DELTA: Dense Depth from Events and LiDAR using Transformer's Attention

Vincent Brebion[1,2,3] Julien Moreau[2] Franck Davoine[3]
[1]Centre for Environmental and Climate Science, Lund University, Sweden – vincent.brebion@cec.lu.se
[2]Université de technologie de Compiègne, CNRS, Heudiasyc, France – julien.moreau@hds.utc.fr
[3]CNRS, INSA Lyon, UCBL, LIRIS, UMR5205, France – franck.davoine@cnrs.fr

Abstract

Event cameras and LiDARs provide complementary yet distinct data: respectively, asynchronous detections of changes in lighting versus sparse but accurate depth information at a fixed rate. To this day, few works have explored the combination of these two modalities. In this article, we propose a novel neural-network-based method for fusing event and LiDAR data in order to estimate dense depth maps. Our architecture, DELTA, exploits the concepts of self- and cross-attention to model the spatial and temporal relations within and between the event and LiDAR data. Following a thorough evaluation, we demonstrate that DELTA sets a new state of the art in the event-based depth estimation problem, and that it is able to reduce the errors up to four times for close ranges compared to the previous SOTA.

1. Introduction

Due to their unique nature, event cameras offer a large paradigm change in the field of computer vision. By capturing changes of lighting for each pixel independently and asynchronously, and by transmitting this information as spikes of data (*events*), these cameras provide ultra-low latency perception (in the order of the microsecond). They allow for novel applications and present an invaluable potential, especially under challenging lighting or motion conditions where traditional frame-based cameras would fail.

Their combination with other modalities like RGB or grayscale cameras is a topic that has been deeply explored over the past few years. Yet, the use of event cameras with LiDARs remains relatively untouched, despite the valuable information a depth sensor can bring for scene analysis. As displayed in Fig. 1, we propose here to exploit both the precise edge detection and the high temporal resolution of the event data to densify both spatially and temporally the LiDAR data, resulting in dense high-rate depth maps.

This article offers four key contributions. We propose **(1)** a novel attention-based network, which we call DELTA (for "Dense depth from Events and LiDAR using Transformer's Attention"). DELTA combines information from low-rate projected LiDAR point clouds with higher-rate small temporal windows of events, in order to derive accurate dense depth maps. Thanks to its attention- and recurrence-based design, unlike traditional CNNs, our network is able to extract the most relevant spatial and temporal features within and between the event and LiDAR data. The introduction of **(2)** a propagation memory for a fusion at the highest input rate and of **(3)** a central memory acting as a main recurrence both allow us to outperform the state of the art, and are critical contributions as shown through ablation studies. An extensive evaluation of DELTA is conducted on multiple automotive datasets, where LiDAR and event sensors are most commonly used together. We demonstrate that in most cases our proposed network is able to offer **(4)** a clear improvement over the state of the art, especially for close ranges (often the most critical in dynamic scenes, *e.g.*, close pedestrians in automotive scenarios), where the mean error in depth estimation is reduced up to four times.

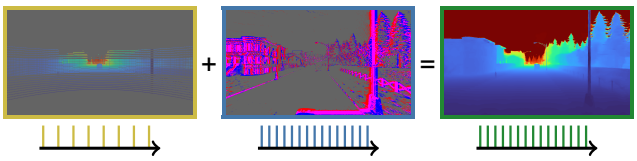

Figure 1. Overall principle of DELTA. Sparse and low-rate projected LiDAR data (■) is densified spatially and temporally using higher-rate temporal windows of event data (■), resulting in dense high-rate depth maps (■). Displayed here is an example of the high-quality depth maps produced by DELTA.

Additional analysis and results are available as supplementary material. The source code and trained models are available at https://vbrebion.github.io/DELTA.

