RPEFlow: Multimodal Fusion of RGB-PointCloud-Event for Joint Optical Flow and Scene Flow Estimation

Zhexiong Wan[1] Yuxin Mao[1] Jing Zhang[2] Yuchao Dai[1]†
[1]Northwestern Polytechnical University & Shaanxi Key Laboratory of Information Acquisition and Processing [2]Australian National University

Abstract

*Recently, the RGB images and point clouds fusion methods have been proposed to jointly estimate 2D optical flow and 3D scene flow. However, as both conventional RGB cameras and LiDAR sensors adopt a frame-based data acquisition mechanism, their performance is limited by the fixed low sampling rates, especially in highly-dynamic scenes. By contrast, the event camera can asynchronously capture the intensity changes with a very high temporal resolution, providing complementary dynamic information of the observed scenes. In this paper, we incorporate **R**GB images, **P**oint clouds and **E**vents for joint optical flow and scene flow estimation with our proposed multi-stage multimodal fusion model, **RPEFlow**. First, we present an attention fusion module with a cross-attention mechanism to implicitly explore the internal cross-modal correlation for 2D and 3D branches, respectively. Second, we introduce a mutual information regularization term to explicitly model the complementary information of three modalities for effective multimodal feature learning. We also contribute a new synthetic dataset to advocate further research. Experiments on both synthetic and real datasets show that our model outperforms the existing state-of-the-art by a wide margin. Code and dataset is available at https://npucvr.github.io/RPEFlow.*

1. Introduction

Optical flow estimation, *i.e.*, estimating the dense 2D motion between consecutive image frames, has been extensively studied and significantly advanced with the development of deep neural networks [1–3]. Scene flow estimation, on the other hand, aims to estimate the 3D motion field with various input configurations, ranging from monocular images [4, 5], stereo images [6, 7], two frames of point clouds [8, 9], images combined with depth maps [10, 11] or point clouds [12, 13]. Both are fundamental to downstream applications such as autonomous driving [14, 15], object tracking [16, 17], scene reconstruction [18, 19], *etc*.

Due to the strong correlation between 2D and 3D motion, *i.e.*, 2D motion can be regarded as the projection of 3D motion on the image plane, recent works [10, 12, 13] make efforts to jointly estimate optical flow and scene flow by combining RGB images and point clouds (or depth maps). Their success indicates that joint 2D and 3D motion estimation within a framework can obtain more accurate results than separate tasks. However, as both conventional RGB cameras and LiDAR (or depth) sensors adopt a fixed frame-by-frame data acquisition mechanism, these methods show unsatisfactory performance when dealing with complex motion scenes (see Fig. 3), which motivates us to alleviate this problem by introducing the event camera.

Event camera, as a bio-inspired imaging sensor, can asynchronously capture the brightness change with very high temporal resolution (in the order of μs) and output an event signal quickly [20]. As each pixel adapts its sampling rate according to the captured changes, the amount of output events usually depends on the complexity of motion (the faster the motion, the more triggered events), thus providing abundant motion information of the observed scene. Based on this, some works use event data alone to estimate optical flow [21, 22], but they show limitations in estimating reliable motion at regions with no events [23]. As compensation for this, image and event data are fused together to estimate dense optical flow [24, 25]. As far as we know, there is no method to incorporate event data within a multimodal learning framework for both 2D and 3D motion estimation.

In this paper, we propose to fuse RGB images, point clouds and events for joint optical flow and scene flow estimation. We find the ability of the event camera to asynchronously capture the brightness changes caused by motion makes it complementary to image cameras and LiDAR sensors, especially for complex dynamics and high-contrast brightness changes. We believe that combining these three modalities together for 2D and 3D motion estimation meets the practical needs, which has been further confirmed by

† Corresponding author (daiyuchao@gmail.com).

existing datasets, such as MVSEC [26] and DSEC [22] that contain these data for driving scenarios.

We formulate this task as a representation-based multimodal learning problem, and exploit the complementary information between these three very different modalities implicitly and explicitly. We aim to exploit the relationships between multimodal and multi-dimensional space observations (images and events in 2D with point clouds in 3D) and explore their contributions to 2D and 3D motion. Specifically, in our RPEFlow framework, we first propose a multimodal attention fusion module with *cross-attention mechanism* to implicitly explore the correlations between three modalities, based on which a pyramid multi-stage fusion structure is introduced to extensively modeling. We observe that each modality can contribute a part to 2D and 3D motion estimation, making representation learning [27] suitable for our multimodal learning framework. Then we introduce cross-modal mutual information minimization in feature space to explicitly maximize the complementary information. We also contribute a new synthetic dataset with simulations that conform to the gravity model and collision detection and contain a larger variety of moving objects and richer annotations than FlyingThings3D [28]. Extensive experimental results validate both our implicit multimodal attention fusion and explicit representation regularization towards effective multimodal learning, leading to a new benchmark on both synthetic and real-captured datasets.

Our main contributions are summarized as follows:

1) We propose to incorporate event cameras with RGB cameras and LiDAR sensors to jointly estimate optical flow and scene flow for complex dynamic scenes, which constitutes a new and practical problem.
2) An implicit multimodal attention fusion module and an explicit representation learning via mutual information regularization are presented in our RPEFlow model, achieving extensive cross-modal relationship modeling.
3) We contribute a large-scale synthetic dataset with ground-truth motion annotations. Experimental results on both synthetic and real datasets show that the proposed RPEFlow outperforms existing state-of-the-art and demonstrates the effectiveness of event data for motion estimation of complex dynamics.

2. Related Work

2.1. Unimodal 2D/3D Motion Estimation

Image only. For learning-based 2D optical flow estimation, FlowNet series [29, 30] first propose end-to-end CNN models for regression. PWC-Net [1] work on constructing feature pyramids with coarse-to-fine refinement. RAFT [2] and its variants [3, 31] build all-pairs correlation and update the optical flow iteratively. For 3D scene flow estimation, learning-based studies [6, 7] use a sequence of stereo images as input [32], achieving faster and better performance compared with earlier optimization-based methods [15, 33]. Some recent works [4, 5, 34] use only monocular image sequences, which are more difficult to model accurate 3D structure and motion than stereo images. Due to the limited frame rate of input images and the difficulty in obtaining 3D structure, the performance of image-only methods is still unsatisfactory when dealing with complex dynamics.

Point Cloud only. Point cloud data from the LiDAR sensor is favorable for 3D motion estimation. Due to the unique data structure, existing methods [8, 9, 35] focus on studying the model structure to represent the point cloud data for scene flow estimation. However, the point cloud lacks semantic information and leads to the difficulty of estimating accurate motion only by the structural information [13].

