Event-image Fusion Stereo using Cross-modality Feature Propagation

Hoonhee Cho and Kuk-Jin Yoon

Visual Intelligence Lab., KAIST, Daejeon, South Korea
{gnsgnsgml, kjyoon}@kaist.ac.kr

Abstract

Event cameras asynchronously output the polarity values of pixel-level log intensity alterations. They are robust against motion blur and can be adopted in challenging light conditions. Owing to these advantages, event cameras have been employed in various vision tasks such as depth estimation, visual odometry, and object detection. In particular, event cameras are effective in stereo depth estimation to find correspondence points between two cameras under challenging illumination conditions and/or fast motion. However, because event cameras provide spatially sparse event stream data, it is difficult to obtain a dense disparity map. Although it is possible to estimate disparity from event data at the edge of a structure where intensity changes are likely to occur, estimating the disparity in a region where event occurs rarely is challenging. In this study, we propose a deep network that combines the features of an image with the features of an event to generate a dense disparity map. The proposed network uses images to obtain spatially dense features that are lacking in events. In addition, we propose a spatial multi-scale correlation between two fused feature maps for an accurate disparity map. To validate our method, we conducted experiments using synthetic and real-world datasets.

Introduction

Stereo matching is the problem of determining correspondence points across two different images. In the rectified stereo camera setup with known parameters, the objective is to determine the horizontal pixel displacement between the left and right images, called disparity. The depth can be calculated using the parameters between the two cameras and the disparity. Therefore, stereo matching is important in 3D structure reconstruction (Yao et al. 2018) and autonomous driving cars (Yang et al. 2019).

Stereo matching conventionally employs frame-based images with RGB channels. Most of recent methods adopt CNN-based deep learning (Zhang et al. 2020, 2019; Kendall et al. 2017; Guo et al. 2019; Chang and Chen 2018; Yang et al. 2018; Tulyakov, Ivanov, and Fleuret 2018) and show reasonable performance. Their primary objective is to effectively extract features from input images, use context features for stereo, and guide the model to learn the correlation between the two features generated from input images. They adopt various embedding, matching, and regularization modules for the stereo model to solve such problems. Although such frame-based stereo has achieved successful performance, the effects of motion blur on estimation and the difficulty of operations in challenging light scenes remain.

Compared with conventional frame-based cameras, event-based cameras provide information on the amount of temporal change in each pixel value. Data with such information are called events. Events are asynchronous stream data with the information about the spatial position, polarity, and timestamps of intensity changes. Some event cameras (e.g., DAVIS346 (Brandli et al. 2014)) also provide aligned active pixel sensor (APS) intensity images. Event cameras provide event data with low latency (without motion blur) and have the advantage of a high dynamic range, which enables operation under extreme lighting conditions, and therefore event cameras can be more suitable for real-world applications.

Recent stereo matching studies with two event cameras (Tulyakov et al. 2019; Ahmed et al. 2021) adopt CNN structures as frame-based stereo matching methods. They embed event data from a stream format to a queue format, considering both spatial and temporal coordinates. The event sequence that has passed through temporal aggregation becomes an event image of size $c \times h \times w$, thus facilitating standard 2D convolution. They effectively estimated the disparity by matching the correspondence points between the two event cameras. However, obtaining a dense disparity map solely with sparse event data as input is an ill-posed problem.

In this study, we propose a novel end-to-end deep stereo architecture to generate a dense disparity map by combining the event features with the image features. Our method adopts both the image and event streams from the event camera as inputs. The proposed method creates a dense disparity map by effectively aggregating the two types of features using the proposed feature fusion module.

The main contributions of our work can be summarized as follows:

- We propose the novel end-to-end architecture of deep dense stereo, combining the event data and image. To fuse the two input data with different modalities into one feature, we propose a feature fusion module.
- We propose a method for generating a correlation volume

considering multi-scale features via spatial correlation. The multi-scale design considers the correlation between features from coarse to fine features.

- We propose a novel branch that uses events to obtain edge-related features from sparse ground-truth disparity. This branch is only adopted for training to reduce the memory footprint of the inference.
- We provide a new synthetic dataset for the event-image fusion stereo. Since no actual event data aligned with RGB images for stereo exist, previous studies of event stereo have solely compared with frame-based stereo using intensity images. In this work, for comparison with stereo method using RGB images, we generated synthetic data.

Related Works

Frame-based Stereo Depth Estimation

The most successful methods of early studies using conventional RGB images have adopted end-to-end deep learning networks (Kendall et al. 2017; Zhong, Dai, and Li 2017; Xu and Zhang 2020; Liang et al. 2018; Zhang et al. 2020, 2019; Guo et al. 2019; Chang and Chen 2018; Yang et al. 2018; Tulyakov, Ivanov, and Fleuret 2018). The networks generally comprise embedding, matching, and regularization modules. The embedding modules applied as shared weights for the left and right images are designed to obtain context features that are difficult to obtain at a pixel-level intensity from the image. In the matching modules, a substantially simple method exists for concatenating the shifted right features of each disparity value for the left feature (Zhong, Dai, and Li 2017; Chang and Chen 2018; Kendall et al. 2017). Alternatively, the matching module obtains a correlation by mapping the right features corresponding to each disparity of the left feature (Xu and Zhang 2020; Liang et al. 2018; Zhang et al. 2020, 2019; Guo et al. 2019; Yang et al. 2018). Regularization modules use 2D or 3D convolution for the disparity regression of the cost volume or correlation volume. However, the effects of motion blur and lightning on depth estimation remains a problem.

