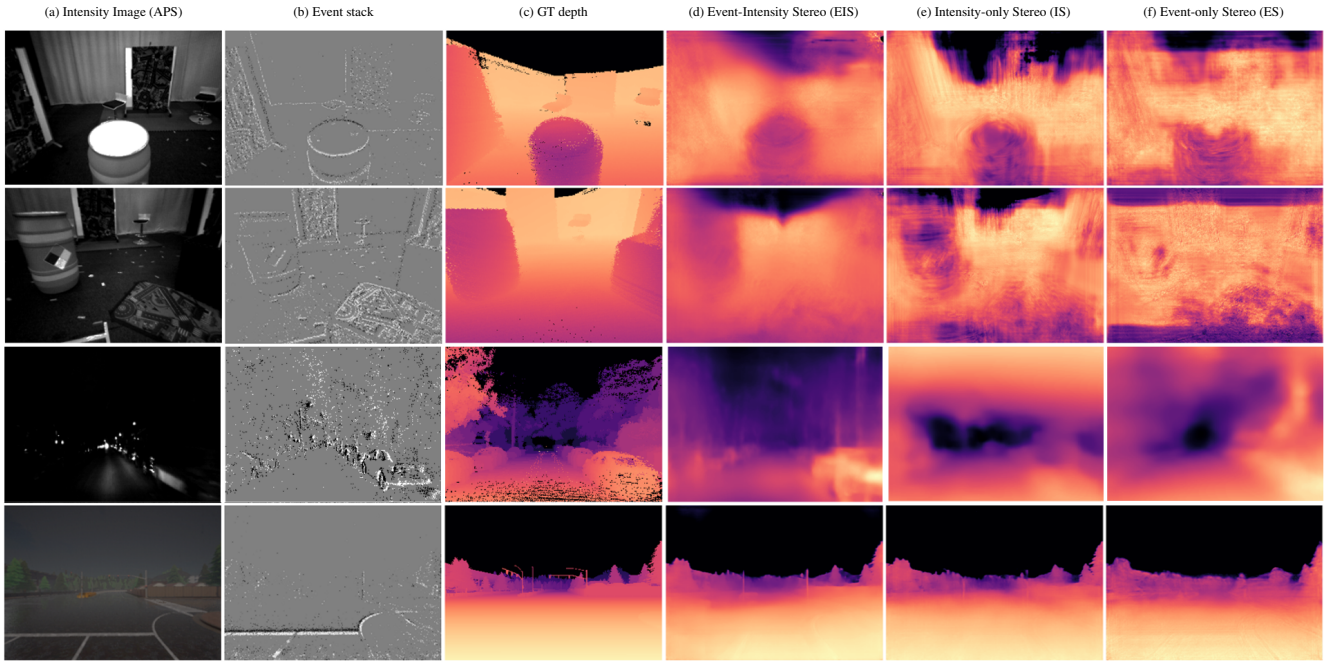

Figure 7. Quantitative comparison among different stereo methods based on their input sources. Our Event-Intensity Stereo (EIS) method (d), utilizes the Intensity (a) and event stacks (b) to estimate more accurate detailed depth in comparison to Intensity-only Stereo (IS) (e), or Event-only Stereo (ES) (f), methods. Data are from the real-world MVSEC [49] and simulated ECAR dataset (last row only).

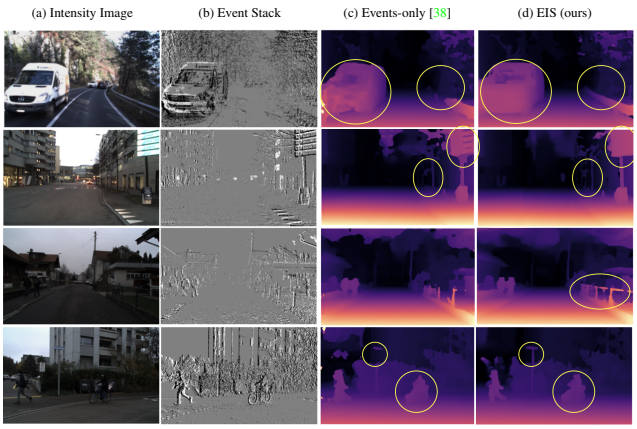

Figure 8. Qualitative comparison with the events-only baseline [38] from the DSEC [12] dataset. Although the GT is not publicly available, by referring to the intensity image and events, we can observe that our EI-Stereo creates more detailed depth values.

without sudden changes. Our network can't generalize well without exploring such samples at training. Furthermore, as split 1 is smaller, errors cannot be normalized over the split. Such disagreements don't exist when using large-scale and diverse datasets such as DSEC, thus comparisons to state-of-the-arts are generally fairer and more reliable on DSEC.

