DERD-Net: Learning Depth from Event-based Ray Densities

Diego Hitzges[*1] **Suman Ghosh**[*1] **Guillermo Gallego**[1,2]

[1]Technische Universität Berlin, Einstein Center Digital Future, Robotics Institute Germany
[2]Science of Intelligence Excellence Cluster, Germany

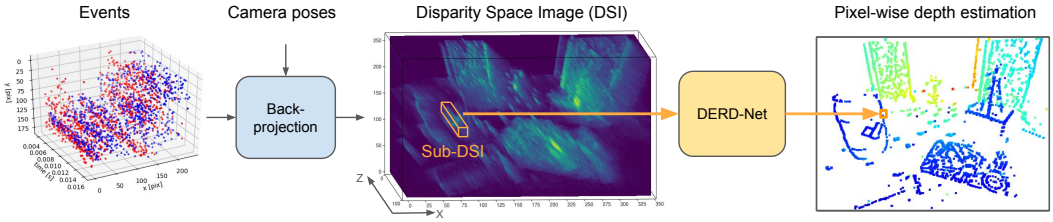

Figure 1: *Overview*. We present a deep-learning–based method to predict depth from event-ray densities (Disparity Space Images –DSIs) obtained by back-projecting events using camera poses. Our deep neural network, DERD-Net, operates in parallel on local volumetric neighborhoods of the DSI data, called Sub-DSIs (in orange).

Abstract

Event cameras offer a promising avenue for multi-view stereo depth estimation and Simultaneous Localization And Mapping (SLAM) due to their ability to detect blur-free 3D edges at high-speed and over broad illumination conditions. However, traditional deep learning frameworks designed for conventional cameras struggle with the asynchronous, stream-like nature of event data, as their architectures are optimized for discrete, image-like inputs. We propose a scalable, flexible and adaptable framework for pixel-wise depth estimation with event cameras in both monocular and stereo setups. The 3D scene structure is encoded into disparity space images (DSIs), representing spatial densities of rays obtained by back-projecting events into space via known camera poses. Our neural network processes local subregions of the DSIs combining 3D convolutions and a recurrent structure to recognize valuable patterns for depth prediction. Local processing enables fast inference with full parallelization and ensures constant ultra-low model complexity and memory costs, regardless of camera resolution. Experiments on standard benchmarks (MVSEC and DSEC datasets) demonstrate unprecedented effectiveness: (i) using purely monocular data, our method achieves comparable results to existing *stereo* methods; (ii) when applied to stereo data, it strongly outperforms all state-of-the-art (SOTA) approaches, reducing the mean absolute error by at least 42%; (iii) our method also allows for increases in depth completeness by more than 3-fold while still yielding a reduction in median absolute error of at least 30%. Given its remarkable performance and effective processing of event-data, our framework holds strong potential to become a standard approach for using deep learning for event-based depth estimation and SLAM. Project page: https://github.com/tub-rip/DERD-Net

[*]Equal contribution

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Depth estimation is a fundamental task in computer vision, with key applications in areas such as robotics, autonomous driving, and augmented reality. Traditional stereo vision techniques rely on synchronized cameras to capture images and infer depth by finding correspondences between them. However, these methods often struggle in low-light and fast-motion conditions. Moreover, conventional cameras produce large amounts of redundant data by capturing entire images at fixed intervals, leading to inefficiencies in both data storage and processing.

Unlike conventional cameras, event cameras operate asynchronously, detecting per-pixel brightness changes (called "events") [1–3]. This provides high temporal resolution and robustness to motion blur, making them well suited for dynamic scenes. Their sparse output enables efficient processing of only relevant areas, making them ideal for real-time tasks such as visual odometry (VO) / SLAM.

Harnessing deep learning for depth estimation with event cameras has the potential to transform such applications. However, adapting neural networks to event data remains challenging due to its asynchronous stream-like nature. Moreover, the scarcity of event camera datasets with ground truth depth [4, 5] results in limited training data, which can lead to overfitting [6]. While simulating data is one way to address this issue, generalizing models trained on simulation to real-world scenarios is far from trivial due to differences in data distribution [7–10]. Thus, instead of directly processing events (which is prone to overfitting and extensive training time) we rely on intermediate event representations informed by the scene geometry.

One promising approach for 3D reconstruction is back-projecting events as rays into space and capturing their intersection densities as disparity space images (DSIs) [11]. DSIs from two or more cameras can be fused, eliminating the need for event synchronization between cameras. This reduces complexity and allows for more robust depth estimation. The Multi-Camera Event-based Multi-View Stereo (MC-EMVS) method [12] recently produced state-of-the-art (SOTA) results, outperforming other techniques in depth benchmarks across several metrics. To obtain pixel-wise depth estimates from a DSI, MC-EMVS selects the disparity level with the highest ray count, effectively using an argmax operation. Ray counting is used as a proxy for finding 3D edges where the rays intersect. A selective threshold filter is applied to predict depth only for pixels with sufficient ray counts.

While straightforward, this approach does not fully utilize the potential of DSIs. It is susceptible to noise and cannot effectively extract cues from surrounding pixels and more complex patterns across disparity levels. Consequently, it leads to fewer pixels obtaining depth estimation and less accurate depth predictions than the DSI might potentially allow for. Therefore, we need more effective approaches for extracting depth from DSIs, that are more reliable, accurate, and produce more depth estimates, by adequately recognizing complex ray intersection patterns.

Our Contribution. We propose a novel deep learning framework for event-based depth estimation that is optimized for SLAM scenarios and addresses the aforementioned limitations. An overview is illustrated in Fig. 1. Our approach estimates pixel-wise depth from a DSI using a neural network with 3D convolutions and a recurrent structure. The framework is directly applicable to both monocular and stereo settings. Our key contributions include:

- **Learning-based Local Processing**: For each selected pixel, a small local subregion of the DSI (Sub-DSI) is used as input to the neural network. This novel design leverages the inherent sparsity of event data and DSIs to efficiently process and produce only relevant information for sparse depth-related tasks. (Sec. 3).
- **Enhanced Data Utilization**: Our model captures complex patterns within the DSI, increasing depth prediction accuracy and the number of pixels for which depth can be reliably estimated. Limiting the input to small local subregions around selected pixels enhances generalization by preventing the network from overfitting to dynamics specific to the training scenes and augmenting the available training set, as each Sub-DSI serves as an individual data instance.
- **Efficiency and Scalability**: By adopting our Sub-DSI approach, we obtain small independent inputs of fixed size. This enables full parallelization and provides an ultra-light network that has the ability to handle *any camera resolution* with constant very short inference time. Our network architecture allows the processing of DSIs of variable depth resolution. (Sec. 3.2).
- **Comprehensive Experiments**: We evaluate our model on both monocular and stereo data from the standard datasets MVSEC [4] and DSEC [13] using cross-validation. It outperforms the

state of the art by a large margin on ten figures of merit. Even using monocular data, our model achieves performance comparable to SOTA methods that require stereo data (Sec. 4). We show downstream applicability and robustness to imperfect (noisy) camera poses.

To the best of our knowledge, our work is the first learning-based multi-view stereo method to (i) use camera poses along with events as input, which is crucial for accurate depth estimation over long intervals (as used in SLAM [14, 15]); (ii) demonstrate successful and robust depth prediction on real-world event data from DSIs; (iii) report good generalization on all *three* MVSEC *indoor flying* sequences [5], even when compared to multi-modal methods that combine stereo intensity frames with events. We provide code, trained models and video results for clarity and reproducibility.

2 Related Work

Stereo depth estimation with event cameras has been a captivating problem since the invention of the first event camera by Mahowald and Mead in the 1990s [3, 5, 16] due to their potential for high temporal resolution and robustness to motion blur. Recent approaches have addressed stereo event-based 3D reconstruction for VO and SLAM [14, 17–22]. These methods assume a static world and known camera motion, using this information to assimilate events over longer time intervals, thereby increasing parallax and producing more accurate semi-dense depth maps. A comprehensive review is provided in [5].

MC-EMVS [12] introduced a novel stereo approach for depth estimation which does not require explicit data association, using DSIs generated from stereo events cameras. By leveraging the sparsity of events and fusing back-projected rays, they outperformed the event-matching–based solution of [19] and thereby achieved SOTA results in 3D reconstruction and VO [14]. Evidently, this DSI-based 3D reconstruction is robust to imperfect poses estimated using an event camera tracking method. We advance this approach by employing a compact neural network specialized to derive explicit depth from the DSIs, creating a standardized and effective framework for processing event-based data in deep learning applications that does not rely on event simultaneity or matching.

Deep Learning for depth estimation from event data. Deep learning has significantly advanced depth estimation in traditional monocular and stereo camera setups, achieving remarkable results [23–25]. However, its application to event camera data remains relatively limited due to the sparse asynchronous nature of event streams, which require specialized frameworks [5]. For example, in monocular vision, [26] uses synthetic data on a recurrent network to capture temporal information from grid-like event inputs. Yet, mismatches between synthetic and real data degrade performance [27], and monocular depth estimation from events is an ill-posed problem, making high accuracy challenging to achieve with this learning-based framework [28].

For stereo depth estimation, [6, 29] present two pioneering studies. Specifically, DDES [6] introduced the first deep-learning–based supervised stereo-matching method, while [29] proposed the first unsupervised learning framework. Both methods use First-In First-Out queues to store events at each position, allowing for concurrent time and polarity reservation. Nevertheless, high event rates lead to greater processing demands and, consequently, increased model complexity and memory requirements, limiting the use of visual cues from both event cameras. Our method overcomes these challenges by maintaining constant input dimensions defined by the size of the DSI subregion around a selected pixel, regardless of the event rate. It thereby provides a significant advancement in applying deep learning to event-based depth estimation on real-world data.