Event-based Stereo Depth Estimation

Event-based stereo matching methods can be divided into two groups: those that perform stereo-based depth estimation using the hand-crafted method representing events with image-based representation (Kogler, Humenberger, and Sulzbachner 2011; Camunas-Mesa et al. 2014; Zou et al. 2016, 2017; Piatkowska, Belbachir, and Gelautz 2013; Rogister et al. 2011; Zhu, Chen, and Daniilidis 2018; Piatkowska et al. 2017; Rebecq et al. 2017; Cho, Jeong, and Yoon 2021), and those that perform stereo matching using learning-based methods via a queue-based representation (Tulyakov et al. 2019; Ahmed et al. 2021). Early handcrafted methods used filter-based or window-based techniques to determine corresponding events (Camunas-Mesa et al. 2014; Zou et al. 2016, 2017). (Piatkowska, Belbachir, and Gelautz 2013) adopted the heuristic cooperative regularization by defining the spatio-temporal neighborhood for each event. Others predicted the depth using multi-view stereo with the known pose of the event camera (Rebecq et al. 2017; Cho, Jeong, and Yoon 2021). They succeeded in generating a depth map using a spatial-temporal sparse event camera; however, it was not dense, and the performance was inferior to that of the learning-based method. (Tulyakov et al. 2019) proposed a novel embedding of a 4D queue containing both temporal and spatial information of event data for deep learning. The queue that has undergone temporal aggregation becomes a 3D vector such as an image, thus facilitating the application of 2D convolution. Supervised learning of queue-based events accumulated for a certain period can create a dense depth map. (Ahmed et al. 2021) improved the performance of deep event stereo by employing the event features used in reconstructing the images. However, the event-only approach has domain-specific (*e.g.,* detailed textures) problems.

In this study, we propose a deep dense stereo matching method using both events and images. To the best of our knowledge, this is one of the first attempts to combine events with images for stereo matching. Our method uses the event for the edge of the structure, which contains powerful information and determines match points in the space of the less texture by using the information of the image.

Proposed Methods

The proposed model comprises six sub-networks: an event embedding network, feature extractor, fusion module, multi-scale correlation, 3D aggregation, and branch for sparse disparity. As illustrated in Fig. 1, as an input to the model, it takes images and events from the event camera in a stereo setup. The event embedding network applied to the event stream creates an event descriptor via 2D convolution. Event and image features generated by the feature extractor are then aggregated using the fusion module. Features fused at various scales become 4D cost volumes via multi-scale correlation. Then, a dense disparity map is extracted by a 3D aggregation network comprising 3D convolution. During training, a sparse branch comprising multi-scale correlation and 3D aggregation predicts sparse disparity solely for the locations where the events occurred recently.

Event Embedding Network

We follow the event sequence embedding (Tulyakov et al. 2019) to represent both the spatial and temporal information of an event in an image-like form. To apply temporal aggregation methods, event sequence embedding contains the first-in first-out (FIFO) queue structure that efficiently accumulates events. The accumulated event queue is a 4D tensor of size $H \times W \times \kappa \times 2$, where κ denotes the queue capacity. In this study, we adopted $\kappa = 7$, which exhibited the best results in (Tulyakov et al. 2019). After the kernel network with a continuous fully connected layer, the event queue becomes an event image with a size of $H \times W \times 32$.

Feature Extractor

The intensity and event images have sizes of $H \times W \times 1$ (size of $H \times W \times 3$ for RGB image) and $H \times W \times 32$, respectively. For the feature extractor, we adopt a ResNet-like