On the DSEC dataset presented in Table 2, we further report the two pixel error, Root mean square error (RMSE) and mean absolute error (MAE) of the disparity. Our EIS estimates stereo depth with much less errors on all metrics in comparison to the state-of-the-art events-only dense stereo depth estimation method [38], presented as the baseline of the DSEC challenge [12]. As the dataset is large-scale and evaluations are automatically generated without presenting the GT to the public, the comparisons on DSEC website are fair and reliable.

Our EIS method outperformed the baseline with a large gap, and also ranked first among all submissions, on all sequence, and in average, over all the metrics of the DSEC challenge at CVPR 2021 workshop on event-based vision [9]. As presented in Table 2, our events-only method ES also outperforms the baseline. Please refer to the DSEC challenge website [9] for more comparisons and detailed evaluations on individual sequences.

We qualitatively present our results and their input APS frames and event stacks for reference. For MVSEC and ECAR datasets we further show the GT depth for reference. In Fig. 7, Quantitatively, our EIS method shows further details in comparison to IS and ES which follows the results reported in Table 1. In Fig. 8, EIS shows reconstructions on parts that the baseline [38] misses when considering the events and APS to infer the GT as it is not publicly available. Note that the MVSEC results in Fig. 7 may look slightly smoothed. This gap performance comes form the limited allocated data in the MVSEC dataset splits, that prevents our network from generalizing. However, such gap does not exist in the DSEC results in Fig. 8. For the MVSEC

Table 3. Stereo depth with missing data. By training our method with missing modalities we can estimate depth even when the input event stack or intensity image is missing from a stereo pair.

Train	Test	MDE	1PE
Normal	Full modality	10.6	85.5
Normal	No left event stack	87.5	8.1
Normal	No left APS frame	91.7	11.3
Missing	Full modality	16.8	84.9
Missing	No left event stack	17.9	82.2
Missing	No left APS frame	20.3	76.5

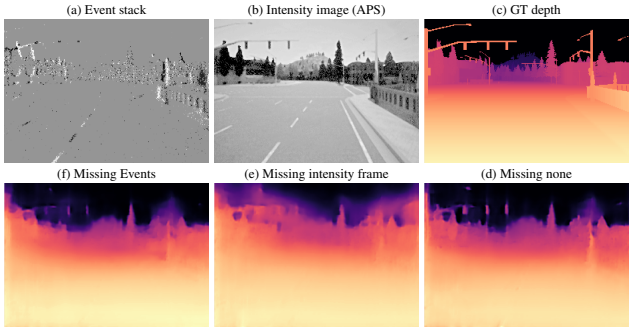

Figure 9. Stereo depth with missing data following Table 3.

night sequences and ECAR data, we follow the MVSEC flying data split trend of [38] and create our data split.

5.4. Ablation study

We ablate the different network components by removing each block from the full network (FN) design and evluate it using a subset of ECAR. In Table 4, each sub-network effectively improves the performance meaningfully.

Table 4. Ablating the effect network components on the depth.

Network	MDE	1PE
Full network (FN)	8.3	78.6
FN - {Feature pyramid net.}	37.8	64.1
FN - {Deformable aggregation net.}	23.4	70.3
FN - {Multi-scale refinement}	13.8	67.2

5.5. Extensions

Stereo Depth with Missing Data. Although we utilize four complementary event and intensity resources, in real-world applications, technical sensor fault or malfunctioning can prevent a source from reporting outputs and reduces the system reliability. As an example, in the outdoor_day sequence of MVSEC, one camera pair suddenly stops reporting the intensity frame, making intensity-only stereo impossible. As an extension we utilize the combination of an event-intensity cameras with a missing resource (a single event or intensity pair) and show how EIS recovers depth under a variety of faulty settings in Fig. 9 and Table 3.

Stereo Depth with Inconsistent Left-Right pairs. Color consistency is the basic assumption in stereo matching. However, there is no guarantee that the stereo pairs share the same event threshold (events), or the exact same dynamic range (intensity). We extend our method to the extreme case of inconsistent left and right intensity pairs with different dynamic ranges or different event thresholds, and show that our method can estimate acceptably accurate depth values in Fig. 10 and Table 5.

Table 5. Stereo depth estimation under inconsistent light settings. Event-intensity (EIS) depth estimation has less error under inconsistent light settings in comparison to intensity-only stereo (IS).