3 Methodology

In this section, we present our supervised-learning–based approach and the related framework in detail, for which a general overview is provided in Fig. 1. We describe the preprocessing of the data, the architecture of our network (shown in Fig. 2) and the training and inference procedures.

3.1 Framework

As an event camera with $W \times H$ pixels moves through a scene, it triggers events $e_k = (x_k, y_k, t_k, \pm_k)$ and produces a near continuous-time stream of data $\mathcal{E} = \{e_k\}$. Following [11, 12], this stream is

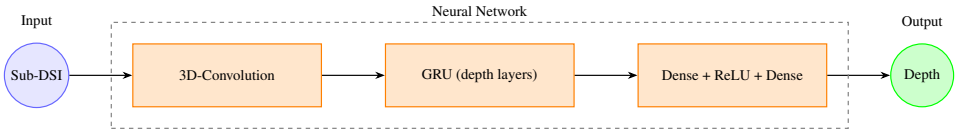

Figure 2: Network Architecture. The parameters of the network's modules are specified in Tab. 1.

Table 1: Details of network layers. K, P, and S stand for kernel-size, padding and stride.

Layer	**Dimensions**	**Details**
Sub-DSI (Input)	$100 \times 1 \times 7 \times 7$	Depth × Channels × Width × Height
3D-Convolution	$50 \times 4 \times 5 \times 5$	K = (3,3,3); P = (1,0,0); S = (2,1,1)
ReLU + Flatten	$50 \times (4 \cdot 5 \cdot 5)$	Flattens channels and frame
GRU	1×100	Selects final hidden state h_{50}
Dense + ReLU	100	Maintains dimension
Dense (Output)	1 or 3×3	Outputs depth value(s)

sliced into time intervals. For every time interval, a DSI is created and associated with a camera viewpoint (called Reference Viewpoint) as follows: given camera poses, all events e_k in the interval are back-projected into 3D space by casting rays from the moving camera optical center through the corresponding pixel (x_k, y_k). The depth axis is discretized into D levels, equidistant in inverse linear space, resulting in a 3D DSI of size $D \times W \times H$ voxels, whose values represent the number of rays passing through each region (i.e., voxel) of space (as shown in Fig. 1). Although input poses are required for building a DSI, they can be obtained from tracking methods or dead-reckoning [12,14,30].

We consider a stereo setup with two synchronized cameras providing two perspectives of the same scene. By leveraging parallax, this configuration enhances depth perception and allows for more accurate 3D reconstruction. For each interval, we construct two DSIs (one for each camera) and fuse them by applying voxel-wise metrics (e.g., harmonic mean) as described in [12]. To compare performance, we also apply our approach to the monocular data of the left camera only.

Since DSIs are typically large and sparse, depth is estimated only for pixels with sufficient information. A confidence map is generated by projecting the DSI onto a 2D grid of size $W \times H$, where each pixel's value represents the maximum ray density among all depth levels [12] (called pixel selection map in Fig. 3). An adaptive Gaussian threshold (AGT) filter is then applied to this grid to select the pixels $\{p_1, \ldots, p_n\}$ with a sufficient maximum ray density for reliable depth estimation. For each selected pixel $p_i = (x_i, y_i)$, a surrounding subregion $\tilde{S}_i$ is extracted from the DSI, including the ray counts of all pixels within L1 radii of r_W, r_H:

$$\tilde{S}_i \doteq \text{DSI}[:,\ x_i - r_W : x_i + r_W,\ y_i - r_H : y_i + r_H].$$

Each subregion $\tilde{S}_i$ is then normalized individually,

$$S_i \doteq \tilde{S}_i \,/\, \max(\tilde{S}_i) \in [0,1]^{D \times (2r_W+1) \times (2r_H+1)}, \tag{1}$$

and serves as input to our neural network. Using only small subregions of the DSIs as inputs leads to an efficient and compact architecture operating independently of camera resolution. Furthermore, it reduces the risk of overfitting by encouraging the model to learn generalizable patterns of *ray intersections* within the Sub-DSIs instead of memorizing semantics and dynamics specific to the training scene. Such localized input processing is possible because DSIs consolidate sparse events into a structured format that preserves geometric information within small spatial regions.

The depth estimates $z_1, \ldots, z_n$ are computed in parallel. Since the amount of selected pixels can be controlled by the AGT filter and the dimensions of the Sub-DSIs are fixed, we achieve constant low model complexity and memory costs, regardless of the number of triggered events.

3.2 Network Architecture

The architecture of the neural network is illustrated in Fig. 2, with the dimensions of each layer listed in Tab. 1. The network receives the normalized Sub-DSI (1) as input. The objective is to capture local geometrical patterns in the Sub-DSI to extract more relevant depth information than the SOTA argmax approach used in [11, 12]. Since established networks like U-Net [31] often include strong spatial

Table 2: Hyperparameters.

Dataset	Dataset Details			DSI Parameters					Gauss. Filter		Training Process				
	Sequences	Resolution	LiDAR Δt	Span	$z_{\min}$	$z_{\max}$	D	Sub-DSI	Window	C	Batch	Optimizer	LR	LF	Epochs
MVSEC	Indoor flying	346 × 260 px	50 ms	1 s	1 m	6.5 m	100	7×7	9×9	−10	64	AdamW	10^{-3}	MAE	3
DSEC	Zurich04a	640 × 480 px	100 ms	0.2 s	4 m	50 m	100	7×7	9×9	−2	64	AdamW	10^{-3}	MAE	3

compression and Transformers tend to impose high data demands for reliable generalization [32], we tailor an ultra-lightweight custom architecture that shares design elements with FireNet [33], adapted for very small frame sizes yet variable depth dimensions. Similarly to RAFT-Stereo [34], we adopt convolutions and a Gated Recurrent Unit (GRU) to efficiently handle different depth resolutions, enabling customization of the desired depth precision without modifying architecture.

First, to capture local patterns, further reduce input size and avoid overfitting, we apply a 3D convolutional filter (3D-Conv) with a ReLU activation, a kernel size of $3 \times 3 \times 3$ and no padding in the spatial dimensions (width and height). For the depth dimension, we set a padding of 1 and a stride of 2 to halve the number of depth layers. We use 4 output channels to capture different patterns simultaneously, resulting in the convolved version of the Sub-DSI (1):

$$S_i^* = \text{3D-Conv}(S_i) \in \mathbb{R}^{\frac{D}{2} \times 4 \times (2r_W - 1) \times (2r_H - 1)}. \tag{2}$$

To create the DSIs, rays were cast from the representative camera location into space, passing sequentially through the different depth levels. Since mapping precision requirement may change from scene to scene, we need to deal with variable depth resolution of the DSI. To efficiently and flexibly model this interdependence of consecutive depth layers for a *variable* D, the convolved depth layers are flattened and successively fed into a GRU [35]. The recurrent structure of the GRU allows us to maintain a constant ultra-low count of only 70k parameters in total. It iteratively embeds each depth layer's information within the context of the previous layers, producing hidden state representations $h_1, \ldots, h_{D/2}$. We then proceed with the final hidden state $h_{D/2}$, which condenses the relevant information from all depth layers along the depth axis:

$$h_{\frac{D}{2}} = \text{GRU}(S_i^*) \in \mathbb{R}^{4 \cdot (2r_W - 1) \cdot (2r_H - 1)}. \tag{3}$$

Finally, a dense layer that preserves the dimension of $h_{D/2}$ with ReLU activation and a subsequent output dense layer are applied to process the hidden state. We introduce two versions of the network for custom modification of depth estimation density. In the single-pixel version, the network predicts the normalized depth for the selected pixel $z_i \in [0, 1]$, while in the multi-pixel version, the output is a 3×3 grid $Z_i \in [0, 1]^{3 \times 3}$, representing the normalized depth predictions of the central pixel and its 8 neighbors. Finally, normalized depth is converted into actual depth by mapping $[0, 1]$ to $[z_{\min}, z_{\max}]$.

3.3 Training and Inference

As supervised loss function for training the neural network model we use the *mean absolute error (MAE)*, with given ground truth depth (this is the case of standard real-world datasets used, such as MVSEC and DSEC – see Sec. 4). To reduce training time and further improve generalization, we additionally employ ensemble learning (EL) [36, 37]. For training, we initialize two identical but independent instances of our neural network with different random weights. The training set is split into two disjoint subsets, enabling parallel training. During testing or inference, each Sub-DSI S_i is processed simultaneously by both networks, and the final depth estimation is obtained by averaging the individual predictions. This helps reduce variance in the predictions, leading to more stable and accurate results. We also present results without EL in Tabs. 11 to 14 in the Appendix Sec. B.

4 Experiments

In this section, we evaluate the performance and reliability of the proposed depth estimation approach. Following prior protocols, we conduct experiments on the MVSEC [4] and the DSEC [13] datasets. Ground truth (GT) depth, captured at fixed intervals using LiDAR sensors, serves as reference locations for constructing the respective DSIs over a defined time span. Pixel selection for depth estimation is based on an AGT filter, where the window size determines the surrounding pixel count considered, and a constant C is subtracted from the observed ray count.

Table 3: Summarized quantitative comparison of the proposed methods with the state of the art. *MVSEC indoor_flying* and *DSEC Zurich_City_04_a*. The full comparison over ten metrics is in the Appendix Sec. B.