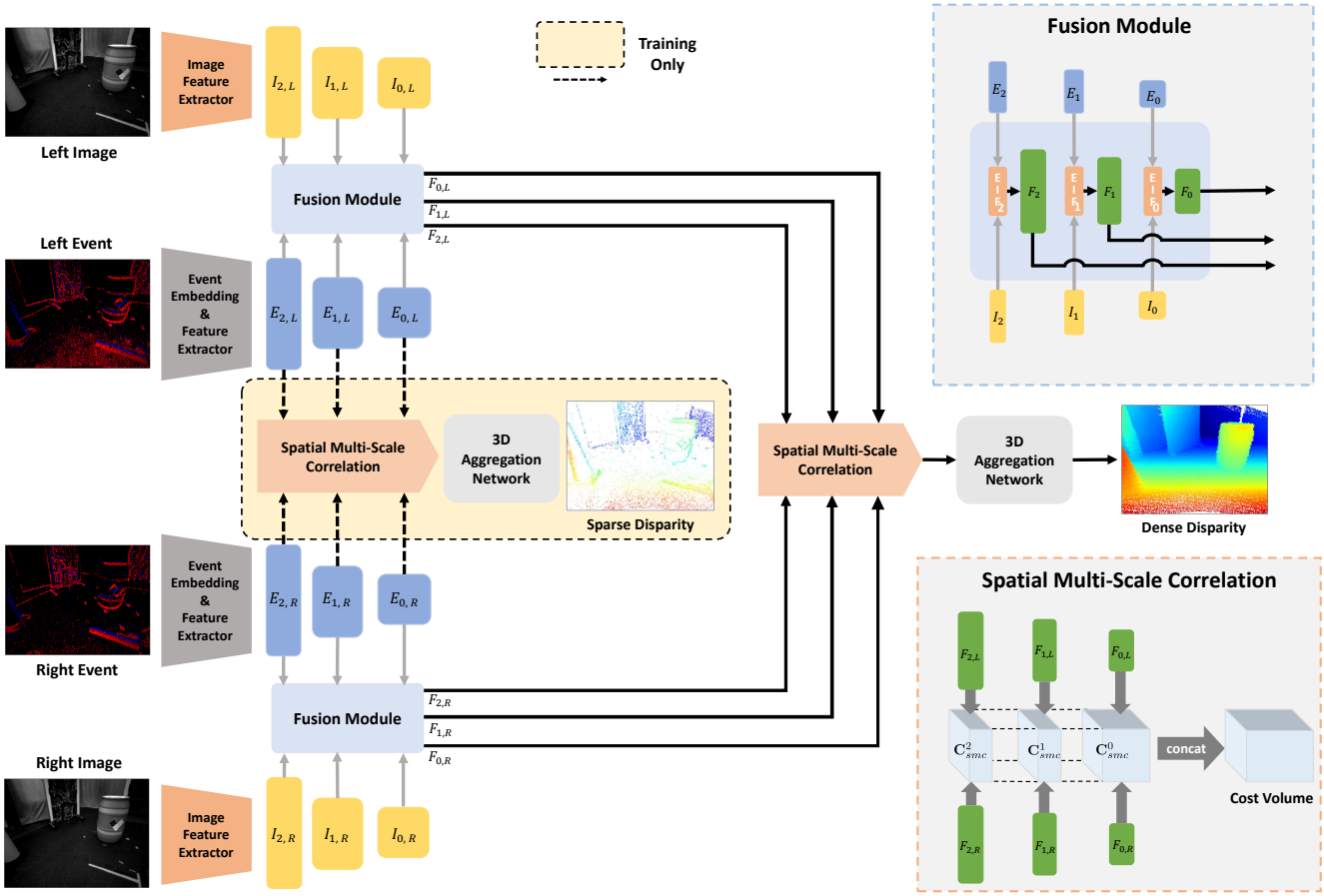

Figure 1: Overall framework of the proposed method. The proposed network employs both images and events from a stereo event camera as inputs. The event stream data are passed through an embedding module to enable the use of a CNN. The image feature extractor shares weights between the image feature extractor, which is also true for event feature extractors. The features of various scales are passed through the fusion and correlation modules to generate a cost volume. $F_{2,L}, F_{1,L}, F_{0,L}$ represent the multi-scale features of the left output of the fusion module, and $F_{2,R}, F_{1,R}, F_{0,R}$ represent the multi-scale features of the right output of the fusion module. The cost volume generated from the spatial multi-scale correlation is passed through a 3D aggregation network for dense disparity extraction. The dotted line is solely used to train the model, and it creates a sparse disparity using the event features alone.

network with half dilation used in (Guo et al. 2019) to increase the receptive field. Feature extractors of images and events have the same structure, except for the first convolution layer, owing to the different input channels in events and images. The left event and left image feature extractors share weights with the right event and right image feature extractors, respectively. The feature extractor generates three feature maps of different sizes as outputs in sizes of $H \times W \times 32$, $H/2 \times W/2 \times 64$, and $H/4 \times W/4 \times 128$.

Fusion Module

Instead of directly concatenating the event feature with the image feature, we fuse the transfer between event and image features with different modalities. Inspired by the *pose-attentional transfer network* (PATN) (Zhu et al. 2019), which comprises several cascaded pose-attentional transfer blocks, we adopt the *event with image-attentional transfer* (EIAT). As illustrated in Fig. 1, the fusion module takes multiple scales of paired event and image features. Paired features of different sizes become fused features via different *events from the image fusion* (EIF) modules. Let E_2, E_1, E_0, and I_2, I_1, and I_0 be the output feature maps of the event and image feature extractors in the order of larger spatial size, then the EIF module is as follows:

$$F_m = \mathcal{S}_m(E_m, I_m; \Theta_m), \ m = 0, 1, 2, \qquad (1)$$

where $\mathcal{S}_m$ and Θ_m denote the proposed EIF module and learnable parameters of the EIF module, respectively.

Each EIF module comprises several EIAT blocks and EIAT downs. As illustrated in Fig. 2, EIAT blocks comprise the event and image pathways. Image codes are generated from the image features using two convolution layers, two instance normalization layers (Ulyanov, Vedaldi, and Lempitsky 2016), and a ReLU layer (Nair and Hinton 2010).