Event only. Event-based motion estimation is dedicated to extracting motion information from event-encoded brightness changes. Some early optimization-based methods estimate the motion flow of moving boundaries [36, 37]. Recent learning-based methods [21, 22, 38] are proposed to regress the dense optical flow directly. Even though the predictions of some of them are densely supervised, the sparse input event data leads to unreliable optical flow estimation in the regions without triggered events [23, 24].

2.2. Multimodal 2D/3D Motion Estimation

As unimodal data only provides partial information, multimodal methods are presented to comprehensively learn from multiple observations. For optical flow estimation, event-based studies [24, 25, 39] combine the advantages of the image in dense representation and events in motion perception for reliable estimation. Besides, [40, 41] incorporate gyroscope or depth sensor to guide the optical flow. For scene flow estimation, using a sequences of RGB-Depth images [10, 11] becomes another trend, then [12, 13] replace depth with point clouds to deal with the limited ideal range of depth camera when applied outdoors. But again, they are still limited by the frame-by-frame acquisition mechanism, which leads us to introduce event data.

2.3. Multimodal Fusion

Given multimodal data, effective multimodal fusion is critical to extensively explore the contribution of each modality [42]. Two main directions have been explored: 1) attention based [43–45] and 2) representation learning [46–49] based. For the former, a specific attention module [50, 51] is designed to implicitly control the contribution of each modal. For the latter, representation similarity is measured to explicitly constrain the reliability of the feature embedding. Within this direction, mutual information (MI) estimation and optimization [48, 52–54] is the widely studied strategy, which is typically used as a regularizer to encourage (via MI maximization) or limit dependency (via

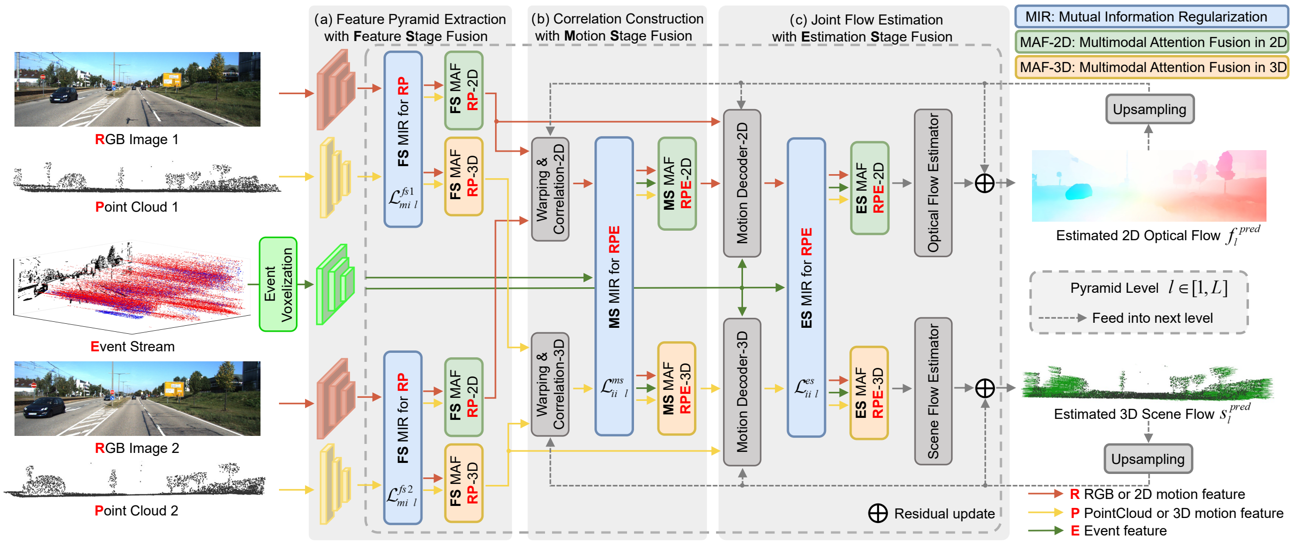
(a) Feature Pyramid Extraction with Feature Stage Fusion
(b) Correlation Construction with Motion Stage Fusion
(c) Joint Flow Estimation with Estimation Stage Fusion
MIR: Mutual Information Regularization
MAF-2D: Multimodal Attention Fusion in 2D
MAF-3D: Multimodal Attention Fusion in 3D
RGB Image 1
Point Cloud 1
Event Stream
RGB Image 2
Point Cloud 2
Event Voxelization
FS MIR for RP
$\mathcal{L}_{mi\ l}^{fs1}$
$\mathcal{L}_{mi\ l}^{fs2}$
FS MAF RP-2D
FS MAF RP-3D
Warping & Correlation-2D
Warping & Correlation-3D
MS MIR for RPE
$\mathcal{L}_{ii\ l}^{ms}$
MS MAF RPE-2D
MS MAF RPE-3D
Motion Decoder-2D
Motion Decoder-3D
ES MIR for RPE
$\mathcal{L}_{ii\ l}^{es}$
ES MAF RPE-2D
ES MAF RPE-3D
Optical Flow Estimator
Scene Flow Estimator
Upsampling
Estimated 2D Optical Flow f_l^{pred}
Pyramid Level $l \in [1, L]$
Feed into next level
Estimated 3D Scene Flow s_l^{pred}
Upsampling
Residual update
R RGB or 2D motion feature
P PointCloud or 3D motion feature
E Event feature

Figure 1: **Our RPEFlow Structure.** We learn motion correlations from the input three modalities (RGB-PointCloud-Event, RPE) by multi-stage fusion (FS, MS and ES) incorporating explicit multimodal attentional fusion (MAF) and implicit representing learning with mutual information regularization (MIR), and perform pyramidal updates from coarse to fine to estimate the optical flow in the 2D branch (top) and the scene flow in the 3D branch (bottom), respectively. Best viewed on screen.

MI minimization) between variables.

3. Our Method

We introduce two main strategies to achieve effective multimodal learning, one is explicit multimodal attention fusion (Sec. 3.1) and the other is implicit mutual information regularization (Sec. 3.2). Based on them, we propose a pyramid multi-stage framework (see Fig. 1) for RGB-PointCloud-Event fusion and joint optical flow and scene flow estimation in 2D and 3D branches (Sec. 3.3).