	Method		MVSEC				DSEC			
	Algorithm	Modality	Mean Err [cm] ↓	Median Err [cm] ↓	bad-pix [%] ↓	#Points [million] ↑	Mean Err [m] ↓	Median Err [m] ↓	bad-pix [%] ↓	#Points [million] ↑
SOTA	EMVS [11]	monocular + F_{orig}	33.78	14.35	3.84	1.27	5.64	2.52	13.68	1.31
SOTA	EMVS [11]	monocular + F_{denser}	50.32	20.81	11.46	4.15	7.01	3.56	24.33	6.09
SOTA	ESVO [19]	stereo	22.70	9.83	2.83	1.56	3.93	1.62	10.54	<u>9.40</u>
SOTA	SGM [38]	stereo	35.42	12.35	6.39	**14.46**	6.74	1.58	15.25	8.30
SOTA	GTS [39]	stereo	389.00	45.43	38.45	0.06	26.24	1.62	32.56	0.11
SOTA	MC-EMVS [12]	stereo + F_{orig}	20.07	9.53	1.35	0.81	3.27	0.90	10.75	1.25
SOTA	MC-EMVS [12]	stereo + F_{denser}	28.38	12.38	3.26	2.77	4.76	1.56	17.42	4.64
SOTA	MC-EMVS [12] + MF	stereo + F_{orig}	20.64	9.72	1.43	3.00	3.51	0.96	11.81	3.83
Ours	DERD-Net	monocular + F_{orig}	23.68	11.55	2.78	1.21	3.12	1.60	5.50	2.10
Ours	DERD-Net	monocular + F_{denser}	28.52	13.85	4.87	4.15	3.01	1.50	6.35	6.09
Ours	DERD-Net	stereo + F_{orig}	**11.69**	**5.50**	**0.89**	0.79	<u>1.61</u>	**0.46**	<u>4.12</u>	1.67
Ours	DERD-Net	stereo + F_{denser}	15.24	6.68	1.70	2.77	1.80	0.54	5.04	4.64
Ours	DERD-Net (multi-pixel)	stereo + F_{orig}	<u>12.02</u>	<u>5.63</u>	<u>0.90</u>	4.32	**1.59**	<u>0.47</u>	**3.81**	6.59
Ours	DERD-Net (multi-pixel)	stereo + F_{denser}	15.68	6.73	1.74	<u>11.33</u>	1.79	0.54	4.61	**14.74**

We investigate the impact of DSIs derived from both monocular and stereo settings during training and testing. Table 2 provides an overview of the key parameters used in the datasets and the training processes of the experiments. Abbreviations are: minimum depth ($z_{\min}$), maximum depth ($z_{\max}$), depth dimensions (D), filter window size (Window), subtractive constant (C), batch size (Batch), learning rate (LR), and loss function (LF).

4.1 Metrics

The performance of the networks is evaluated using *ten* standard metrics commonly employed in depth estimation tasks [12]. We calculate both mean and median errors between the estimated and GT depths, with median errors providing robustness against outliers. Additionally, we report the number of reconstructed points, reflecting the algorithm's ability to generate valid depth estimations, and the number of outliers (bad-pix [40]), representing the proportion of significant depth estimation errors. In the Appendix, we also compute the scale-invariant logarithmic error (SILog Err) to evaluate the error while considering scale, and the sum of absolute relative differences (AErrR) to assess the relative accuracy of the depth predictions. Finally, we report δ-accuracy values, which indicate the percentage of points whose estimated depth falls within specified limits relative to GT [41].

4.2 Baseline Methods

We compare our approach against several SOTA methods that have been benchmarked on the task of *long-term* event-based depth estimation [5], thus evaluated under the same input conditions (events and camera poses) and output format (semi-dense depth maps) supportive of SLAM. In the absence of other deep stereo methods that learn from input camera poses, we also include comparisons against the SOTA instantaneous end-to-end learning-based stereo methods in Sec. 4.5 for completeness. We adopt the same train-test splits established in prior work and standard benchmarks [5].

The Generalized Time-Based Stereovision (GTS) method [39] utilizes a two-step process: first performing stereo matching based on a time-consistency score for each event, followed by depth estimation through triangulation. The Semi-Global Matching (SGM) method [38] is adapted for event-based data by generating time images and subsequently applying stereo matching, with the depth map being refined by masking it at the locations of recent events. Another method, Event-based Stereo Visual Odometry (ESVO) [19] (ESVO2 [42]), integrates depth estimates by employing Student-t filters, ensuring robust spatio-temporal consistency between stereo time image patches.

The two closest baseline methods for performance comparison of our method are EMVS for monocular vision [11] and MC-EMVS for stereo vision [12]. Both methods extract pixel-wise depth from DSIs by applying the argmax function. To ensure consistency and fairness, we benchmark the methods following the procedure established in prior works [5].

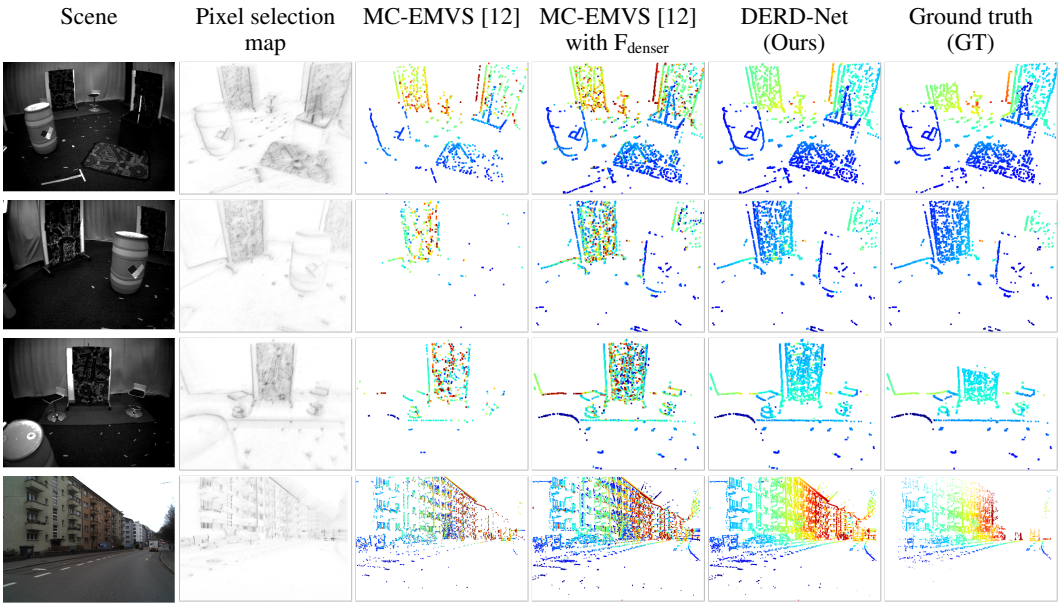

Figure 3: *Depth estimation*. Qualitative comparison of depth estimated using the MC-EMVS method [12], applying it to the new selected pixels F_{denser} and our method DERD-Net, for the MVSEC *indoor_flying* [4] (top 3 rows) and DSEC *Zurich_City_04_a* (bottom row) sequences. Ground truth depth from LiDAR is masked by pixels with valid depth estimate. Our method estimates depth even at pixels with no GT depth. Depth maps are pseudo-colored, from blue (close) to red (far), in the range 1-6.5m for MVSEC and 4-50m for DSEC.

4.3 Experiments on MVSEC Dataset

This section describes the experiments conducted on the *indoor_flying* sequences 1,2,3 of the MVSEC dataset [4] to evaluate the performance of the proposed depth estimation method. Most stereo methods do not evaluate on *indoor_flying_4* (because of noisy events from the low-texture floor as the drone flies very low) and the driving sequences (because the stereo baseline is too small for the given depth range and low camera resolution). We employed three-fold cross-validation by utilizing two sequences for supervised training and reserving the remaining sequence for testing, repeating this process for all three possible combinations of sequences to ensure robustness in our evaluation.

Two pixel-selection filter settings: F_{orig} and F_{denser}. We first trained the single-pixel version of our network on monocular DSIs to compare its performance to EMVS [11]. Subsequently, we retrained it on stereo DSIs fused via the harmonic mean and compared its performance to MC-EMVS [12]. These two baseline methods used an AGT filter F_{orig} with a window of 5×5 px and a subtractive constant of $C_{\text{orig}} = -14$. Given that our network is designed to extract additional information from the geometrical patterns within the DSI, we hypothesized that it would still produce reliable depth estimates for pixels with lower confidence. To test this hypothesis, we used a larger filter window size of 9×9 px and a subtractive constant of $C_{\text{denser}} = -10$, resulting in a less strict filter F_{denser}, enabling depth estimation for more pixels. To ensure a representative comparison, we evaluated our networks as well as EMVS and MC-EMVS on both sets of pixels created by F_{orig} and F_{denser}.

Multipixel vs. Morphological Filter. One apparent drawback of MC-EMVS is the limited number of pixels for which depth is estimated compared to other SOTA methods. To address this, [12] presented the option of adding a 4-neighbor morphological filter (MF), which dilates the depth estimation map to increase the number of depth-estimated pixels. We compare this with our framework's ability to further increase the number of depth-estimated pixels by training and evaluating the multi-pixel version of our network.

Results. The averaged results over all three sequences are displayed in the left half of Tab. 3 for all discussed modalities, while the individual results (including performance without EL) are detailed in the Appendix, in Tabs. 12 to 14. Notably, our network's performance converges after only 3 epochs

of training. This rapid convergence is particularly advantageous for future applications where the network might be trained on more heterogeneous datasets or retrained for specific scenarios.