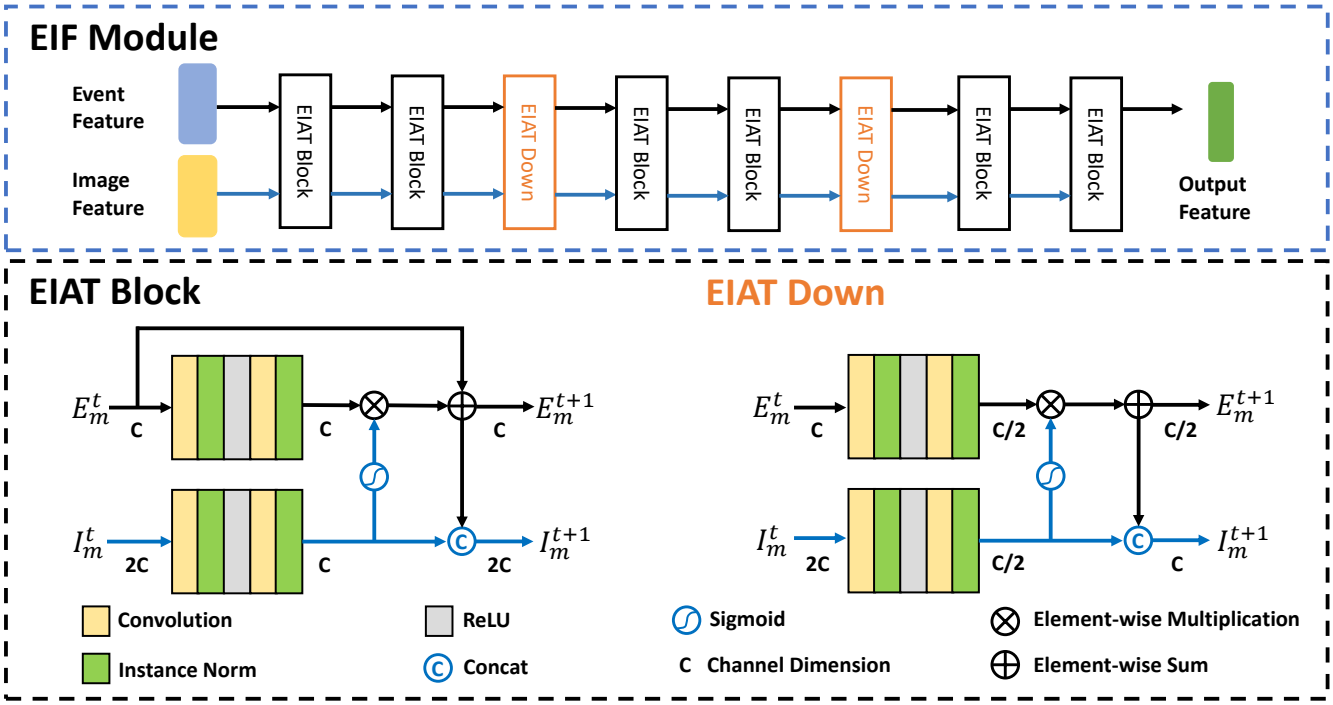

Figure 2: The proposed fusion module. The fusion module receives the image and event features paired with each other in different sizes. It comprises three event image fusion (EIF) modules. Each EIF consists of an EIAT block and an EIAT down.

The attention maps A_m^t, which are values from zero to one, are computed by applying the sigmoid function to the image codes:

$$\begin{aligned} \text{Layer}_I &= \text{norm}(\text{conv}(\text{ReLU}(\text{norm}(\text{conv}(\cdot))))), \\ A_m^t &= \sigma(\text{Layer}_I(I_m^t)) \end{aligned} \tag{2}$$

where t denotes the block sequence number. Note that except for the first $t = 1$, the channel dimension of the image codes is reduced by half, compared to the input image features. The event codes E_m^t are also generated from the convolution layers, instance normalization layers, and ReLU layer; however, the channel dimensions of the event feature and the event codes are the same. The event codes are updated by multiplying them with the attention maps A_m^t. The result of the multiplication is added with event feature E_m^t via the residual connection:

$$\begin{aligned} \text{Layer}_E &= \text{norm}(\text{conv}(\text{ReLU}(\text{norm}(\text{conv}(\cdot))))), \\ E_m^{t+1} &= \text{Layer}_E(E_m^t) \odot A_m^t + E_m^t, \end{aligned} \tag{3}$$

where $\odot$ denotes element-wise multiplication. The image codes should also be updated, including updates of the new event codes.

$$I_m^{t+1} = \text{concat}(\text{Layer}_I(I_m^t), E_m^{t+1}), \tag{4}$$

where concat denotes the channel-wise concatenation of feature level.

The channel dimensions of the features passing through the EIF module are reduced. To compress the dimensions of the feature, the EIAT down block was employed in the EIF module. As illustrated in Fig. 2, the overall structure of the EIAT down is the same as that of the EIAT block, except for the residual connection of event features and convolution layers. In the EIAT down, the event codes passing through the layer have a channel dimension that is reduced by half of the input event feature. For image codes passing through the layer, the channel dimension is reduced by a quarter, compared to the input image feature. The outputs of the image features and event features passing through each branch have channel dimensions reduced by half compared to those before passing through the EIAT down block. In the EIF module, EIF_0 and EIF_1 contain two EIAT down blocks and six EIAT blocks. However, EIF_2 contains one EIAT down and four EIAT blocks. Left and Right fusion modules share weights.

Spatial Multi-scale Correlation (SMC)

The left outputs of the fusion module and the right outputs of the fusion module are denoted by $F_{m,L}$ and $F_{m,R}$, respectively, with a scale factor of m. The outputs of the fusion module are generated from different scales with a $1/2^{2-m}$ size of the original image dimension. As presented in Fig. 3, considering coarse to fine features, SMC calculates the feature correlation of multiple scales using patches proportional to the size of the features for the width and height dimensions. The shape of the patch is $2^m \times 2^m$, and the size of the dilation is also 2^m. The value of each voxel in the cost volume considers the similarity between the left and right features of the corresponding patch. The SMC of the cost

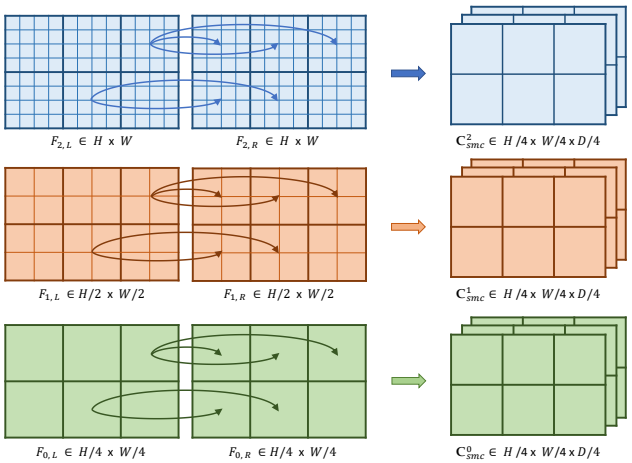

Figure 3: Spatial multi-scale correlation (SMC). SMC generates correlation volumes of the same size from the features of different scales.

volume is computed as:

$$\mathbf{C}^m_{smc}(d,x,y,c) = \frac{1}{2^{2m}} \sum_{p=0}^{2^m-1} \sum_{q=0}^{2^m-1} \langle F_{m,L}(2^m x + p, 2^m y + q, c), F_{m,R}(2^m x + p - 2^m d, 2^m y + q, c) \rangle, \quad (5)$$

where $m = 0, 1, 2$, and $\langle \cdot, \cdot \rangle$ represent the inner product. When $m = 0$, the SMC becomes a conventional correlation. The correlation is computed for all disparities d and all channel levels c. Then, the outputs of the fusion module of multiple scales form a cost volume with the same shape as $D_{max}/4 \times H/4 \times W/4 \times C$, except for the channel dimension, where C is 32, 16, 16, respectively, and D_{max} is the maximum disparity. The final cost volume is the concatenation of $\mathbf{C}^0_{smc}$, $\mathbf{C}^1_{smc}$, and $\mathbf{C}^2_{smc}$ in the channel dimension. SMC calculates the correlation between the left and right features in patch-based, and efficiently considers large-scale features.