Train and Test	MDE	1PE	Train and Test	MDE	1PE
Consistent IS	17.3	79.5	Consistent EIS	10.6	85.5
Inconsistent IS	69.0	57.2	Inconsistent EIS	32.0	65.7

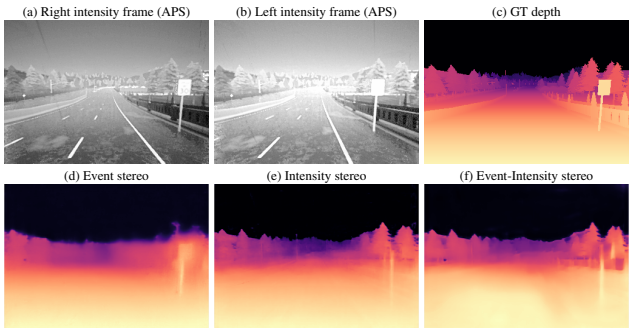

Figure 10. Inconsistent left-right stereo pairs following Table 5.

6. Conclusion

We present an end-to-end network capable of predicting dense depth from a stereo tuple of event and intensity cameras within a single framework. We unify events and images for stereo matching and perform deformable aggregations to exploit the benefits of our event-intensity stereo framework, and benchmark it to event-only and image-only solutions. We evaluate on real-world and simulated data and show the superiority of event-intensity stereo depth estimation. We further extend the reliability and robustness of our method to stereo depth estimation with missing data, and stereo depth estimation with inconsistent left-right pairs. Reaching a faster and fully asynchronous design using spiking neural networks [13], remains as our future direction.

Acknowledgement. This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2019R1C1C1009283), the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (NRF-2018R1A2B3008640) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST)), (No.2019-0-01351, Development of Ultra Low-Power Mobile Deep Learning Semiconductor With Compression/Decompression of Activation/Kernel Data, 20%) and (No. 2021-0-02068, Artificial Intelligence Innovation Hub). Authors thank Yeong-woo Nam for helping with the DCSEC challenge.

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. 3

[2] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 5

[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. 3

[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. 1, 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*, pages 5410–5418, 2018. 1, 2, 5

[6] 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. 3

[7] 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

[8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 6

[9] "DSEC competition 2021" https://dsec.ifi.uzh.ch/cvpr-2021-competition-results. 6, 7

[10] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 2, 5

[11] Mohsen Firouzi and Jörg Conradt. Asynchronous event-based cooperative stereo matching using neuromorphic silicon retinas. *Neural Processing Letters*, 43(2):311–326, 2016. 3

[12] Mathias Gehrig, Willem Aarents, Daniel Gehirg, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 2021. 3, 5, 6, 7

[13] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002. 8

[14] Jin Han, Chu Zhou, Peiqi Duan, Yehui Tang, Chang Xu, Chao Xu, Tiejun Huang, and Boxin Shi. Neuromorphic camera guided high dynamic range imaging. In *IEEE CVPR*, pages 1730–1739, 2020. 3

[15] 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

[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] S. Mohammad Mostafavi I., Yeongwoo Nam, Jonghyun Choi, and Kuk-Jin Yoon. E2sri: Learning to super-resolve intensity images from events. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3, 4, 5

[18] Daniel Gehrig Javier Hidalgo-Carrio and Davide Scaramuzza. Learning monocular dense depth from events. *IEEE International Conference on 3D Vision.(3DV)*, 2020. 6

[19] 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, 5

[20] 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. 1, 2, 5

[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] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 5

[24] 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

[25] 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

[26] Mohammad Mostafavi, Lin Wang, and Kuk-Jin Yoon. Learning to reconstruct hdr images from events, with applications to depth and flow prediction. *International Journal of Computer Vision*, pages 1–21, 2021. 5

[27] 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, 4, 5, 6

[28] 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. 3

[29] 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. 3

[30] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on Robot Learning*, pages 969–982, 2018. 6

[31] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE T-PAMI*, 2019. 4, 6

[32] 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

[33] 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. 1, 2

[34] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conference on Computer Vision*, pages 308–324. Springer, 2018. 3

[35] 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

[36] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the sim-to-real gap for event cameras. *Eur. Conf. Comput. Vis.*, Aug. 2020. 5, 6

[37] 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

[38] 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. 3, 6, 7, 8

[39] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001, 2018. 3

[40] 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

[41] 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*, pages 1959–1968, 2020. 1, 2, 5

[42] Kuk-Jin Yoon and In So Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656, 2006. 2

[43] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019. 2, 5

[44] Kang Zhang, Yuqiang Fang, Dongbo Min, Lifeng Sun, Shiqiang Yang, Shuicheng Yan, and Qi Tian. Cross-scale cost aggregation for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1590–1597, 2014. 5

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE CVPR*, pages 586–595, 2018. 5

[46] 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. 3

[47] 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

[48] 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. 3

[49] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE RA-L*, 3(3):2032–2039, 2018. 2, 5, 6, 7