Monocular setup. On the monocular DSIs filtered by F_{orig}, our single-pixel network achieves results comparable to those of *stereo* SOTA methods and significantly outperforms EMVS [11] by 30% in MAE. Remarkably, even on the 3.27 times larger set of pixels created by F_{denser}, it still achieves better scores than EMVS at F_{orig} for all metrics, except bad-pix. Applying EMVS to the same expanded set of pixels leads to a 76% increase in MAE compared to our framework.

Stereo setup. Applying our single-pixel network to stereo DSIs filtered by F_{denser} allows us to predict depth at significantly more pixels than any other method, except for the SGM method [38], while consistently surpassing all benchmarks across all metrics. The number of pixels increases by 242%, while the MAE and MedAE reduce by 24% and 30%, compared to MC-EMVS [12] with F_{orig}. The only exception is the bad-pix measure, where MC-EMVS performs slightly better. When both methods are compared on the same set of pixels, our approach yields a reduction in both MAE and MedAE of 42% for F_{orig} and 46% for F_{denser}, respectively. Performance remains consistent when using the multi-pixel network, which increases the amount of depth-estimated pixels by a factor of 5.47 for F_{orig} and 4.09 for F_{denser} compared to its single-pixel version, indicating it to be a superior approach to the morphological filter of MC-EMVS, which only rises the number of points by a factor of 3.70 for F_{orig}. As a consequence, the multi-pixel version of our framework estimates depth for almost as many pixels as the SGM method while delivering new SOTA results.

Qualitative comparison. To further illustrate the effectiveness of our method, Fig. 3 compares depth maps generated by our single-pixel network against those produced by SOTA method MC-EMVS. Our network not only provides a denser depth estimation, which improves the recognition of contours, but also effectively eliminates the visible outliers produced by MC-EMVS. This improvement is evident when comparing our method to MC-EMVS applied both to the expanded and the original set of pixels, highlighting the robustness and superiority of our approach.

4.4 Experiments on DSEC Dataset

To assess the applicability of our network architecture to different data, we retrained and tested it on DSIs obtained from a stereo setting in the DSEC dataset [13]. This dataset presents unique challenges due to its outdoor driving scenarios, which differ significantly from the indoor environments of the MVSEC dataset, its higher spatial resolution (640×480 px) and different noise characteristics (Prophesee camera vs. DAVIS346 camera). Moreover, straight driving sequences are especially challenging for event-based multi-view stereo due to the little motion parallax present in them.

Setup. We select the commonly used *Zurich_City_04_a* sequence to provide a focused in-depth evaluation. We split the sequence into two halves for training and testing. The DSIs were created by fusing the left and right DSIs via the harmonic mean. Analogous to Sec. 4.3, we use the original filter from MC-EMVS [12] F_{orig} with a window size of 5×5 px and $C_{\text{orig}} = -4$, and a denser filter F_{denser} with a window size of 9×9 px and $C_{\text{denser}} = -2$. First, the network was trained for 3 epochs on the DSIs of the first half of the sequence and tested on those of the second. The process was then reversed and each network was used to predict in its testing half of the data sequence.

Results. The results of these experiments are displayed on the right half of Tab. 3 and illustrated in Fig. 3. Our approach drastically outperforms every other method across all metrics, with our multi-pixel network achieving even slightly better performance than the single-pixel network. For F_{orig}, it reduces the MAE by 55% on a 1.72x higher number of pixels compared to MC-EMVS with a morphological filter. For F_{denser}, depth estimation density is increased by an additional factor of 2.24 while performance remains mostly stable, yielding a reduction in MAE of 62% compared to the argmax operation from MC-EMVS. Remarkably, even on purely monocular DSIs filtered by F_{denser}, our framework achieved superior performance to all benchmarked methods for every metric except MedAE. These results underscore the robustness and versatility of our approach, even in complex real-world outdoor scenes.

4.5 Robustness of DERD-Net compared to other deep-learning stereo methods

Since there are no comparable learning-based methods that use prior camera poses, Tab. 4 compares end-to-end learning-based stereo methods, which are "instantaneous" (do not take into account

camera poses) and output dense depth. In order to use their output for efficient VO/SLAM, we would need an extra step of extracting features (keypoints). Instead, DERD-Net's semi-dense depth maps help avoid unnecessary computation by outputting 3D edges for direct visual odometry, as in [14].

We use the same train-test splits established as the other learning-based methods [5]. While absolute accuracy is not directly comparable, evaluating the errors in the different splits relative to each other is informative about robustness: we observe that our method is the first one to generalize robustly across all three sequences (Tab. 4). No other method reports good generalization on "split 2" of MVSEC because of the difference in dynamic characteristics of events in training and testing on that split [6, 43]. This is true even when compared to hybrid approaches, despite them also using stereo intensity frames ("2E+2F" input data modality). The observed robustness of our method to such shifts may be supported by the architectural choice of processing only small subregions as input (see Tab. 2 for Sub-DSI frame size), which encourages the model to learn generalizable patterns within the Sub-DSIs rather than memorizing global scene layout or dataset-specific context.

Table 4: Mean depth error [cm] of deep stereo methods on MVSEC indoor data. Values are collected from original sources.

Method	**Modality**	**Split 1**	**Split 2**	**Split 3**
DDES [6]	2E	16.7	29.4	27.8
EIT-Net [43]	2E	14.2	-	19.4
DTC-SPADE [44]	2E	13.5	-	17.1
Liu et al [45]	2E	20	25	31
StereoSpike [46]	2E	16.5	-	18.4
ASNet [47]	2E	20.46	28.74	22.15
Ghosh et al. [48]	2E	12.1	-	15.6
Chen et al [49]	2E	13.9	-	14.6
StereoFlow-Net [50]	2E	13	-	15
EIS (ICCV 2021)	2E + 2F	13.74	18.43	22.36
SCS-Net [51]	2E + 2F	11.4	-	13.5
N. Uddin et al [29]	2E + 2F	19.7	-	26.4
Zhao et al. [52]	2E + 2F	9.7	-	11.1
DERD-Net	2E	11.69	11.11	12.28

4.6 Sensitivity Analyses

In this section we carry out experiments varying the settings in Tab. 2. Furthermore, we analyze the robustness of our method to noisy camera poses obtained from an event-based SLAM system.

Sensitivity with respect to sub-DSI size. Varying the horizontal and vertical extent of the Sub-DSIs has an impact on our method's performance. Our experiments show that the performance of DERD-Net can be improved by increasing the frame size of the Sub-DSIs, at the expense of increasing the network complexity (e.g., parameter count and computational cost). See Appendix Sec. A.1.

Sensitivity with respect to DSI transformations. We analyzed how DERD-Net behaves in the case of previously unseen but structurally similar environments, obtained by means of horizontal and vertical flips of the DSIs. Although its performance worsened slightly, it still outperformed all baseline methods. This demonstrates robustness to the aforementioned transformations. See Appendix Sec. A.2.

Sensitivity with respect to noisy camera poses. To assess the importance of having accurate camera poses during DSI construction, we test our framework using noisy poses with drift, mimicking real-world SLAM conditions. Instead of ground-truth (GT) poses from LiDAR-IMU odometry, we use poses estimated by the stereo event-based VO system ES-PTAM [14], which reports an Absolute Trajectory Error (ATE) of 131.62 cm over a 50 m-deep scene in the *DSEC Zurich_City_04_a* sequence. Running DERD-Net with these imperfect poses yields the results shown in the top rows of Tab. 5. The percentage values in parentheses denote the relative differences with respect to the performance obtained using ideal (GT) poses (Tab. 3). Remarkably, performance improved across all metrics (likely due to the slight reduction in the number of evaluated points of comparable magnitude), demonstrating strong robustness of DERD-Net to noisy poses obtained from an event-based SLAM system.

We conduct an additional experiment where the original DERD-Net depth predictions were used to re-estimate the camera poses (in an offline manner, using the camera tracking module in [14]). The resulting poses were then used to build DSIs on which DERD-Net was evaluated. This "reprojection" loop allows us to assess, using standard depth-based metrics, the robustness of our method to noise in camera poses introduced by DERD-Net's own depth inaccuracies. The results are reported in the bottom rows of Tab. 5. The performance shows only minor degradation, particularly for F_{orig}, with no metric worsening by more than 13%. Remarkably, even under such self-induced pose noise,

Table 5: Depth estimation performance on *DSEC zurich_city_04_a* using poses computed by ES-PTAM or by camera tracking on DERD-Net's output ("Reprojection" rows). Relative changes with respect to Tab. 3, which reports results obtained using GT poses, are presented in parentheses.

Algorithm	Poses	Filter	Mean Err [m] ↓	Median Err [m] ↓	bad-pix [%] ↓	#Points [million] ↑
DERD-Net	ES-PTAM	F_{orig}	1.56 (-3.11%)	0.45 (-2.17%)	3.84 (-6.8%)	1.61 (-3.59%)
DERD-Net	ES-PTAM	F_{denser}	1.74 (-3.33%)	0.52 (-3.7%)	4.84 (-3.97%)	4.49 (-3.23%)
DERD-Net	Reprojection	F_{orig}	1.66 (+3.11%)	0.49 (+6.52%)	4.19 (+1.7%)	1.46 (-12.57%)
DERD-Net	Reprojection	F_{denser}	1.95 (+8.33%)	0.60 (+11.11%)	5.65 (+12.1%)	4.09 (-11.85%)

DERD-Net's depth estimation errors remain roughly 50% lower than those of prior SOTA methods using ideal poses. These results demonstrate the practical viability of deploying DERD-Net as a depth-estimation module within a self-sustaining SLAM system. Overall, our experiments confirm that DERD-Net remains remarkably robust even when the input poses are significantly degraded, as would be expected in real-world scenarios. See also Appendix Secs. A.3 and A.4

4.7 Runtime

Our network achieved an average inference time of only 0.37 ms per Sub-DSI on an NVIDIA RTX A6000. Since predictions are made independently per pixel, inference for each Sub-DSI can be parallelized on the GPU. The total inference time to estimate a depth map of average density (500 pixels) from MVSEC with F_{orig} is 1.12 ms.