3D Aggregation Network

We utilize the stacked hourglass architecture proposed in (Guo et al. 2019). They modified the stacked hourglass architecture proposed in (Chang and Chen 2018). While they added an auxiliary output module to improve accuracy, they eliminated residual connections and used $1 \times 1 \times 1$ 3D convolution for short connections to save inference computation. There are four output modules for training, and only the last output module is adopted for inference. In each output module, the probability volume with a size of $D_{max} \times H \times W \times 1$ is generated using two 3D convolutions with upsampling and softmax function. The estimated disparity $\widetilde{\mathbf{D}}$ of each pixel can be obtained as follows:

$$\widetilde{\mathbf{D}} = \sum_{d=0}^{D_{max}-1} d \cdot p_d, \quad (6)$$

where d and p_d denote the possible disparity value and corresponding probability, respectively.

Dual Learning with Sparse Disparity

We dual-train the branch for sparse disparity estimation during training to use edge-related information more effectively from events. For sparse disparity estimation, the loss is solely computed in sparse locations corresponding to the 15,000 most recent events. As presented in Fig. 1, the sparse branch estimates sparse disparity via correlation and 3D aggregation sub-networks using only the event features passing through the feature extractor. Before passing through the correlation network, the channel dimension of event features with multiple scales is reduced through convolution and ReLU layers as in the fusion module. By reducing the channel dimension, a cost volume with the same size as in dense disparity prediction is generated. To save computation time during inference, the sparse disparity branch is solely used during training. The experimental results indicate that only the extra sparse estimation branch during training is complementary to dense disparity estimation, without significantly increasing the memory footprint.

Objective Function

We adopt the smooth L_1 loss function to train the proposed model. Smooth L_1 can be obtained as:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (7)$$

The predicted dense disparity maps from the four output modules are denoted as $\widetilde{\mathbf{D}}_0$, $\widetilde{\mathbf{D}}_1$, $\widetilde{\mathbf{D}}_2$, $\widetilde{\mathbf{D}}_3$, and the predicted sparse disparity maps are represented as $\widetilde{\mathbf{d}}_0$, $\widetilde{\mathbf{d}}_1$, $\widetilde{\mathbf{d}}_2$, $\widetilde{\mathbf{d}}_3$. Then,

$$L_{dense} = \sum_{i=0}^{i=3} \lambda_i \cdot \text{smooth}_{L_1}\left(\widetilde{\mathbf{D}}_i - \mathbf{D}^*\right), \quad (8)$$

where $\mathbf{D}^*$ denotes the ground-truth for the dense disparity map, and

$$L_{sparse} = \sum_{j=0}^{j=3} \lambda_{j+4} \cdot \text{smooth}_{L_1}\left(\widetilde{\mathbf{d}}_j - \mathbf{d}^*\right), \quad (9)$$

where $\mathbf{d}^*$ denotes the ground truth for the sparse disparity map. We apply the sparse loss (L_{sparse}) to the location that corresponds to the 15,000 most recent events. Our final loss (L) is obtained by combining the dense (L_{dense}) and the sparse (L_{sparse}) losses as

$$L = L_{dense} + L_{sparse}. \quad (10)$$

Experiments and Results

Datasets

We used two different datasets for the performance evaluation. One dataset is the MVSEC (Zhu et al. 2018) of actual event data, and the other is the simulated dataset that we generated in this work.

MVSEC comprises two DAVIS cameras in a stereo setting, which provides an intensity image and event stream with a spatial resolution of 346×260. We split indoor flying