3.1. Multimodal Attention Fusion (MAF)

As shown in Fig. 1, we have both 2D and 3D branches for optical flow and scene flow estimation. Due to the different data structures of the two branches [8, 13], we design symmetric attention fusion strategy for both two branches in Fig. 2, which consists of two steps, namely feature projection and cross-attention fusion. In the 2D branch, we treat the RGB image as the primary modality and project point cloud feature into the image plane, then fuse with auxiliary features, *i.e.*, event and point cloud. For the 3D branch, we define the point cloud data as the primary modality, then project the others into the 3D space and fuse them together. **Multimodal Attention Fusion in 3D branch (MAF RPE-3D).** We take the fusion module in the 3D branch of a single pyramid level as a detailed example, where the encoded image feature is $e_r \in \mathbb{R}^{H \times W \times C_{2D}}$, event feature is $e_{ev} \in \mathbb{R}^{H \times W \times C_{2D}}$, and point cloud feature is $e_{pc} \in \mathbb{R}^{N \times C_{3D}}$ with the point positions $\mathbf{p} = \{\mathbf{p}_{x_i}, \mathbf{p}_{y_i}, \mathbf{p}_{z_i}\}^N \in \mathbb{R}^{N \times 3}$ in 3D space. Note that H, W and N are the feature size at the current pyramid level, not the original input size. We first use the point position to sample the corresponding image and event feature into 3D space with focal length f and denote the projected point position at the image plane as:

$$\{(u_i, v_i)\}^N = \{(f \frac{\mathbf{p}_{x_i}}{\mathbf{p}_{z_i}}, f \frac{\mathbf{p}_{y_i}}{\mathbf{p}_{z_i}})\}^N \in \mathbb{R}^{N \times 2}. \tag{1}$$

Thus, the projected image and event features are:

$$e_r^{pj} = \{e_r(u_i, v_i)\}^N, e_{ev}^{pj} = \{e_{ev}(u_i, v_i)\}^N \in \mathbb{R}^{N \times C_{2D}}, \tag{2}$$

where $e_r(u_i, v_i)$ represents the feature obtained by bilinear interpolation sampling at (u_i, v_i) position in image plane.

After projection, we feed the features of auxiliary modalities e_r^{pj} and e_{ev}^{pj} with the primary modality feature $X_{pri}^{3D} = e_{pc}$ into the attention fusion module. In the original self-attention mechanism [50, 51], all of the keys K, values V and queries Q come from the same modality. Here we adapt it to accommodate inputs from multiple modalities and propose our cross-attention fusion structure. Specifically, we first combine the auxiliary features and align the number of channels with the master feature by 1×1 convolution, yielding the aligned auxiliary feature $Y_{aux}^{3D} = W_a \left[e_r^{pj}, e_{ev}^{pj}\right]$ ($\mathbb{R}^{N \times C_{3D}} \Leftarrow \mathbb{R}^{N \times (C_{2D} + C_{2D})}$), where $[\cdot, \cdot]$ is the concatenation operation. With layer normalization [55] (LN), we apply 3×3 depth-wise convolution to encode spatial and channel information with queries $Q_{pri}^{3D} = W_d^Q \text{LN}(X_{pri}^{3D})$, keys $K_{aux}^{3D} = W_d^K \text{LN}(Y_{aux}^{3D})$ and values $V_{aux}^{3D} = W_d^V \text{LN}(Y_{aux}^{3D})$, respectively, obtaining the cross-modal self-attention as:

$$\textbf{Attention}(Q, K, V) = V \textbf{Softmax}\left(\frac{Q^T K}{\tau}\right), \tag{3}$$

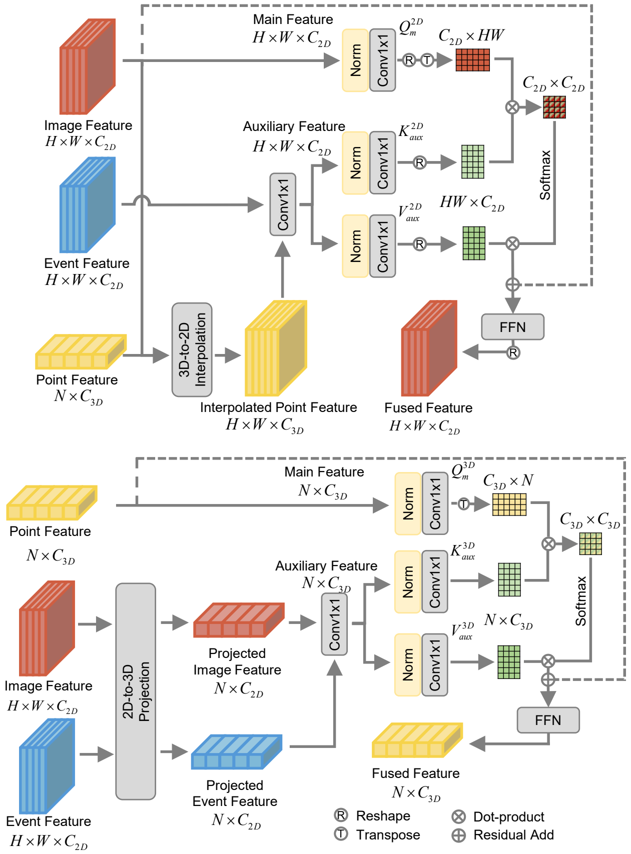

Figure 2: **Our proposed Multimodal Attention Fusion Module fuses both three modal features in 2D (top) and 3D (bottom) branches**, including feature projection and cross-attention fusion. In particular, only image and point features are fused in Feature Stage (FS).

where τ is a learnable scaling factor. We input Q^{3D}_{pri}, K^{3D}_{aux}, V^{3D}_{aux} to the above attention module, and the resultant attention map with dimension $\mathbb{R}^{C_{3D} \times C_{3D}}$ is much smaller and more efficient than the original implementation [51] with dimension $\mathbb{R}^{HW \times HW}$ adopted from [56]. With the cross-modal attention in Eq. 3, we obtain the fused feature $X^{3D'}_{pri}$ corresponding to the primary modality e_{pc} by another 1×1 convolution projection and residual connection as:

$$X^{3D'}_{pri} = W_p \textbf{Attention}(Q^{3D}_{pri}, K^{3D}_{aux}, V^{3D}_{aux}) + X^{3D}_{pri}. \tag{4}$$

Multimodal Attention Fusion in 2D branch (MAF RPE-2D). The fusion process in 2D is similar to the 3D branch above. To project the sparse point feature into a dense feature at the image plane, we adopt a learnable fusion-aware interpolation [13] and get the projected point feature $e^{pj}_{pc} \in \mathbb{R}^{H \times W \times C_{3D}}$. Then we regard the image feature as the primary modality feature $X^{2D}_{pri} = e_r$, then the aligned auxiliary feature $Y^{2D}_{aux} = W^{2D}_{a}\left[e^{pj}_{pc}, e_{ev}\right]$. After similar cross-attention, the fused feature $X^{2D'}_{pri}$ is obtained as:

$$X^{2D'}_{pri} = W_p \textbf{Attention}(Q^{2D}_{pri}, K^{2D}_{aux}, V^{2D}_{aux}) + X^{2D}_{pri}. \tag{5}$$

3.2. Mutual Information Regularization (MIR)

We incorporate mutual information minimization as a regularizer to explicitly model the cross-modal dependency following disentangled representation learning [27] based on the observation that each modality, *i.e.* RGB image, point cloud and event data, contributes partially to the output, and mutual information minimization is suitable for our task to explore the complementary information of each modality.