Taking MVSEC as an example (DAVIS cameras of 346×260 pixels) and DSIs back-projecting 2 million events onto $D = 100$ depth planes, then each DSI creation takes ≈45 ms, DSI fusion takes ≈26 ms, and pixel selection takes ≈0.2 ms on an 8-core computer with Intel Xeon(R) W-2225 CPU operating at 4.10 GHz. These values are common for both the state-of-the-art method MC-EMVS and DERD-Net. It has been shown that DSI creation does not hamper real-time performance [53] because the 3D map can be updated infrequently and on-demand.

Our network adds only a very small runtime compared to the DSI creation time. This ultra-fast performance, combined with its lightweight architecture, enables efficient execution, making DERD-Net ideal for real-world applications requiring low-latency depth estimation.

5 Conclusion

We have developed the first learning-based multi-view stereo method for event-based depth estimation. Our approach combines input camera poses with events to produce intermediate geometric representations (DSIs) from which depth is estimated using deep learning. It is directly applicable to both monocular and stereo camera setups. By processing small independent subregions of DSIs in parallel, the framework operates independently of camera resolution and facilitates an efficient network under 1 MB in size with an inference time of only 0.37 ms.

Our framework consistently demonstrated superior performance across several metrics compared to other stereo methods and achieved comparable performance when using purely monocular data. It is the first learning-based depth estimation approach that reports robust generalization on all three *indoor flying* sequences of the MVSEC dataset. Adaptability to different scenes was confirmed on the outdoor driving DSEC dataset, for which it drastically outperformed benchmark approaches across all metrics. Moreover, our framework significantly increased the number of points for which depth can be robustly estimated from DSIs. It also showed strong robustness to noise in camera poses.

Given its exceptional performance, ultra-lightweight architecture, scalability and flexibility across different configurations, our method holds strong potential to become a standard approach for learning depth from events and is highly suitable for real-world robotic applications requiring low latency and low memory such as SLAM [15].

Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135.

References

[1] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck, "A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[2] Thomas Finateu *et al.*, "A 1280x720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86μm pixels, 1.066Geps readout, programmable event-rate controller and compressive data-formatting pipeline," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 112–114, 2020.

[3] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, 2022.

[4] Alex Zihao Zhu, Dinesh Thakur, Tolga Ozaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3D perception," *IEEE Robot. Autom. Lett.*, vol. 3, pp. 2032–2039, July 2018.

[5] Suman Ghosh and Guillermo Gallego, "Event-based stereo depth estimation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 10, pp. 9130–9149, 2025.

[6] Stepan Tulyakov, Francois Fleuret, Martin Kiefel, Peter Gehler, and Michael Hirsch, "Learning an event sequence embedding for dense event-based deep stereo," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 1527–1537, 2019.

[7] Maxim Maximov, Kevin Galim, and Laura Leal-Taixé, "Focus on defocus: bridging the synthetic to real domain gap for depth estimation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 1071–1080, 2020.

[8] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony, "Reducing the sim-to-real gap for event cameras," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 534–549, 2020.

[9] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego, "Secrets of event-based optical flow," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 628–645, 2022.

[10] Shuang Guo, Friedhelm Hamann, and Guillermo Gallego, "Unsupervised joint learning of optical flow and intensity with event cameras," in *Int. Conf. Comput. Vis. (ICCV)*, 2025.

[11] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza, "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time," *Int. J. Comput. Vis.*, vol. 126, pp. 1394–1414, Dec. 2018.

[12] Suman Ghosh and Guillermo Gallego, "Multi-event-camera depth estimation and outlier rejection by refocused events fusion," *Adv. Intell. Syst.*, vol. 4, no. 12, p. 2200221, 2022.

[13] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza, "DSEC: A stereo event camera dataset for driving scenarios," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4947–4954, 2021.

[14] Suman Ghosh, Valentina Cavinato, and Guillermo Gallego, "ES-PTAM: Event-based stereo parallel tracking and mapping," in *Eur. Conf. Comput. Vis. Workshops (ECCVW)*, pp. 70–87, 2024.

[15] Guillermo Gallego, Javier Hidalgo-Carrió, and Davide Scaramuzza, "Event-based SLAM," in *SLAM Handbook. From Localization and Mapping to Spatial Intelligence* (Luca Carlone, Ayoung Kim, Frank Dellaert, Timothy Barfoot, and Daniel Cremers, eds.), pp. 282–302, Cambridge University Press, 2026.

[16] Misha Mahowald and Tobi Delbrück, *Cooperative Stereo Matching Using Static and Dynamic Image Features*, pp. 213–238. Boston, MA: Springer US, 1989.

[17] Yi Zhou, Guillermo Gallego, Henri Rebecq, Laurent Kneip, Hongdong Li, and Davide Scaramuzza, "Semi-dense 3D reconstruction with a stereo event camera," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 242–258, 2018.

[18] Alex Zihao Zhu, Yibo Chen, and Kostas Daniilidis, "Realtime time synchronized event-based stereo," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 438–452, 2018.

[19] Yi Zhou, Guillermo Gallego, and Shaojie Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433–1450, 2021.

[20] Anass El Moudni, Fabio Morbidi, Sebastien Kramm, and Rémi Boutteau, "An event-based stereo 3D mapping and tracking pipeline for autonomous vehicles," in *IEEE Intell. Transp. Sys. Conf. (ITSC)*, pp. 5962–5968, 2023.

[21] Shintaro Shiba, Yannick Klose, Yoshimitsu Aoki, and Guillermo Gallego, "Secrets of event-based optical flow, depth, and ego-motion by contrast maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 7742–7759, 2024.

[22] Sheng Zhong, Junkai Niu, and Yi Zhou, "Deep visual odometry for stereo event cameras," *IEEE Robot. Autom. Lett.*, vol. 10, pp. 11078—11085, Nov. 2025.

[23] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *arXiv e-prints*, 2019.

[24] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun, "A survey on deep learning techniques for stereo-based depth estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1738–1764, 2020.

[25] Fabio Tosi, Luca Bartolomei, and Matteo Poggi, "A survey on deep stereo matching in the twenties," *Int. J. Comput. Vis.*, vol. 133, pp. 4245–4276, Feb. 2025.

[26] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza, "Learning monocular dense depth from events," in *Int. Conf. 3D Vision (3DV)*, pp. 534–542, Nov. 2020.

[27] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[28] Xu Zheng, Yexin Liu, Yunfan Lu, Tongyan Hua, Tianbo Pan, Weiming Zhang, Dacheng Tao, and Lin Wang, "Deep learning for event-based vision: A comprehensive survey and benchmarks," *arXiv e-prints*, 2023.

[29] S. M. Nadim Uddin, Soikat Hasan Ahmed, and Yong Ju Jung, "Unsupervised deep event stereo for depth estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7489–7504, 2022.

[30] Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza, "EMVS: Event-based multi-view stereo," in *British Mach. Vis. Conf. (BMVC)*, 2016.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

[32] Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han, "Understanding the difficulty of training transformers," in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[33] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza, "Fast image reconstruction with an event camera," in *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pp. 156–163, 2020.

[34] Lahav Lipson, Zachary Teed, and Jia Deng, "Raft-stereo: Multilevel recurrent field transforms for stereo matching," in *Int. Conf. 3D Vision (3DV)*, pp. 218–227, 2021.

[35] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, ACL, 2014.

[36] Michael Opitz, Horst Possegger, and Horst Bischof, "Efficient model averaging for deep neural networks," in *Asian Conf. Comput. Vis. (ACCV)*, pp. 205–220, 2016.

[37] Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord, "Diverse weight averaging for out-of-distribution generalization," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, pp. 10821–10836, 2022.

[38] Heiko Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 328–341, Feb. 2008.

[39] Sio-Hoi Ieng, Joao Carneiro, Marc Osswald, and Ryad Benosman, "Neuromorphic event-based generalized time-based stereovision," *Front. Neurosci.*, vol. 12, p. 442, 2018.

[40] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 3354–3361, 2012.

[41] Chengxi Ye, Anton Mitrokhin, Chethan Parameshwara, Cornelia Fermüller, James A. Yorke, and Yiannis Aloimonos, "Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pp. 5831–5838, 2020.

[42] Junkai Niu, Sheng Zhong, Xiuyuan Lu, Shaojie Shen, Guillermo Gallego, and Yi Zhou, "ESVO2: Direct visual-inertial odometry with stereo event cameras," *IEEE Trans. Robot.*, vol. 41, pp. 2164–2183, 2025.

[43] Soikat Hasan Ahmed, Hae Woong Jang, S M Nadim Uddin, and Yong Ju Jung, "Deep event stereo leveraged by event-to-image translation," *AAAI Conf. Artificial Intell.*, vol. 35, pp. 882–890, May 2021.

[44] Kaixuan Zhang, Kaiwei Che, Jianguo Zhang, Jie Cheng, Ziyang Zhang, Qinghai Guo, and Luziwei Leng, "Discrete time convolution for fast event-based stereo," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 8666–8676, June 2022.

[45] Peigen Liu, Guang Chen, Zhijun Li, Huajin Tang, and Alois Knoll, "Learning local event-based descriptor for patch-based stereo matching," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 412–418, 2022.