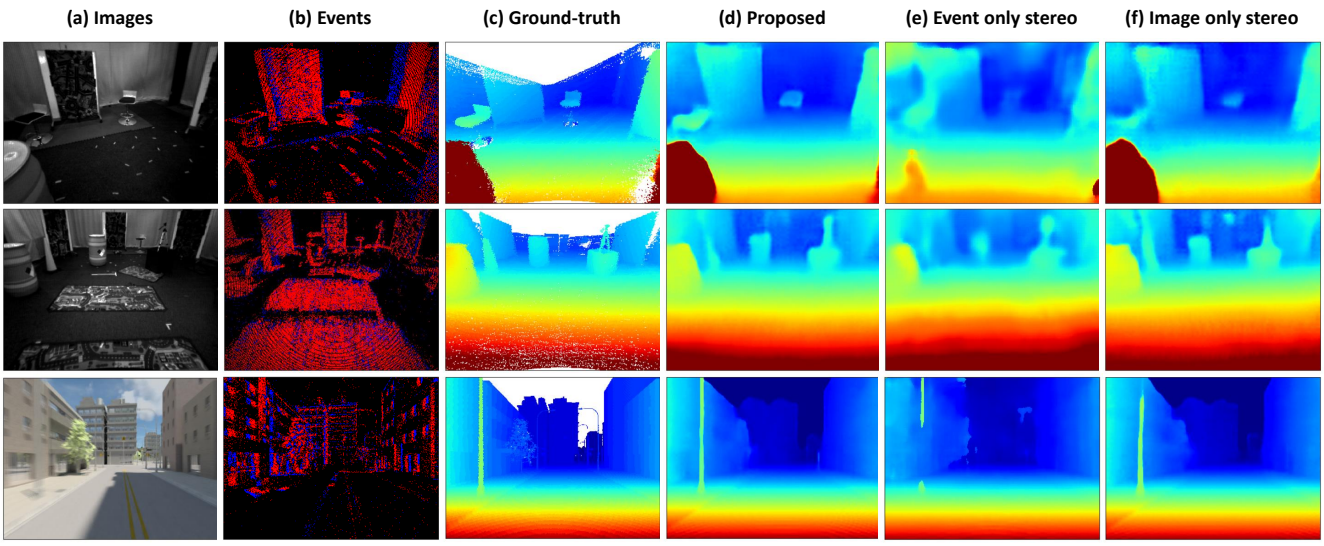

Figure 4: Qualitative comparison of the proposed method with an event-based method and a frame-based method. The first two rows are split 1 and split 3 from the MVSEC (real-world) dataset, and the third row is the RGB frame-based synthetic dataset, respectively. In (a) and (b), we visualize the image and the 15,000 most recent events from the left camera. Note that (e) and (f) are the results of (Tulyakov et al. 2019) and (Guo et al. 2019), respectively. Our proposed method (d) utilizes an image with the corresponding events.

Table 1: Results obtained for dense disparity estimation on MVSEC datasets. I indicates that the intensity image is adopted as the model input data, and E implies that the event data are adopted as the input. E + I means both conditions are adopted. The time per image denotes the time taken to infer a disparity image.

Model	**Using data**	**Mean disparity error[pix]** ↓		**One-pixel accuracy [%]** ↑		**Mean depth error [cm]** ↓		**Median depth error [cm]** ↓		**time per image [sec]** ↓
		Split 1	Split 3	Split 1	Split 3	Split 1	Split 3	Split 1	Split 3	
PSMNet (Chang and Chen 2018)	I	0.57	0.68	88.6	83.1	15.9	18.3	8.0	10.2	0.10
GwcNet-gc (Guo et al. 2019)	I	0.53	0.64	89.9	85.8	15.0	17.4	7.5	9.3	0.06
PSN (Tulyakov, Ivanov, and Fleuret 2018)	I	0.63	1.03	87.2	71.7	16.8	23.8	8.5	15.2	**0.05**
GC-Net (Kendall et al. 2017)	I	0.55	0.75	88.6	83.8	15.3	18.7	7.8	11.1	0.13
PSN (Tulyakov et al. 2019)	E	0.59	0.94	89.8	82.5	16.6	23.5	6.8	14.7	0.06
(Ahmed et al. 2021)	E	0.55	0.75	92.1	89.6	14.2	19.4	5.9	10.4	–
SMC-Net w/o Sprase Branch (ours)	E + I	0.40	0.53	93.9	91.9	11.8	14.6	4.7	6.4	0.12
SMC-Net (ours)	E + I	**0.37**	**0.52**	**94.3**	**92.0**	**11.2**	**14.5**	**4.5**	**6.3**	0.12

from MVSEC into three and used the two of them, split 1 and split 3, following (Tulyakov et al. 2019; Ahmed et al. 2021).

Although RGB can be sufficiently effective information for stereo, previous studies compared event stereo solely with intensity image-based stereo owing to the limitations of the dataset. For comparison with RGB images, we created a synthetic dataset containing RGB images. Our synthetic dataset was generated using a 3D computer graphics software called Blender (Community 2018). We generated RGB images with a spatial resolution of 346×260 and depth maps using open-source data (Zhang et al. 2016), including city driving scenarios; then, we simulated the event streams using the event simulator ESIM (Rebecq, Gehrig, and Scaramuzza 2018). We split the data into 9,000 samples for training, 200 samples for validation, and 2,000 samples for the test set. We introduced blur to the image by averaging seven RGB frames with a high frame rate to cover actual driving scenarios.

Experimental Setup

The coefficients of Eq. 8 were set to $\lambda_0 = 0.5$, $\lambda_1 = 0.5$, $\lambda_2 = 0.7$, $\lambda_3 = 1.0$. Similarly, the coefficients of Eq. 9 were set to $\lambda_4 = 0.5$, $\lambda_5 = 0.5$, $\lambda_6 = 0.7$, $\lambda_7 = 1.0$. For comparison, we trained both our networks and other models using the RMSprop optimizer. The coefficients used for each model of the other models are in agreement with those suggested in (Chang and Chen 2018; Guo et al. 2019; Tulyakov, Ivanov, and Fleuret 2018; Kendall et al. 2017; Tulyakov et al. 2019). We adopted a single NVIDIA TITAN

Table 2: Results obtained for dense disparity estimation of synthetic datasets. RGB indicates that the RGB image is used as the frame-based model input data, and E represents the event data used as the input. E + RGB means both are adopted.

Model	Using data	Mean disparity error[pix] ↓	>2px [%] ↓	>3px [%] ↓	Mean depth error [m] ↓	time per image [sec] ↓
PSMNet (Chang and Chen 2018)	RGB	0.98	7.87	5.35	0.38	0.15
GwcNet-gc (Guo et al. 2019)	RGB	0.88	7.42	4.52	0.38	0.12
PSN (Tulyakov, Ivanov, and Fleuret 2018)	RGB	1.04	7.84	5.14	0.48	0.08
GC-Net (Kendall et al. 2017)	RGB	1.12	11.18	7.67	0.51	0.19
PSN (Tulyakov et al. 2019)	E	1.25	12.36	6.74	0.60	**0.07**
SMC-Net w/o Sparse Branch (ours)	E + RGB	0.84	6.27	3.88	0.35	0.17
SMC-Net (ours)	E + RGB	**0.82**	**6.07**	**3.46**	**0.32**	0.17

Table 3: Ablation studies of the proposed fusion module on MVSEC datasets. We evaluate the performance of the fusion module while maintaining the correlation method and the overall framework, which is the proposed network: SMC-Net.