We start with the case of two modalities with RGB image and event feature embeddings e_r and e_{ev}. Note that these feature embeddings are the features that have been projected into the same spatial space (2D or 3D) in Sec. 3.1.

To explicitly model the cross-modal correlation, we define cross-modal mutual information between e_r and e_{ev} as $I(e_r; e_{ev}) = \mathbb{E}_{p(e_r, e_{ev})}\left[\log \frac{p(e_{ev}|e_r)}{p(e_{ev})}\right]$, where $p(e_r, e_{ev})$ is the joint distribution, $p(e_{ev}|e_r)$ is the conditional distribution and $p(e_{ev})$ is the marginal distribution. With importance sampling [57], we introduce a variational marginal approximation $q(e_{ev})$ with a variational upper bound I^{vub} [54] of mutual information $I(e_r; e_{ev})$ as:

$$\begin{aligned} I(e_r; e_{ev}) \leq & \mathbb{E}_{p(e_r, e_{ev})}\left[\log \frac{p(e_{ev}|e_r)}{q(e_{ev})}\right] \\ & = D_{KL}(p(e_{ev}|e_r) \| q(e_{ev})) = I^{vub} = \mathcal{L}_{\text{mi}}, \end{aligned} \tag{6}$$

where $q(e_{ev})$ can be fixed as a standard normal distribution [58], *i.e.* $q(e_{ev}) = \mathcal{N}(e_{ev}; 0, \mathbf{I})$, and $p(e_{ev}|e_r)$ can be modeled with the reparameterization trick [59], thus the Kullback-Leibler (KL) divergence term D_{KL} within I^{vub} can be solved in closed form.

In the case of three modalities feature embeddings e_r, e_{pc} and e_{ev}, respectively, the interaction information $II(e_r; e_{pc}; e_{ev})$ [60], as a multivariate generalization of the mutual information, is upper bounded by:

$$\begin{aligned} & II(e_r; e_{pc}; e_{ev}) \leq \min\{I(e_r; e_{pc}), I(e_{pc}; e_{ev}), I(e_r; e_{ev})\} \\ & \leq \min\{I^{vub}(e_r; e_{pc}), I^{vub}(e_{pc}; e_{ev}), I^{vub}(e_r; e_{ev})\}. \end{aligned} \tag{7}$$

To compute I^{vub}, we need to design a transition function, achieving the transformation of one modality to the other and assume the variational marginal approximation q as the standard normal distribution for closed-form KL divergence computation. In practice, the standard normal distribution assumption of q leads to high-bias mutual information estimation. Alternatively, we first map the representation of each modality (e_r, e_{pc}, e_{ev}) to a common manifold, with reparameterization trick [59] in the end to achieve Gaussian latent code of each modality. Then, we compute KL divergence between the two Gaussian latent codes, leading to the final mutual information regularization term:

$$\mathcal{L}_{\text{ii}} = I^{vub}(e_r; e_{pc}) + I^{vub}(e_{pc}; e_{ev}) + I^{vub}(e_r; e_{ev}), \quad (8)$$

where we compute the sum of I^{vub} instead of choosing the minimum for stable training.

3.3. Pyramid Fusion and Joint Estimation Model

With the proposed multimodal attention fusion and the mutual information regularization term, we achieve once multimodal feature fusion. Inspired by CamLiFlow [13], we further perform multi-stage feature fusion implicitly and explicitly. Here we present the details of multi-stage feature fusion, and more details about the network structure are given in the supplementary materials. As shown in Fig. 1, our model contains both 2D and 3D branches, where each branch consists of feature extraction, correlation construction and flow estimation.

In feature extraction, we first voxelized the raw events $E = \{x_i, y_i, t_i, p_i\}^K$ into one event voxel $EV \in \mathbb{R}^{H \times W \times B}$ that can be used as input to our network, where K is the number of events during the period between two frames and B is the manually set number of time intervals to sample events. We apply three Siamese encoders to construct feature pyramids ($\{e_{r_1}, e_{r_2}\}_l$, $\{e_{pc_1}, e_{pc_2}\}_l$ and e_{ev_l} with pyramid layers $l \in [1, L]$) for three modalities respectively. As spans between two frames, the event data is not included in *Feature Stage Fusion*, which is applied to fuse the features of two frames RGB images and corresponding point clouds with two-modal attention fusion, *i.e.* simplify the auxiliary feature to a single modality in Eq. 4, 5 and mutual information regularization in Eq. 6 for both 2D and 3D branches.

In correlation construction, we first warp the second frame of image and point cloud features using the coarse optical flow and scene flow (initialize with zero) from the previous pyramid layer, then construct correlation motion features by computing 2D and 3D cost volumes and fused them with the event feature at *Motion Stage Fusion*. In flow estimation, we construct a motion decoder and flow estimator, and perform an *Estimation Stage Fusion* between them to fuse the hidden motion features from the two branches decoder with the event feature. In these two fusion stages, we conduct multimodal attention fusion in Eq. 4, 5 and mutual information regularization in Eq. 8 for 2D and 3D motion features and event feature, because events can provide complementary information to enhance the motion correlation construction. The estimated optical flow and scene flow are fed into the next pyramid layer to achieve coarse-to-fine predictions. We take the optical flow and scene flow from the last pyramid layer as our final joint estimations.

3.4. Objective Functions

The objective functions in the training of our model are divided into feature representation loss and task loss. The former consists of multiple mutual information regularizations at each fusion stage. ${\mathcal{L}_{\text{mi}}^{fs1}}_l$, ${\mathcal{L}_{\text{mi}}^{fs2}}_l$ represent the mutual information minimization loss imposed on the RGB image and point cloud features of the first and the second frame at the **F**eature **S**tage fusion, and ${\mathcal{L}_{\text{ii}}^{ms}}_l$ and ${\mathcal{L}_{\text{ii}}^{es}}_l$ represent on both RGB image, point cloud and event features at the **M**otion **S**tage and **E**stimation **S**tage fusion. Thus the feature representation loss is the sum of all stages and is weighted at each pyramid level:

$$\mathcal{L}_{\text{feat}} = \sum_{l=1}^{L} \lambda_l \left[{\mathcal{L}_{\text{mi}}^{fs1}}_l + {\mathcal{L}_{\text{mi}}^{fs2}}_l + {\mathcal{L}_{\text{ii}}^{ms}}_l + {\mathcal{L}_{\text{ii}}^{es}}_l \right], \quad (9)$$

where l is the pyramid level, and λ_l is used to reweight the contribution of each pyramid level.