2. Related Work

2.1. Transformers for Event-based Data

The Transformer [45] has become the state-of-the-art architecture in numerous domains. Its attention mechanism

models explicitly the relations between relevant elements in a sequence, making it able to understand underlying structures. For computer vision, the arrival of the Vision Transformer (ViT) [12] has been a notable landmark, outperforming more traditional convolution-based networks. As such, researchers have started investigating how the Transformer architecture could be adapted to event-based cameras. Two philosophies have emerged over the years. **(1)** Some authors use directly the raw stream of events (without any preprocessing) as the input sequence to their network, and use the Transformer architecture to process it. This approach is particularly complex, as each event contains little information, making the modeling of their relations difficult. To contain enough context, sequences of events should also be of consequent size. As of today, this method has only been applied to the tasks of object detection and classification [23, 27, 32], where the event data can be highly compressed. **(2)** To circumvent these issues, most authors instead accumulate events in a frame-like representation, and process it using the standard patch-based format proposed with ViT. Investigated tasks include video reconstruction [48], object detection [17], classification [35, 36, 46], depth estimation [36], and optical flow [43]. In this work, we follow principle **(2)**, as it allows the network to extract meaningful spatial and temporal features from the event and LiDAR data for the depth densification task.

2.2. Fusion of Event and LiDAR Data

To this day, most research works using both the LiDAR and event-based modalities address the problem of extrinsic calibration [9, 22, 40, 41], or use them as part of the construction of a dataset [3, 5, 18, 49]. Recently, authors have started investigating the issues of enhancing point clouds with event-based data [26], of estimating dense depth maps from event and LiDAR data [3, 10], or of tracking humans in adversarial lighting conditions [37].

2.3. Event-Based Depth Estimation

The idea of estimating sparse or dense depth maps from events has been actively explored over the past decade. Three main approaches can be distinguished. **(1)** Some authors estimate depths in a monocular fashion, using only events from a single event camera [6, 21, 24, 29, 31, 33, 50], or using events and frames [16, 20, 28, 36] from a DAVIS camera [2]. These approaches are particularly challenging, as they lack any three-dimensional information. **(2)** Some authors have tried to estimate depth in a stereo fashion, by using a pair of event cameras [7, 19, 30, 33, 38, 39], with [7] and [33] achieving notably good results. **(3)** Finally, some authors prefer to use directly a depth sensor, and use the stream of event as a mean to densify and/or to temporally upsample the depth data. This depth sensor can either be an RGB-D camera [47] or a LiDAR [3, 10, 26], with [10, 26] being based respectively on 3D- and 2D-geometry methods, and with [3] (our previous work) being based on a CNN.

2.4. Positioning of our Work Compared to the State of the Art

Like [3, 10, 26], we exploit LiDAR data to solve the problem of event-based depth estimation. Like these works, we argue (and we show in Sec. 4) that having reference depth points (even if sparse both spatially and temporally) is of great help for solving the event-based depth estimation problem. But unlike [10, 26], we do not use a geometry-based approach, due to limitations highlighted by these two works: depth estimation is only possible for areas with close LiDAR points, output depth maps frequency is restricted to the one of the LiDAR data, and these methods are sensitive to noise. Our approach is more similar to our previous work [3], as we use a learning-based approach to solve these limitations. But while we used a convolution-based network in [3], we propose here instead a novel attention-based network, which can better capture the spatial and temporal relations within and between event and LiDAR data, without being restricted by the limited reception field of convolutions. As shown in Sec. 4, this novel architecture greatly outperforms the state of the art, especially for short ranges.

3. Method

3.1. A Single Depth Map

In [3], we argued (and still argue) that, as an event describes a change between two illumination values, it may also be linked to a change between two depth values, hence the need of estimating a depth *before* the event happened ($\mathrm{d_{bf}}$) and a depth *after* the event happened ($\mathrm{d_{af}}$). However, in [3], the computation of the depths is not done at the event level, but at the temporal window level. Therefore, we proposed to estimate two depth maps for each temporal window of events ($\mathrm{D_{bf}}$ and $\mathrm{D_{af}}$), and to then use the events as a mask to assign to each of them their two depths $\mathrm{d_{bf}}$ and $\mathrm{d_{af}}$.