[46] Ulysse Rançon, Javier Cuadrado-Anibarro, Benoit R. Cottereau, and Timothée Masquelier, "Stereospike: Depth learning with a spiking neural network," *IEEE Access*, vol. 10, pp. 127428–127439, 2022.

[47] Zhu Jianguo, Wang Pengfei, Huang Sunan, Xiang Cheng, and Teo Swee Huat Rodney, "Stereo depth estimation based on adaptive stacks from event cameras," in *IECON 49th Annual Conf. IEEE Industrial Electr. Soc.*, pp. 1–6, 2023.

[48] Dipon Kumar Ghosh and Yong Ju Jung, "Two-stage cross-fusion network for stereo event-based depth estimation," *Expert Systems with Applications*, vol. 241, p. 122743, 2024.

[49] Wu Chen, Yueyi Zhang, Xiaoyan Sun, and Feng Wu, "Event-based stereo depth estimation by temporal-spatial context learning," *IEEE Signal Process. Lett.*, vol. 31, pp. 1429–1433, 2024.

[50] Hoonhee Cho, Jae-Young Kang, and Kuk-Jin Yoon, "Temporal event stereo via joint learning with stereoscopic flow," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 294–314, 2024.

[51] Hoonhee Cho and Kuk-Jin Yoon, "Selection and cross similarity for event-image deep stereo," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 470–486, 2022.

[52] Fengan Zhao, Qianang Zhou, and Junlin Xiong, "Edge-guided fusion and motion augmentation for event-image stereo," in *Eur. Conf. Comput. Vis. (ECCV)*, 2024.

[53] Suman Ghosh and Guillermo Gallego, "Event-based stereo depth estimation from ego-motion using ray density fusion," *Eur. Conf. Comput. Vis. Workshops (ECCVW)*, 2022.

A Appendix: Sensitivity Analyses

In this section we report additional experiments to assess our method's robustness to changes in hyperparameter (Tab. 6), changes in the scene (Tab. 7) and noise in camera poses (Tabs. 8 and 9). We also complement the evaluation on the downstream task of camera tracking (Tab. 10).

A.1 Sensitivity with respect to Sub-DSI Size

The hyperparameters used for our experiments are detailed in Tab. 2. The LiDAR's sampling interval, the estimated minimum and maximum depths, and the number of depth layers D have all been defined to match the protocol in [12]. In Sec. 4.3, we already provided a comparison between the two AGT filters F_{orig} and F_{denser}. In this section, we therefore analyze the impact of the size of the Sub-DSI.

We retrained the network for one epoch on the *indoor_flying 2* and *3* sequences and compared its test performance on sequence 1 using radii $r_W = r_H$ of 2, 3, and 4 px, effectively creating Sub-DSI frames of size 5×5, 7×7, and 9×9 px, respectively. Layer dimensions were adapted, while the overall network architecture remained fixed.

Table 6: Sensitivity analysis of DERD-Net's performance for different Sub-DSI frame sizes after one epoch of training. *MVSEC indoor_flying_1*.

Sub-DSI frame size	Modality	Mean Err [cm] ↓	Median Err [cm] ↓	bad-pix [%] ↓	SILog Err ×100 ↓	AErrR [%] ↓	log RMSE ×100 ↓	$\delta < 1.25$ [%] ↑	$\delta < 1.25^2$ [%] ↑	$\delta < 1.25^3$ [%] ↑	#Points [million]↑
5×5	stereo + F_{orig}	13.92	6.51	0.96	1.20	5.71	10.97	96.07	98.56	99.60	0.99
7×7	stereo + F_{orig}	12.13	5.82	0.85	0.97	5.04	9.93	96.81	98.85	99.66	0.98
9×9	stereo + F_{orig}	11.58	5.54	0.82	0.92	4.85	9.64	96.97	98.91	99.68	0.98
5×5	stereo + F_{denser}	18.42	7.88	1.79	2.10	7.64	14.50	93.71	97.58	99.10	3.03
7×7	stereo + F_{denser}	15.41	6.80	1.35	1.55	6.34	12.56	95.24	98.20	99.36	3.01
9×9	stereo + F_{denser}	14.59	6.39	1.30	1.48	6.07	12.21	95.52	98.29	99.37	2.99

From the results reported in Tab. 6 it can be inferred that performance appears to improve as the radii increase. The network was able to achieve better results after a single epoch using a frame size of 9×9 than when fully trained on 7×7 frames (see Tab. 12 in this Appendix), highlighting its potential for further performance improvements.

Nevertheless, increasing the frame size to 9×9 yielded a reduction of 5% in MAE for both filters after a single training epoch. In contrast to that, the network had to apply 65% more 3D-convolutional operations and its total amount of parameters raised from 70k to 270k. We therefore decided for a 7×7 frame size for this study. Future research could explore the evident potential to further boost performance by optimizing the sub-DSI size, considering the trade-off between accuracy, parameter count and computational costs.

A.2 Sensitivity with respect to DSI Transformations

Next, we analyze the robustness of DERD-Net with respect to transformations of the DSI, in particular axis-aligned reflections of the DSIs generated from the *indoor_flying_1* sequence of the MVSEC dataset. Specifically, we flipped the DSIs horizontally, vertically, and both horizontally and vertically. These transformations effectively generate scenes with similar geometric properties (e.g., distance ranges) but novel spatial configurations. This allows us to evaluate how well the network generalizes to previously unseen, yet structurally similar environments. We therefore purposely used *no data augmentation during training* to ensure a representative assessment of the network's inherent robustness. Analogous to previous experiments, we used the single-pixel network that was trained solely on the original *indoor_flying 2* and *3* for evaluation. No retraining was performed.

The results of these experiments are displayed in Tab. 7. Performance worsened only slightly, with results that still significantly outperform all SOTA methods for all tested configurations, indicating that our network might effectively generalize to scenes that share similar depth ranges and texture with those on which it was originally trained.

Table 7: Performance of DERD-Net when applying axis-aligned reflections. *MVSEC indoor_flying_1*.

Reflection	Modality	Mean Err [cm] ↓	Median Err [cm] ↓	bad-pix [%] ↓	SILog Err ×100 ↓	AErrR [%] ↓	log RMSE ×100 ↓	$\delta < 1.25$ [%] ↑	$\delta < 1.25^2$ [%] ↑	$\delta < 1.25^3$ [%] ↑	#Points [million] ↑
none	stereo + F_{orig}	11.69	5.42	0.86	0.97	4.90	9.88	96.87	98.85	99.65	0.98
vertical	stereo + F_{orig}	12.81	6.27	0.92	1.06	5.43	10.34	96.56	98.80	99.64	0.98
horizontal	stereo + F_{orig}	14.26	6.81	1.01	1.23	5.88	11.08	95.85	98.56	99.60	0.98
horizontal + vertical	stereo + F_{orig}	14.28	6.79	1.02	1.23	5.92	11.11	95.79	98.53	99.60	0.98
none	stereo + F_{denser}	14.86	6.47	1.31	1.49	6.14	12.30	95.45	98.24	99.37	3.01
vertical	stereo + F_{denser}	16.27	7.42	1.44	1.65	6.79	12.96	94.91	98.09	99.31	3.01
horizontal	stereo + F_{denser}	18.23	7.87	1.60	1.91	7.32	13.84	93.73	97.71	99.25	3.01
horizontal + vertical	stereo + F_{denser}	18.86	8.37	1.68	1.98	7.64	14.14	93.46	97.61	99.22	3.01

A.3 Sensitivity with respect to Noise in Camera Poses

In Section Sec. 4.6 we summarized the sensitivity of DERD-Net with respect to noise in the camera poses used to build DSIs, on DSEC data. For completeness, we now show results on MVSEC data.

We repeat the same experiment as that in the top rows of Tab. 5 on the *MVSEC indoor_flying_1* sequence, for which ES-PTAM reported an ATE of 14.93 cm over a 6 m depth range. The results shown in Tab. 8, and compared to those obtained with GT poses in Tab. 12, again highlight DERD-Net's strong robustness to noisy poses estimated by an event-based SLAM system: the MAE and MedAE increased only slightly, while the bad-pix metric even improved. The most pronounced decline was in the number of evaluated points. Nevertheless, DERD-Net with F_{denser} still predicts depth for 69% more pixels than MC-EMVS with F_{orig} under ideal poses, while achieving a 30% lower MAE. For its multi-pixel variant, DERD-Net evaluated with noisy poses maintains superior performance over all state-of-the-art methods using GT poses, while still predicting the highest number of points.

Table 8: Quantitative depth estimation performance on *MVSEC indoor_flying_1* using poses computed downstream of ES-PTAM. Relative changes with respect to Tab. 12, which reports results obtained using GT poses, are presented in parentheses.

Algorithm	Poses	Filter	Mean Err [m] ↓	Median Err [m] ↓	bad-pix [%] ↓	#Points [million] ↑
DERD-Net	ES-PTAM	F_{orig}	12.72 (+8.81%)	6.33 (+16.79%)	0.65 (-24.42%)	0.68 (-30.61%)
DERD-Net	ES-PTAM	F_{denser}	15.76 (+6.06%)	7.36 (+13.76%)	1.17 (-10.69%)	1.62 (-46.18%)
DERD-Net (multi-pixel)	ES-PTAM	F_{orig}	13.53 (+10.27%)	6.76 (+19.01%)	0.65 (-24.42%)	3.41 (-33.91%)
DERD-Net (multi-pixel)	ES-PTAM	F_{denser}	16.60 (+6.62%)	7.65 (+16.08%)	1.21 (-11.68%)	6.27 (-46.46%)

In the interest of thoroughness, we also used poses from the state-of-the-art event-based stereo visual–inertial odometry system ESVO2 [42], which is notably more accurate than ES-PTAM, to run DERD-Net on *MVSEC indoor_flying_1*, *indoor_flying_2*, and *indoor_flying_3*. The mean results are reported in Tab. 9. Compared to Tab. 3, DERD-Net shows only a slight decrease in depth estimation performance on F_{orig}, while it even improves on F_{denser}, confirming its robustness to pose noise from an event-based SLAM system integrating events and inertial data.