The latter task loss measures the L_2 distance between the ground-truth and the model output at each pyramid level by:

$$\begin{aligned}\mathcal{L}_{\text{task}} = \sum_{l=1}^{L} \lambda_l \Big[& \sum_{\mathbf{x}} (\| f_l^{pred}(\mathbf{x}) - f_l^{gt}(\mathbf{x}) \|_2) + \\ & \alpha \sum_{\mathbf{p}} (\| s_l^{pred}(\mathbf{p}) - s_l^{gt}(\mathbf{p}) \|_2) \Big],\end{aligned} \quad (10)$$

where $\mathbf{x}$, $\mathbf{p}$ are the valid image positions and point coordinates, f_l^{pred}, s_l^{pred} are the estimated 2D optical flow and 3D scene flow and f_l^{gt}, s_l^{gt} are the corresponding resized ground truth at the l-th pyramid level, respectively. α is the weight to balance 2D and 3D errors.

The total loss is a weighted sum of the above:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \beta \mathcal{L}_{\text{feat}}, \quad (11)$$

where β is the weight to balance two losses in training.

4. Experiment

4.1. Implementation Details

Datasets. Since there is no large-scale scene flow dataset with real event data, we use synthetic data for pretraining. Follow the preprocess pipeline [13, 62], we generate point clouds from depth images for FlyingThings3D [28] dataset, which contains 19,640 and 3,824 RGB-PointColud pairs for "train" and "val" splits. We use the popular video-to-events conversion method [63] to generate the corresponding events. In addition, we use kubric [64] to simulate 15,367 RGB-PointCloud-Event pairs with rich annotations (including optical flow and scene flow ground truths), denoted as **EKubric**, which aims to simulate photo-realistic

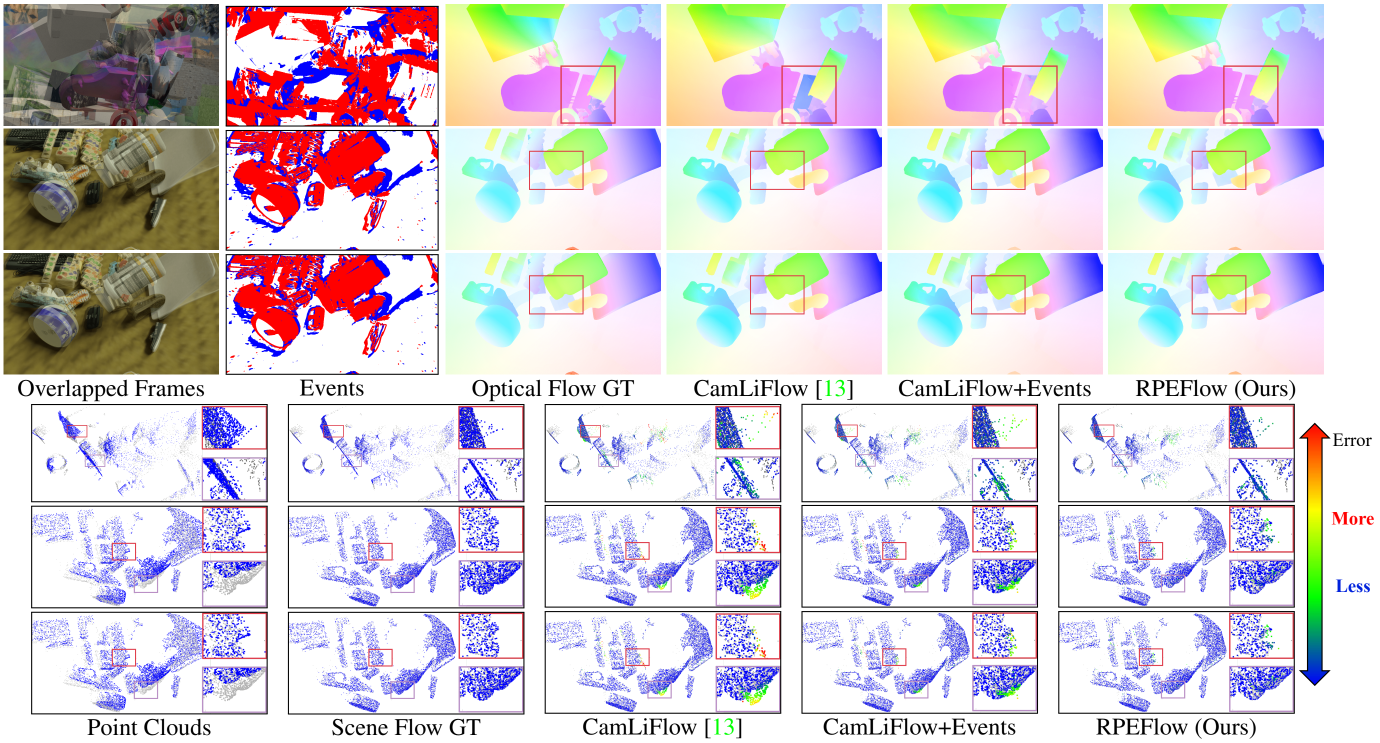

Figure 3: **Visual comparisons** on simulated data, in which the top row is on the "val" split of FlyingThings3D [28] dataset and the reset two are on the test split of our simulated EKubric dataset. For the bottom 3D comparisons, blue indicates a lower error, red indicates a higher error, and green indicates the median. Best to zoom in on the screen for detailed comparisons.

Table 1: **Performance comparison** on the "val" split of the FlyingThings3D [28] subset.

Input	Method	$\mathrm{EPE}_{\mathrm{2D}}$	$\mathrm{ACC}_{\mathrm{1px}}$	$\mathrm{EPE}_{\mathrm{3D}}^{\mathrm{N.Occ}}$	$\mathrm{ACC}_{.05}^{\mathrm{N.Occ}}$	$\mathrm{EPE}_{\mathrm{3D}}^{\mathrm{Full}}$	$\mathrm{ACC}_{.05}^{\mathrm{Full}}$
RGB	RAFT [2]	3.12	81.1%	-	-	-	-
	FlowFormer [3]	3.02	82.6%	-	-	-	-
PC	Meteornet* [61]	-	-	-	-	0.209	-
	PointPWC [8]	-	-	0.112	51.8%	-	-
	SCTN [9]	-	-	0.038	84.7%	-	-
RGB+Depth	RAFT-3D [10]	2.37	**87.1**%	0.062	84.5%	0.089	71.1%
RGB+PC	DeepLiDARFlow [12]	6.04	47.1%	-	27.2%	-	-
	CamLiFlow [13]	2.20	84.6%	0.033	91.7%	0.059	86.0%
RGB+Event	RAFT+Event	2.49	84.6%	-	-	-	-
RGB+PC+Event	CamLiFlow+Event	1.56	84.4%	0.028	91.7%	0.048	85.9%
	RPEFlow (Ours)	**1.40**	86.2%	**0.024**	**93.1**%	**0.042**	**88.0**%

scenes with collision detection, gravity model and ambient illumination and has more object kinds than FlyingThings3D. We also use the DSEC [22] dataset, which contains 8,170 pairs of real-captured samples in driving scenarios.