Fusion module	Using data	Mean disparity error[pix] ↓		One-pixel accuracy [%] ↑		Mean depth error [cm] ↓		Median depth error [cm] ↓	
		Split 1	Split 3	Split 1	Split 3	Split 1	Split 3	Split 1	Split 3
Concat	E + I	0.43	0.53	93.5	91.0	12.5	15.0	5.2	6.6
SPADE (Park et al. 2019)	E + I	0.42	**0.51**	93.5	91.8	12.6	14.7	5.4	**6.2**
EIAT (ours)	E + I	**0.40**	0.53	**93.9**	**91.9**	**11.8**	**14.6**	**4.7**	6.4

RTX GPU for training and inference. Regarding the model used for testing, the models with the best performance in the validation set were selected among those trained for up to 30 epochs until convergence.

Qualitative and Quantitative Results

As illustrated in Fig. 4, we qualitatively compared the results of the proposed method with those of other methods. The proposed method, which adopts both events and images, outperformed other methods that employed either events or images. Hence, the proposed method provides a better dense depth even in a region with less textures and similar intensity to the surroundings. In contrast, the event- and image-based methods are limited in areas with less textures and in the edge and ground areas, respectively.

For quantitative analysis, we compared the results of our proposed model with event- and image-based stereos using mean depth error, median depth error, mean disparity error, and one-pixel accuracy, in accordance with (Tulyakov et al. 2019; Ahmed et al. 2021). Table 1 presents a comparison of the proposed method with previous methods using actual MVSEC datasets. There was no case of testing frame-based methods on the MVSEC dataset. For comparisons with the frame-based stereo method, we trained the frame-based model using intensity images on the MVSEC dataset. The proposed model outperforms frame- and event-based methods by a large margin for all metrics as in Table 1.

Furthermore, we evaluated the proposed model using a synthetic dataset. In the synthetic dataset, the frame-based method was trained using RGB images, and for the event-based method, only PSN (Tulyakov et al. 2019) with published codes was trained. We trained the proposed model by employing RGB images and events. We applied the same metrics as the real-world dataset to the synthetic dataset for quantitative performance evaluation, except for one-pixel accuracy. In this case, we adopted 2 pixel (>2px) and 3 pixel errors (>3px) for the synthetic dataset. Table 2 indicates that the event-based method exhibits a lower performance than the RGB-based method; however, the proposed model, which employs both RGB images and events, outperforms the RGB-based model.

Ablation Studies

We performed ablation studies to confirm the effectiveness of the proposed methods. In the last two rows of Table 1 and Table 2, the sparse branch in training improves performance without additional time consumption with the inference in both MVSEC and synthetic datasets.

In addition, we validated the ablation studies of the proposed fusion module and correlation network using the MVSEC dataset. To validate the effectiveness of the fusion module, we maintained the entire model as the proposed SMC-Net and replaced the fusion module with concat and SPADE (Park et al. 2019). For concat, we added convolution after concat, such that the number of channels of fused features is the same as that of the proposed EIF module. Table 3 presents the results of the ablation study on the proposed fusion module. From Table 3, except for the mean disparity and median depth errors in split 3, EIF exhibits the best performance. This proves the effectiveness of the fusion module.

In addition, to validate the effectiveness of the proposed correlation network, we trained the other models without altering the overall structure. Instead, we added the proposed EIF module as a fusion module that aggregates event and

Table 4: Ablation studies of the proposed correlation network on the MVSEC datasets. For evaluation using both events and images, the proposed EIF module, which exhibits the best performance, is employed for the fusion module.

Model	Using data	Mean disparity error[pix] ↓		One-pixel accuracy [%] ↑		Mean depth error [cm] ↓		Median depth error [cm] ↓		time per image [sec] ↓
		Split 1	Split 3	Split 1	Split 3	Split 1	Split 3	Split 1	Split 3	
PSMNet (Chang and Chen 2018)	E + I	0.42	0.59	93.5	89.8	12.4	16.0	5.1	7.9	0.16
GwcNet-gc (Guo et al. 2019)	E + I	0.41	0.58	**93.9**	91.0	12.1	15.0	4.9	7.4	0.12
PSN (Tulyakov, Ivanov, and Fleuret 2018)	E + I	0.56	0.65	90.7	88.4	15.6	19.3	7.1	9.1	**0.10**
GC-Net (Kendall et al. 2017)	E + I	0.44	0.58	92.5	90.2	12.6	15.1	5.5	7.6	0.25
SMC-Net (ours)	E + I	**0.40**	**0.53**	**93.9**	**91.9**	**11.8**	**14.6**	**4.7**	**6.4**	0.12

image features. In addition, we adopted the same structure as the image feature extractor proposed in each model for the event feature extractor. The detail of the architecture is provided in the supplementary material. We also trained the proposed model by removing the sparse branch for a fair comparison. Table 4 presents the results of the ablation study on the proposed correlation network. From Table 4, the proposed network is determined as the best network considering all accuracy metrics, which also proves the effectiveness of the proposed correlation network.

Conclusion

We presented an end-to-end network to estimate a dense depth map using both images and events. The two types of source input complemented each other for stereo, and the proposed model outperformed the performance of only the image or only event-based methods by significant margins. Accordingly, we proposed an attention-based fusion module to aggregate event features with image features. In addition, we proposed a spatial multi-scale correlation method to consider the coarse to fine scale of features. We also proposed a novel sparse branch mechanism to improve robustness, using the guidance on the disparity of edges from events. To the best of our knowledge, this is one of the first attempts to fuse an event and image for stereo matching. In addition, because the advanced methods of frame-based stereo employ RGB images, we demonstrated the advantage of using RGB images and events together via a comparison with RGB frame-based stereo. Based on these ablation studies, we infer that the proposed network and modules effectively improve the performance of stereo matching.