Table 9: Quantitative depth estimation performance averaged over *MVSEC indoor_flying_1, _2*, and *_3* using poses computed downstream of ESVO2. Relative changes with respect to Tab. 3, which reports results obtained using GT poses, are presented in parentheses.

Algorithm	Poses	Filter	Mean Err [m] ↓	Median Err [m] ↓	bad-pix [%] ↓	#Points [million] ↑
DERD-Net	ESVO2	F_{orig}	11.53 (-1.37%)	5.73 (+4.18%)	1.09 (+22.47%)	0.61 (-22.78%)
DERD-Net	ESVO2	F_{denser}	13.69 (-10.17%)	6.32 (-5.39%)	1.46 (-14.12%)	1.45 (-47.65%)

A.4 Downstream Camera Tracking Performance Analysis

As intermediate results to those in the bottom part of Tab. 5, we report offline camera tracking performance using the edge-alignment camera tracking module in [14] acting on input events and the local maps built using DERD-Net's depth predictions (from GT poses). Camera tracking performance is given in terms of ATE and Absolute Rotation Error (ARE) on the DSEC [13] driving dataset in Tab. 10. Although our model was trained only on the *Zurich_City_04_a* sequence, we evaluate it on all *Zurich_City_04* sequences to highlight its generalization capabilities.

Table 10: Camera tracking performance on *DSEC zurich_city_04, without DERD-Net retraining.*

Sequence	zc04a	zc04b	zc04c	zc04d	zc04e	zc04f
Duration [s]	35	13.4	53	47.8	13.6	43.1
ATE RMSE [cm] ↓	17.07	7.85	14.00	55.64	5.71	36.11
ARE RMSE [deg] ↓	0.31	0.08	0.45	0.67	0.11	0.72

The obtained pose errors in the 50 m depth range scenes across all sequences show strong performance of DERD-Net for downstream tasks such as pose estimation via simple photometric edge alignment on event images, as well as its robust generalization even when trained on a single sequence. Training on a more diverse set of DSEC sequences would be expected to further enhance these results. These values are not comparable to online SLAM tracking results because they assume that the 3D map was pre-built offline using DERD-Net with GT poses. Therefore, the estimated camera poses reported here do not accumulate drift.

B Appendix - Detailed per-Sequence Results

The average results of different SOTA methods compared to DERD-Net are presented in Tab. 11. In Tabs. 12 to 14, the individual performance on each of the respective sequences *indoor_flying 1, 2, 3* from the MVSEC dataset are displayed. Table 15 presents the corresponding results for the *Zurich_City_04_a* sequence from the DSEC dataset.

MVSEC Averaged

Table 11: Quantitative comparison of the proposed methods with the state of the art. *MVSEC indoor_flying* (average).

	Algorithm	Modality	Mean Err [cm] ↓	Median Err [cm] ↓	bad-pix [%] ↓	SILog Err ×100 ↓	AErrR [%] ↓	log RMSE ×100 ↓	$\delta < 1.25$ [%] ↑	$\delta < 1.25^2$ [%] ↑	$\delta < 1.25^3$ [%] ↑	#Points [million]↑
SOTA	EMVS [11]	monocular + F_{orig}	33.78	14.35	3.84	4.20	12.74	20.72	84.75	94.87	97.99	1.27
	EMVS [11]	monocular + F_{denser}	50.32	20.81	11.46	11.37	20.87	33.75	73.43	88.09	93.71	4.15
	ESVO [19]	stereo	25.00	10.59	3.35	3.48	10.19	18.83	90.44	95.76	97.98	2.04
	ESVO indep. 1s	stereo	22.70	9.83	2.83	3.03	9.59	17.53	91.82	96.50	98.38	1.56
	SGM indep. 1s	stereo	35.42	12.35	6.39	8.45	16.17	29.49	85.34	93.05	96.03	**14.46**
	GTS indep. 1s	stereo	389.00	45.43	38.45	74.47	102.92	89.08	49.56	62.19	69.36	0.06
	MC-EMVS [12]	stereo + F_{orig}	20.07	9.53	1.35	1.72	7.80	13.24	95.04	98.08	99.21	0.81
	MC-EMVS [12]	stereo + F_{denser}	28.38	12.38	3.26	3.43	10.94	18.60	89.41	96.09	98.33	2.77
	MC-EMVS [12] + MF	stereo + F_{orig}	20.64	9.72	1.43	1.80	7.94	13.54	94.74	97.95	99.17	3.00
Ours	DERD-Net	monocular + F_{orig}	23.68	11.55	2.78	2.62	10.18	16.20	90.25	97.36	99.02	1.21
	DERD-Net	monocular + F_{denser}	28.52	13.85	4.87	3.77	12.33	19.46	85.78	95.77	98.50	4.15
	DERD-Net without EL	stereo + F_{orig}	<u>12.00</u>	5.73	0.92	<u>0.98</u>	5.15	<u>9.92</u>	<u>96.99</u>	<u>98.86</u>	<u>99.63</u>	0.79
	DERD-Net	stereo + F_{orig}	**11.69**	**5.50**	**0.89**	**0.96**	**5.05**	**9.83**	**96.99**	**98.89**	**99.64**	0.79
	DERD-Net	stereo + F_{denser}	15.24	6.68	1.70	1.54	6.41	12.44	95.00	98.19	99.39	2.77
	DERD-Net multi-pixel	stereo + F_{orig}	12.02	<u>5.63</u>	<u>0.90</u>	0.99	<u>5.13</u>	9.94	96.89	98.83	99.63	4.32
	DERD-Net multi-pixel	stereo + F_{denser}	15.68	6.73	1.74	1.59	6.54	12.61	94.75	98.10	99.36	<u>11.33</u>

MVSEC Indoor Flying 1

Table 12: Quantitative comparison of the proposed methods with the state of the art. *MVSEC indoor_flying_1*.

	Algorithm	Modality	Mean Err [cm] ↓	Median Err [cm] ↓	bad-pix [%] ↓	SILog Err ×100 ↓	AErrR [%] ↓	log RMSE ×100 ↓	$\delta < 1.25$ [%] ↑	$\delta < 1.25^2$ [%] ↑	$\delta < 1.25^3$ [%] ↑	#Points [million] ↑
SOTA	EMVS [11]	monocular + F_{orig}	39.37	14.95	3.05	4.72	13.25	22.10	82.03	93.43	97.62	1.21
	EMVS [11]	monocular + F_{denser}	60.65	24.20	12.56	13.88	24.49	37.29	69.04	84.58	91.54	4.64
	ESVO [19]	stereo	24.04	10.21	2.54	2.94	9.76	17.17	91.43	96.53	98.55	1.95
	ESVO indep. 1s	stereo	23.39	10.03	2.18	2.79	9.78	16.72	91.57	96.84	98.79	1.41
	SGM indep. 1s	stereo	35.45	13.61	5.54	7.35	15.03	27.46	85.96	93.51	96.40	11.64
	GTS indep. 1s	stereo	700.37	38.39	32.51	79.26	111.21	91.44	54.27	67.16	73.39	0.03
	MC-EMVS [12]	stereo + F_{orig}	22.53	9.72	1.30	1.94	7.91	14.11	93.49	97.50	99.17	0.96
	MC-EMVS [12]	stereo + F_{denser}	31.43	13.14	3.11	3.99	11.69	20.03	88.16	95.28	97.91	3.01
	MC-EMVS [12] + MF	stereo + F_{orig}	23.33	9.90	1.39	2.08	8.12	14.61	93.16	97.28	99.05	3.48
Ours	DERD-Net	monocular + F_{orig}	25.76	12.60	1.89	2.62	10.39	16.19	89.42	97.60	99.24	1.50
	DERD-Net	monocular + F_{denser}	30.90	15.00	3.34	3.86	12.62	19.65	85.37	95.94	98.61	4.64
	DERD-Net without EL	stereo + F_{orig}	12.05	5.65	0.86	0.97	4.98	9.90	96.89	98.85	99.66	0.98
	DERD-Net	stereo + F_{orig}	11.69	5.42	0.86	0.97	4.90	9.88	96.87	98.85	99.65	0.98
	DERD-Net	stereo + F_{denser}	14.86	6.47	1.31	1.49	6.14	12.30	95.45	98.24	99.37	3.01
	DERD-Net multi-pixel	stereo + F_{orig}	12.27	5.68	0.86	0.99	5.06	9.98	96.76	98.77	99.65	5.16
	DERD-Net multi-pixel	stereo + F_{denser}	15.57	6.59	1.37	1.58	6.39	12.61	95.13	98.11	99.33	11.71

MVSEC Indoor Flying 2

Table 13: Quantitative comparison of the proposed methods with the state of the art. *MVSEC indoor_flying_2*.