Training, Hyper-parameters and Metrics. Our model is trained with PyTorch on four RTX3090 GPUs and evaluated on one. We use the Adam optimizer with weight decay 10^{-6}. The number of event bins is $B = 10$, pyramid layers is $L = 5$. Loss weights are $\alpha = 10.0$, $\beta = 0.01$, $\lambda_l = 2^{(l-2)}$ for $l \in [1, L]$. Following [10, 13], we evaluate using 2D and 3D end-point error ($\mathrm{EPE}_{\mathrm{2D}}$ and $\mathrm{EPE}_{\mathrm{3D}}$), and $\mathrm{ACC}_{\mathrm{1px}}$ and $\mathrm{ACC}_{.05}$ to measure the portion of accuracy within 1 pixel and 5cm. The scene flow metrics with $\mathrm{N.Occ}$ superscript indicates only the not occluded positions are calculated, while Full or none indicates all positions including occlusion. More details about datasets and training are provided in the supplementary materials.

Table 2: **Finetuned performance comparison** on the test split of our simulated EKubric dataset.

Input	Method	EPE$_{\text{2D}}$	ACC$_{\text{1px}}$	EPE$_{\text{3D}}^{\text{N.Occ}}$	ACC$_{.05}^{\text{N.Occ}}$	EPE$_{\text{3D}}^{\text{Full}}$	ACC$_{.05}^{\text{Full}}$
RGB	RAFT [2]	0.757	93.70%	-	-	-	-
	FlowFormer [3]	0.683	93.92%	-	-	-	-
RGB+Depth	RAFT-3D [10]	0.715	94.33%	0.016	95.20%	0.049	92.62%
RGB+PC	CamLiFlow [13]	0.761	95.00%	0.009	98.39%	0.032	94.90%
RGB+Event	RAFT+Event	0.487	95.25%	-	-	-	-
RGB+PC+Event	CamLiFlow+Event	0.505	95.41%	0.008	98.48%	0.031	95.01%
	RPEFlow (Ours)	**0.442**	**96.08%**	**0.007**	**98.68%**	**0.027**	**95.30%**

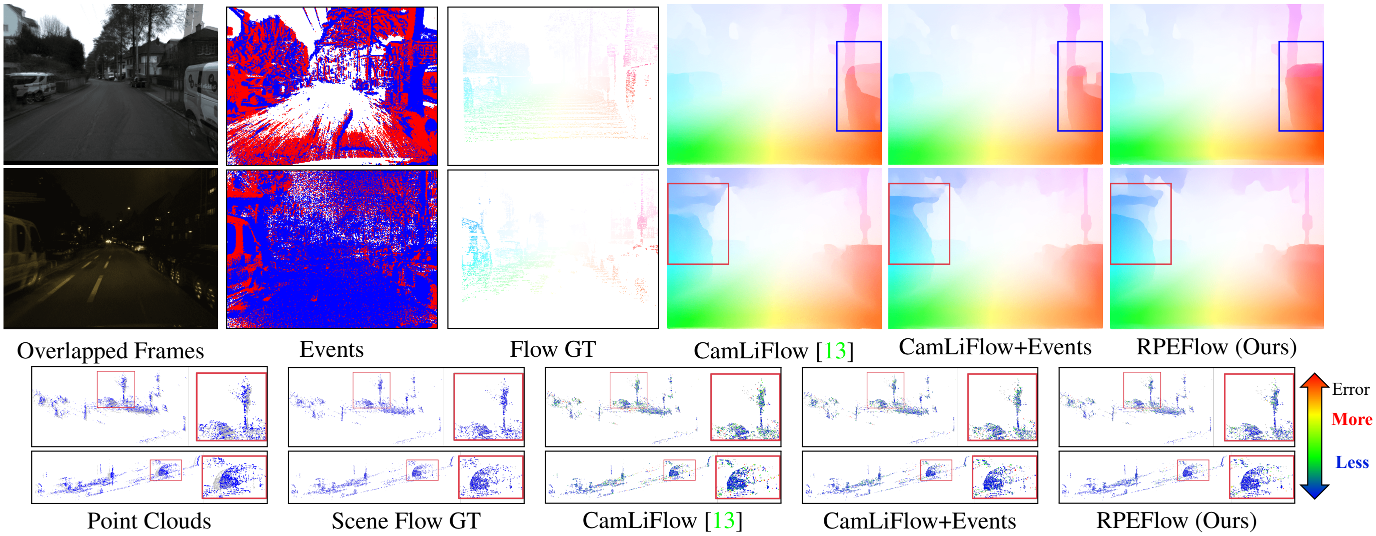
Overlapped Frames
Events
Flow GT
CamLiFlow [13]
CamLiFlow+Events
RPEFlow (Ours)
Point Clouds
Scene Flow GT
CamLiFlow [13]
CamLiFlow+Events
RPEFlow (Ours)

Figure 4: **Visual comparisons** on real-captured data, *i.e.* the "val" split of DSEC [22] dataset.

4.2. Comparisons with Synthetic Events

Following the conventional setting, our model is first pre-trained on FlyingThings3D [28], and Table 1 shows the quantitative comparison performance on the "val" split. Both results are evaluated using their pre-trained model on FlyingThings3D. Meteornet* [61] has not released model for scene flow estimation, so we compare the EPE3D result from its paper. We also finetune them on our simulated EKubric dataset, and the performance comparisons on the test set are shown in Table 2, where the compared models were all pre-trained on FlyingThings3D and then finetuned on EKubric with the same training settings. Note that we did not evaluate the PC-only methods on our EKubric dataset due to their different data preprocessing strategies.

We report the results of two representative methods [2, 8] and two latest methods [3, 9] using only one modal for one task, and find that combining multiple modalities can significantly improve model accuracy for both optical flow and scene flow estimation. Furthermore, the methods that combine all three modalities, *i.e.* RGB+PC+Event, achieve better results than the rest [10, 12, 13], which illustrates the significant benefit of introducing event data for accurate motion estimation. With the same input setting, we compare our model with CamLiFlow+Event and observe improved accuracy, which is because our proposed multimodal attention fusion module is able to fully mine valuable information from continuous event data for accurate motion estimation. The comparison of visualization results in Fig. 3 and in the supplementary materials is also consistent with the above observations, especially in high dynamic and detail moving or motion-blurred areas.