Acknowledgements

This research was supported by the Defense Challengeable Future Technology Program of Agency for Defense Development, Republic of Korea. This work was also partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2018R1A2B3008640).

References

Ahmed, S. H.; Jang, H. W.; Uddin, S. N.; and Jung, Y. J. 2021. Deep Event Stereo Leveraged by Event-to-Image Translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 882–890.

Brandli, C.; Berner, R.; Yang, M.; Liu, S.-C.; and Delbruck, T. 2014. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10): 2333–2341.

Camunas-Mesa, L. A.; Serrano-Gotarredona, T.; Ieng, S. H.; Benosman, R. B.; and Linares-Barranco, B. 2014. On the use of orientation filters for 3D reconstruction in event-driven stereo vision. *Frontiers in neuroscience*, 8: 48.

Chang, J.-R.; and Chen, Y.-S. 2018. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5410–5418.

Cho, H.; Jeong, J.; and Yoon, K.-J. 2021. EOMVS: Event-Based Omnidirectional Multi-View Stereo. *IEEE Robotics and Automation Letters*, 6(4): 6709–6716.

Community, B. O. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam.

Guo, X.; Yang, K.; Yang, W.; Wang, X.; and Li, H. 2019. Group-wise correlation stereo network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3273–3282.

Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.; Kennedy, R.; Bachrach, A.; and Bry, A. 2017. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, 66–75.

Kogler, J.; Humenberger, M.; and Sulzbachner, C. 2011. Event-based stereo matching approaches for frameless address event stereo data. In *International Symposium on Visual Computing*, 674–685. Springer.

Liang, Z.; Feng, Y.; Guo, Y.; Liu, H.; Chen, W.; Qiao, L.; Zhou, L.; and Zhang, J. 2018. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2811–2820.

Nair, V.; and Hinton, G. E. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*.

Park, T.; Liu, M.-Y.; Wang, T.; and Zhu, J.-Y. 2019. Semantic Image Synthesis With Spatially-Adaptive Normalization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2332–2341.

Piatkowska, E.; Belbachir, A.; and Gelautz, M. 2013. 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*, 45–50.

Piatkowska, E.; Kogler, J.; Belbachir, N.; and Gelautz, M. 2017. Improved cooperative stereo matching for dynamic vision sensors with ground truth evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 53–60.

Rebecq, H.; Gallego, G.; Mueggler, E.; and Scaramuzza, D. 2017. EMVS: Event-Based Multi-View Stereo—3D Reconstruction with an Event Camera in Real-Time. *International Journal of Computer Vision*, 126: 1394–1414.

Rebecq, H.; Gehrig, D.; and Scaramuzza, D. 2018. ESIM: an Open Event Camera Simulator. *Conf. on Robotics Learning (CoRL)*.

Rogister, P.; Benosman, R.; Ieng, S.-H.; Lichtsteiner, P.; and Delbruck, T. 2011. Asynchronous event-based binocular stereo matching. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2): 347–353.

Tulyakov, S.; Fleuret, F.; Kiefel, M.; Gehler, P.; and Hirsch, M. 2019. Learning an event sequence embedding for dense event-based deep stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1527–1537.

Tulyakov, S.; Ivanov, A.; and Fleuret, F. 2018. Practical deep stereo (pds): Toward applications-friendly deep stereo matching. *arXiv preprint arXiv:1806.01677*.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2016. Instance Normalization: The Missing Ingredient for Fast Stylization. *ArXiv*, abs/1607.08022.

Xu, H.; and Zhang, J. 2020. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1959–1968.

Yang, G.; Song, X.; Huang, C.; Deng, Z.; Shi, J.; and Zhou, B. 2019. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 899–908.

Yang, G.; Zhao, H.; Shi, J.; Deng, Z.; and Jia, J. 2018. Segstereo: Exploiting semantic information for disparity estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 636–651.

Yao, Y.; Luo, Z.; Li, S.; Fang, T.; and Quan, L. 2018. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 767–783.

Zhang, F.; Prisacariu, V.; Yang, R.; and Torr, P. H. 2019. Ganet: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 185–194.

Zhang, F.; Qi, X.; Yang, R.; Prisacariu, V.; Wah, B.; and Torr, P. 2020. Domain-invariant stereo matching networks. In *European Conference on Computer Vision*, 420–439. Springer.

Zhang, Z.; Rebecq, H.; Forster, C.; and Scaramuzza, D. 2016. Benefit of large field-of-view cameras for visual odometry. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 801–808. IEEE.

Zhong, Y.; Dai, Y.; and Li, H. 2017. Self-supervised learning for stereo matching with self-improving ability. *arXiv preprint arXiv:1709.00930*.

Zhu, A. Z.; Chen, Y.; and Daniilidis, K. 2018. Realtime time synchronized event-based stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 433–447.

Zhu, A. Z.; Thakur, D.; Özaslan, T.; Pfrommer, B.; Kumar, V. R.; and Daniilidis, K. 2018. The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. *IEEE Robotics and Automation Letters*, 3: 2032–2039.

Zhu, Z.; Huang, T.; Shi, B.; Yu, M.; Wang, B.; and Bai, X. 2019. Progressive Pose Attention Transfer for Person Image Generation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2342–2351.

Zou, D.; Guo, P.; Wang, Q.; Wang, X.; Shao, G.; Shi, F.; Li, J.; and Park, P.-K. 2016. Context-aware event-driven stereo matching. In *2016 IEEE International Conference on Image Processing (ICIP)*, 1076–1080. IEEE.

Zou, D.; Shi, F.; Liu, W.; Li, J.; Wang, Q.; Park, P.-K.; Shi, C.-W.; Roh, Y. J.; and Ryu, H. E. 2017. Robust dense depth map estimation from sparse DVS stereos. In *British Mach. Vis. Conf.(BMVC)*, volume 1.