	Algorithm	Modality	Mean Err [cm] ↓	Median Err [cm] ↓	bad-pix [%] ↓	SILog Err ×100 ↓	AErrR [%] ↓	log RMSE ×100 ↓	$\delta < 1.25$ [%] ↑	$\delta < 1.25^2$ [%] ↑	$\delta < 1.25^3$ [%] ↑	#Points [million] ↑
SOTA	EMVS [11]	monocular + F_{orig}	31.42	13.01	6.15	4.56	13.37	21.80	84.07	94.72	97.88	1.17
	EMVS [11]	monocular + F_{denser}	45.69	17.96	14.66	11.74	19.86	34.81	72.69	88.27	94.01	3.65
	ESVO [19]	stereo	21.34	8.97	3.75	3.48	9.32	19.14	91.60	95.88	97.86	1.89
	ESVO indep. 1s	stereo	20.42	8.63	3.50	3.24	9.14	18.35	92.03	96.19	98.19	1.41
	SGM indep. 1s	stereo	32.94	8.75	8.29	9.50	15.82	31.54	84.40	92.33	95.48	16.95
	GTS indep. 1s	stereo	167.14	37.23	43.08	71.91	94.78	86.93	49.36	60.54	67.76	0.07
	MC-EMVS [12]	stereo + F_{orig}	18.20	8.49	1.77	1.78	8.13	13.59	95.53	98.13	99.08	0.65
	MC-EMVS [12]	stereo + F_{denser}	25.81	10.34	4.65	3.48	10.89	18.91	89.10	95.93	98.35	2.25
	MC-EMVS [12] + MF	stereo + F_{orig}	18.58	8.68	1.86	1.81	8.19	13.71	95.27	98.07	99.09	2.42
Ours	DERD-Net	monocular + F_{orig}	23.37	10.43	4.98	3.31	11.07	18.41	88.30	96.23	98.46	0.98
	DERD-Net	monocular + F_{denser}	27.65	12.75	8.68	4.60	13.39	21.76	82.75	94.34	97.90	3.65
	DERD-Net without EL	stereo + F_{orig}	11.44	5.23	1.34	1.13	5.45	10.67	96.67	98.60	99.52	0.58
	DERD-Net	stereo + F_{orig}	11.11	4.94	1.26	1.10	5.34	10.50	96.69	98.66	99.54	0.58
	DERD-Net	stereo + F_{denser}	14.46	5.92	2.78	1.72	6.74	13.17	94.05	97.88	99.32	2.25
	DERD-Net multi-pixel	stereo + F_{orig}	11.29	4.88	1.28	1.14	5.39	10.66	96.50	98.61	99.54	3.21
	DERD-Net multi-pixel	stereo + F_{denser}	14.92	5.95	2.85	1.80	6.86	13.47	93.67	97.76	99.29	9.52

MVSEC Indoor Flying 3

Table 14: Quantitative comparison of the proposed methods with the state of the art. *MVSEC indoor_flying_3*.

	Algorithm	Modality	Mean Err [cm] ↓	Median Err [cm] ↓	bad-pix [%] ↓	SILog Err ×100 ↓	AErrR [%] ↓	log RMSE ×100 ↓	$\delta < 1.25$ [%] ↑	$\delta < 1.25^2$ [%] ↑	$\delta < 1.25^3$ [%] ↑	#Points [million] ↑
SOTA	EMVS [11]	monocular + F_{orig}	30.54	15.09	2.31	3.33	11.59	18.27	88.16	96.45	98.47	1.42
	EMVS [11]	monocular + F_{denser}	44.62	20.26	7.15	8.50	18.26	29.15	78.55	91.41	95.57	4.15
	ESVO [19]	stereo	29.62	12.61	3.78	4.02	11.50	20.20	88.28	94.88	97.52	2.29
	ESVO indep. 1s	stereo	24.29	10.84	2.81	3.05	9.84	17.54	91.87	96.46	98.16	1.86
	SGM indep. 1s	stereo	37.86	14.69	5.33	8.52	17.65	29.46	85.67	93.31	96.21	14.81
	GTS indep. 1s	stereo	299.48	60.66	39.75	72.24	102.77	88.87	45.04	58.86	66.94	0.08
	MC-EMVS [12]	stereo + F_{orig}	19.49	10.38	0.99	1.43	7.35	12.01	96.09	98.60	99.38	0.82
	MC-EMVS [12]	stereo + F_{denser}	27.89	13.65	2.01	2.83	10.25	16.85	90.97	97.05	98.73	3.04
	MC-EMVS [12] + MF	stereo + F_{orig}	20.02	10.59	1.02	1.50	7.50	12.30	95.79	98.51	99.36	3.11
Ours	DERD-Net	monocular + F_{orig}	21.91	11.62	1.46	1.93	9.07	14.01	93.02	98.25	99.36	1.14
	DERD-Net	monocular + F_{denser}	27.01	13.80	2.58	2.85	10.99	16.96	89.23	97.04	98.98	4.15
	DERD-Net without EL	stereo + F_{orig}	12.50	6.31	0.57	0.84	5.03	9.20	97.41	99.13	99.72	0.82
	DERD-Net	stereo + F_{orig}	12.28	6.13	0.55	0.82	4.91	9.11	97.41	99.15	99.74	0.82
	DERD-Net	stereo + F_{denser}	16.39	7.64	1.02	1.40	6.36	11.84	95.49	98.45	99.48	3.04
	DERD-Net multi-pixel	stereo + F_{orig}	12.50	6.34	0.56	0.84	4.93	9.17	97.41	99.12	99.71	4.59
	DERD-Net multi-pixel	stereo + F_{denser}	16.55	7.65	1.01	1.38	6.36	11.76	95.44	98.43	99.47	12.77

DSEC

Table 15: Quantitative comparison of the proposed methods with the state of the art. *DSEC Zurich_City_04_a*.

	Algorithm	Modality	Mean Err [m] ↓	Median Err [m] ↓	bad-pix [%] ↓	SILog Err ×100 ↓	AErrR [%] ↓	log RMSE ×100 ↓	$\delta < 1.25$ [%] ↑	$\delta < 1.25^2$ [%] ↑	$\delta < 1.25^3$ [%] ↑	#Points [million]↑
SOTA	EMVS [11]	monocular + F_{orig}	5.64	2.52	13.68	13.23	25.52	36.49	72.56	87.12	93.56	1.31
	EMVS [11]	monocular + F_{denser}	7.01	3.56	24.33	23.07	41.74	48.52	63.00	79.81	87.71	6.09
	ESVO [19]	stereo	3.93	1.62	10.54	8.30	17.66	28.90	84.37	92.81	96.05	9.40
	SGM [38]	stereo	6.74	1.58	15.25	17.95	18.42	42.51	80.66	89.12	93.16	8.30
	GTS [39]	stereo	26.24	1.62	32.56	61.58	33.45	79.26	68.07	78.39	85.85	0.11
	MC-EMVS [12]	stereo + F_{orig}	3.27	0.90	10.75	8.19	17.48	28.73	83.30	91.56	95.62	1.25
	MC-EMVS [12]	stereo + F_{denser}	4.76	1.56	17.42	15.84	30.67	40.45	76.37	86.01	90.97	4.64
	MC-EMVS [12] + MF	stereo + F_{orig}	3.51	0.96	11.81	8.89	18.84	29.99	81.72	90.68	95.07	3.83
Ours	DERD-Net	monocular + F_{orig}	3.12	1.60	5.50	3.96	12.19	19.92	86.06	96.29	98.61	2.10
	DERD-Net	monocular + F_{denser}	3.01	1.50	6.35	4.04	12.24	20.12	86.46	96.07	98.41	6.09
	DERD-Net	stereo + F_{orig}	1.61	**0.46**	4.12	2.78	7.03	16.68	93.50	97.05	98.66	1.67
	DERD-Net	stereo + F_{denser}	1.80	0.54	5.04	2.91	7.59	17.06	92.09	96.72	98.56	4.64
	DERD-Net multi-pixel	stereo + F_{orig}	**1.59**	0.47	**3.81**	**2.54**	**6.76**	**15.93**	**93.60**	**97.18**	**98.78**	6.59
	DERD-Net multi-pixel	stereo + F_{denser}	1.79	0.54	4.61	2.76	7.46	16.62	92.31	96.82	98.62	**14.74**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The framework and it advantages are explained in Sec. 3. The results are displayed in Sec. 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mention the requirement of available Ground Truth for training (Sec. 4) and explicitly point out that accuracy cannot be directly compared to end-to-end learning-based methods (Sec. 4.5). We state the need for camera pose in the construction of the DSIs (Sec. 3.1). More details about limitations of DSIs are available in the linked source [12].

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: No theoretical results are included.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Tab. 1, Tab. 2, Sec. 3 and Sec. 4. We also provide code and model checkpoints in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplementary material of our submission contains the full code. The datasets are publicly available.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Tab. 1, Tab. 2 and Sec. 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following standard evaluation protocols used by comparable prior methods to ensure fairness through consistent experimental settings, we do not plot error bars. Instead, we test several variations of our network and report differences in the estimation errors (Tabs. 11 to 15). We also provide a quantitative evaluation of the effects of uncertainty on our networks' accuracy by comparing the pixel selection maps – visualizations of uncertainty – to the depth maps created by the networks in Fig. 3 and the video included in the supplementary material. Furthermore, we provide an analysis of how variations to the dataset (Sec. A.2) and hyperparameters (Sec. A.1) affect the network's performance.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Sec. 4.7.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research does not infringe the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our paper has no foreseeable significant societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We see no parts of our paper being at high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We did correctly cite the creators of MC-EMVS for the DSI approach [12] as well as the creators for the datasets of MVSEC [4] and DSEC [13].

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The full documentation for code and models is included in the supplementary material of our submission.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing experiments or research with human subjects has been part of this study.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: There are no such risks to report for our study.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs have not been used as an essential component in this study.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.