4.3. Comparisons with Real Events

We further conduct experiments on the real-captured DSEC [22] dataset. We divide the public set into "train" and "val" splits for finetuning and evaluation. Since the dense depth required by RAFT-3D [10] is not available, we use CFNet [65] to obtain a pseudo-dense depth map by stereo matching as input. E-RAFT* [22] does not have a publicly available training code, thus we can only use the pretrained model on the entire DSEC dataset including the "val" split (marked with *). Both quantitative and qualitative (Table 3 and Fig. 4) comparisons demonstrate the superiority of our

Table 3: **Finetuned Performance comparison** on the “val” split of DSEC [22] dataset.

Method	EPE_{2D}	ACC_{1px}	$\text{EPE}_{\text{3D}}^{\text{Full}}$	$\text{ACC}_{.05}^{\text{Full}}$
RAFT [2]	0.572	89.63%	-	-
E-RAFT* [22]	0.473	92.11%	-	-
RAFT-3D [10]	0.567	90.55%	0.140	51.30%
CamLiFlow [13]	0.383	94.92%	0.120	53.49%
RAFT+Event	0.537	90.08%	-	-
CamLiFlow+Event	0.361	95.07%	0.116	55.26%
RPEFlow (Ours)	**0.332**	**95.27%**	**0.104**	**60.50%**

Table 4: **Ablation Studies.** The results validate the effectiveness for introducing event data, attention fusion and mutual information regularization, respectively.

	Event	Fusion	MI	EPE_{2D}	ACC_{1px}	$\text{EPE}_{\text{3D}}^{\text{Full}}$	$\text{ACC}_{.05}^{\text{Full}}$
(a)	-	Concat	-	2.200	84.62%	0.059	86.02%
(b)	-	Attention	-	2.133	84.74%	0.058	86.53%
(c)	-	Attention	✓	2.067	84.92%	0.055	86.85%
(d)	✓	$\text{Concat}_{\text{W/o E}}$	-	1.561	84.37%	0.048	85.86%
(e)	✓	$\text{Concat}_{\text{W/ E}}$	-	1.519	85.34%	0.046	86.63%
(f)	✓	Attention	-	1.494	86.01%	0.043	87.39%
(g)	✓	Attention	✓	**1.402**	**86.22%**	**0.042**	**88.01%**

model for real-captured data, which is consistent with the observations on synthetic datasets above. In particular, our method performs better in night driving (the 2nd sample in Fig. 4), because the event camera is still sensitive to brightness changes even in low-light scenes. These comparisons further illustrate the importance of introducing events and the applicability of our model to practical needs.

4.4. Ablation Studies

In Table 4, We conduct ablation experiments to verify the contribution of each component in our model. All variations are trained on the “training” split and evaluated on the “val” split of FlyingThings3D [28] dataset. In addition to the following discussion, in the supplementary materials we analyze the impact of real and simulated events and the role of combining the two tasks.

Event data. As we are the first to introduce the event data for joint optical flow and scene flow estimation, we explore the effectiveness of event data in two aspects. Firstly, we remove the event data in our framework (see Table 4 (a)-(d) and (c)-(g)), leading to significant performance degradation, especially in optical flow error. This is in line with our claim that event data with continuous observations of scene brightness changes can provide significant help for accurate motion estimation. Secondly, in order to validate the benefit of introducing events to other methods, we concatenate the extracted event feature as an additional input of flow decoder for RAFT [2] and CamLiFlow [13], denoted as RAFT+Event and CamLiFlow+Event, respectively. Results in Table 1, 2 and 3 show that introducing event data to existing models can also significantly improve the performance, rather than just on our proposed framework.

Setting	EPE_{2D}	ACC_{1px}	$\text{EPE}_{\text{3D}}^{\text{N.Occ}}$	$\text{ACC}_{.05}^{\text{N.Occ}}$	$\text{EPE}_{\text{3D}}^{\text{Full}}$	$\text{ACC}_{.05}^{\text{Full}}$
2D only	1.937	83.17%	-	-	-	-
3D only	-	-	0.043	86.37%	0.078	79.05%
2D&3D	**1.402**	**86.22%**	**0.024**	**93.14%**	**0.042**	**88.01%**

Table 5: **Joint vs independent tasks.** These models both use three modalities and the results validate that combining the two tasks makes better use of motion information.

Fusion Structure. Unlike CamLiFlow [13] that concat the features from two modalities, we propose an attention-based multimodal fusion module. We conduct experiments with both no-event and with event settings (see Table 4 (a)-(b) and (d)(e)-(f)), and our proposed attention fusion improves both 2D and 3D motion estimation performance. This shows that our multimodal attention fusion module can effectively explore the complementary information and correlations among the three modalities and generate fused features that are suitable for subsequent motion estimation.

Mutual Information. The purpose of the mutual information regularization term is to explicitly constrain the network to learn complementary information from different modalities of data. To verify its contribution to motion estimation, we conduct experiments with both no-event and with event settings (see Table 4 (b)-(c) and (f)-(g)). The results show that with this explicit constraint, the fused features can further improve the accuracy of motion estimation. In addition, we believe that the multimodal attention fusion module and mutual information regularization term both play an important role in motion estimation, because their contribution to 2D and 3D motion estimation is significant in both the with-event and without-event settings.

4.5. Computation Cost

Our model is relatively efficient compared to the latest unimodal and multimodal methods in model size and runtime, i.e., FlowFormer [3]: 17.6M, 1.35s, SCTN [9]: 7.8M, 242ms, RAFT-3D [10]: 45M, 593ms, CamLiFlow [13]: 7.7M, 88ms (≈2s with refine) and Ours: 9.75M, 112ms (both on a single RTX3090 GPU with 1280×720 size).

4.6. Joint 2D and 3D estimation.

In previous comparisons, the methods of joint optical flow and scene flow estimation perform significantly better than independent methods. To further verify the necessity of jointing these two tasks, we compare the models that only supervise the optical flow or scene flow estimation in Table 5. Although using all three modalities for a single task (2D only and 3D only) is more beneficial than the methods using fewer modality data in Table 1 of main paper.

However, combining the two tasks in 2D and 3D benefits from the tight correlation between the two deeply supervised branches, allowing more adequate exploiting of the correlation between 2D and 3D input modalities and more accurate estimating of 2D and 3D motion jointly.

5. Conclusion

We introduce a multimodal fusion framework for joint 2D optical flow and 3D scene flow estimation by fusing RGB images, point clouds and event data. Our contributions are threefold: **1)** By incorporating event data, our new framework could handle highly dynamic scenes. **2)** We fuse representations of the three very different modalities both implicitly and explicitly through multimodal attention and mutual information regularization, respectively. **3)** We contribute a new simulation dataset to further advocate research in this direction. Our work shows that event cameras can play an important role in 2D and 3D motion estimation, and reveals the prospect of event-based 3D vision.

Limitation. Our model is not specially designed for extreme situations such as dark nights or sensor failures, and we plan to address them in the future.