We argue here that this solution was ill-posed: multiple events can be produced by a single pixel during a temporal window, with each of these events requiring their own individual depths $\mathrm{d_{bf}}$ and $\mathrm{d_{af}}$. Furthermore, at the temporal window level, estimating two depth maps is redundant, as the $\mathrm{D_{af}}$ depth map of a temporal window can be recovered from the $\mathrm{D_{bf}}$ depth map of the following temporal window.

Since our objective here is not to estimate depths at the event level but at the temporal window level, we follow these conclusions and only estimate a single depth map for each temporal window of events, *i.e.*, the $\mathrm{D_{bf}}$ map.

3.2. Architecture

We propose a novel attention-based recurrent network to estimate depth maps from LiDAR and event data, result-

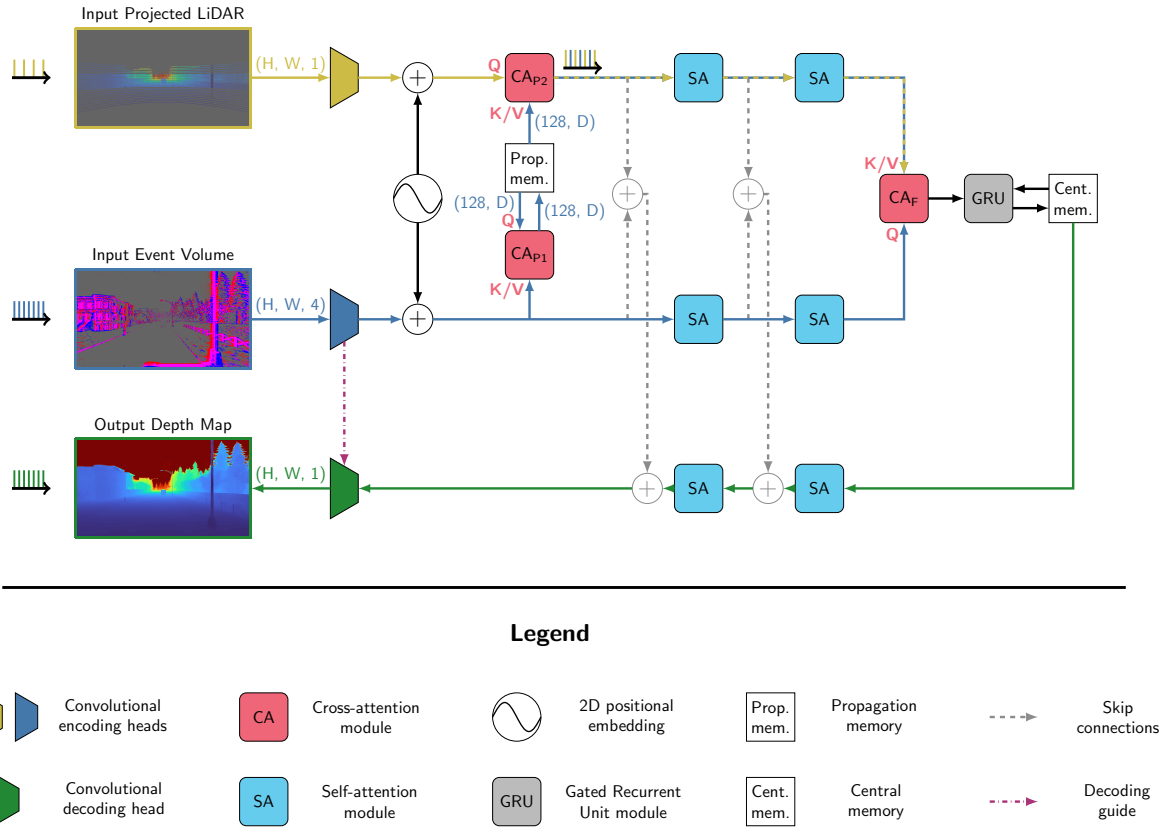

Figure 2. The complete architecture of our DELTA network. Unless noted, data is of shape (N, D), where N is the number of patches, and D their dimensionality (please refer to Secs. 3.2 and 4.2 for more details).

ing from an iterative refinement. As illustrated in Fig. 2, our DELTA network is based on a U-Net architecture [34], with two input branches for frame-like representations of the LiDAR and event data, a propagation memory, a central memory state, and a decoding branch.