6. Acknowledgments

This research was supported in part by National Natural Science Foundation of China (62271410), the Fundamental Research Funds for the Central Universities, and Zhejiang Lab (NO.2021MC0AB05). Zhexiong Wan is sponsored by the scholarship from China Scholarship Council (202306290193) and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (CX2023013).

References

[1] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018. 1, 2

[2] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 402–419, 2020. 1, 2, 6, 7, 8

[3] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A transformer architecture for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 668–685, 2022. 1, 2, 6, 7, 8

[4] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7396–7405, 2020. 1, 2

[5] Vitor Guizilini, Kuan-Hui Lee, Rareş Ambruş, and Adrien Gaidon. Learning optical flow, depth, and scene flow without real-world labels. *IEEE Robotics and Automation Letters*, 7(2):3491–3498, 2022. 1, 2

[6] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *European Conference on Computer Vision (ECCV)*, pages 614–630, 2018. 1, 2

[7] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3614–3622, 2019. 1, 2

[8] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *European Conference on Computer Vision (ECCV)*, pages 88–107. Springer, 2020. 1, 2, 3, 6, 7

[9] Bing Li, Cheng Zheng, Silvio Giancola, and Bernard Ghanem. SCTN: Sparse convolution-transformer network for scene flow estimation. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, pages 1254–1262, 2022. 1, 2, 6, 7, 8

[10] Zachary Teed and Jia Deng. RAFT-3D: Scene flow using rigid-motion embeddings. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8375–8384, 2021. 1, 2, 6, 7, 8

[11] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1334–1343, 2020. 1, 2

[12] Rishav Rishav, Ramy Battrawy, René Schuster, Oliver Wasenmüller, and Didier Stricker. Deeplidarflow: A deep learning architecture for scene flow estimation using monocular camera and sparse lidar. In *IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)*, pages 10460–10467. IEEE, 2020. 1, 2, 6, 7

[13] Haisong Liu, Tao Lu, Yihui Xu, Jia Liu, Wenjie Li, and Lijun Chen. Camliflow: Bidirectional camera-lidar fusion for joint optical flow and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5791–5801, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 1

[15] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015. 1, 2

[16] Lorenzo Porzi, Markus Hofinger, Idoia Ruiz, Joan Serrat, Samuel Rota Bulo, and Peter Kontschieder. Learning multi-object tracking and segmentation from automatic annotations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6846–6855, 2020. 1

[17] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3876–3886, 2021. 1

[18] Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura, and Lei Zhang. Flowfusion: Dynamic dense rgb-d slam based on optical flow. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7322–7328. IEEE, 2020. 1

[19] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6498–6508, 2021. 1

[20] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(1):154–180, 2022. 1

[21] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 989–997, 2019. 1, 2

[22] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *International Conference on 3D Vision (3DV)*, pages 197–206, 2021. 1, 2, 6, 7, 8

[23] Min Liu and Tobias Delbrück. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In *British Machine Vision Conference (BMVC)*, page 280, 2018. 1, 2

[24] Liyuan Pan, Miaomiao Liu, and Richard Hartley. Single image optical flow estimation with an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1669–1678, 2020. 1, 2

[25] Zhexiong Wan, Yuchao Dai, and Yuxin Mao. Learning dense and continuous optical flow from an event camera. *IEEE Transactions on Image Processing (TIP)*, 31:7237–7251, 2022. 1, 2

[26] 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 Robotics and Automation Letters*, 3(3):2032–2039, 2018. 2

[27] Ricky T. Q. Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2, 4

[28] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016. 2, 5, 6, 7, 8

[29] 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 *IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 2

[30] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017. 2

[31] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8121–8130, 2022. 2

[32] Sundar Vedula, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(3):475–480, 2005. 2

[33] Frédéric Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2007. 2

[34] Fabian Brickwedde, Steffen Abraham, and Rudolf Mester. Mono-sf: Multi-view geometry meets single-view depth for monocular scene flow estimation of dynamic traffic scenes. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2780–2790, 2019. 2

[35] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Flot: Scene flow on point clouds guided by optimal transport. In *European Conference on Computer Vision (ECCV)*, pages 527–544. Springer, 2020. 2

[36] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3867–3876, 2018. 2

[37] Sio-Hoi Ieng, João Carneiro, and Ryad B Benosman. Event-based 3d motion flow estimation using 4d spatio temporal subspaces properties. *Frontiers in Neuroscience*, 10:596, 2017. 2

[38] Chankyu Lee, Adarsh Kumar Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy. Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks. In *European Conference on Computer Vision (ECCV)*, pages 366–382, 2020. 2

[39] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 884–892, 2016. 2

[40] Matteo Poggi, Filippo Aleotti, and Stefano Mattoccia. Sensor-guided optical flow. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7908–7918, 2021. 2

[41] Andrea Conti, Matteo Poggi, Filippo Aleotti, and Stefano Mattoccia. Unsupervised confidence for lidar depth maps and applications. In *IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)*, 2022. 2

[42] Peng Xu, Xiatian Zhu, and David A Clifton. Multimodal learning with transformers: A survey. *arXiv preprint arXiv:2206.06488*, 2022. 2

[43] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. Attention-based multimodal fusion for video description. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4193–4202, 2017. 2

[44] Xi Wei, Tianzhu Zhang, Yan Li, Yongdong Zhang, and Feng Wu. Multi-modality cross attention network for image and sentence matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10941–10950, 2020. 2

[45] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool. Event-based fusion for motion deblurring with cross-modal attention. In *European Conference on Computer Vision (ECCV)*, pages 412–428, 2022. 2

[46] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1798–1828, 2013. 2

[47] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International Conference on Machine Learning (ICML)*, pages 1558–1567, 2017. 2

[48] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning (ICML)*, pages 531–540, 2018. 2

[49] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Xin Yu, Yiran Zhong, Nick Barnes, and Ling Shao. Rgb-d saliency detection via cascaded mutual information minimization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4338–4347, 2021. 2

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 2, 3

[51] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 3, 4

[52] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[53] David Barber and Felix V. Agakov. The im algorithm: A variational approach to information maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 201–208, 2003. 2

[54] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International Conference on Machine Learning (ICML)*, pages 1779–1788, 2020. 2, 4

[55] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[56] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5728–5739, 2022. 4

[57] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations (ICLR)*, 2016. 4

[58] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2017. 4

[59] Diederik Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 4

[60] R.W. Yeung. A new outlook on shannon's information measures. *IEEE Trans. on Information Theory*, 37(3):466–474, 1991. 4

[61] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. Meteornet: Deep learning on dynamic 3d point cloud sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9246–9255, 2019. 6, 7

[62] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–537, 2019. 5

[63] Daniel Gehrig, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Video to events: Recycling video datasets for event cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3586–3595, 2020. 5

[64] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3749–3761, 2022. 5

[65] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13906–13915, 2021. 7