Encoding heads To be able to apply attention on them, the event volumes and the projected LiDAR data (both of shape (H, W, C), where C is the number of channels) are first split into N small patches of size $P \times P$, as originally proposed by Dosovitskiy *et al*. [12]. This splitting is performed through stacked convolutional layers, and results in data of shape (N, D), where N is the number of patches, and D is the dimensionality of each patch. As attention is an order-independent operation, these encoded patches are summed with a fixed 2-dimensional positional embedding, following the formulation of Carion *et al*. [4]. This way, each patch has its own unique signature, making the network able to distinguish them.

Data encoding and fusion Event and LiDAR patches each go through two self-attention modules, to encode their own internal relations. A cross-attention module (CA_F in Fig. 2) is then used to encode the cross-relations between the event and LiDAR patches, resulting in a single fused representation. This step is crucial for ensuring the accurate fusion of the two modalities, as shown in Sec. 4.6.

LiDAR propagation Due to the use of the cross-attention module CA_F, contrary to the models of [3, 16], the LiDAR and event branches can not be totally decorrelated, as both LiDAR and event data are necessary at each time step. On the basis that event volumes are more frequently available than LiDAR point clouds, unless new LiDAR data is available, we propagate at each time step the previous LiDAR data using the incoming event data. To do so, the input events update a small (Sec. 4.2) propagation memory via a cross-attention module (CA_{P1} in Fig. 2). This propagation memory is then used in a second cross-attention module (CA_{P2}), where the previous LiDAR data queries an update from the propagation memory in order to produce an updated LiDAR representation.

Memory update Considering the case where the event camera and the LiDAR could be placed on a dynamic platform (*e.g*., a road vehicle), then if that platform was to come

to a halt (*e.g.*, at an intersection or in a traffic jam), few to no event would be produced by the camera. As such, densifying the LiDAR would become a difficult task, as the events would not be able to provide any guiding information. To solve this issue, given the sequential nature of the inputs, we add a central memory. This way, even if the event camera and LiDAR become static, the network can still exploit the memory of the previous information from these sensors to derive accurate predictions. This memory state also allows for temporal smoothness of the output of the network (as shown in Sec. 4.6), which is crucial given the high noise in the event data. Regarding the implementation, the fused LiDAR and event data are given to a Gated Recurrent Unit (GRU) module [8], which updates this central memory.

Decoding To obtain the final depth map, the data from the central memory passes through two self-attention modules, each followed by a skip connection with the corresponding summed and normalized LiDAR and event data from the encoding branch. To obtain an image-like output, a final decoding head regroups the decoded patches and reshapes the data to its original size. This decoding head is composed of stacked convex upsampling modules [42], where the upsampling is guided by the corresponding data from the events encoding head. Output is of shape $(H, W, 1)$, *i.e.*, the same spatial resolution as the inputs.

3.3. Loss Functions

To train DELTA, two complementary losses are used: a pixel-wise ℓ_1 loss $\mathcal{L}_{\ell_1}$, and a multiscale gradient-matching loss $\mathcal{L}_{\text{msg}}$ [44]. The role of $\mathcal{L}_{\ell_1}$ is to ensure the correctness of the depth prediction of all pixels compared to the ground truth. However, an ℓ_1 loss alone tends to produce blurry predictions, so a second regulatory loss is required to produce sharper depth maps. This is the role of $\mathcal{L}_{\text{msg}}$, which ensures that the gradients of the predicted depth maps at different scales are consistent with the ones of the ground truth depth maps. We follow the formulation of [3] for $\mathcal{L}_{\text{msg}}$, with the number of scales set to 5. Our final loss $\mathcal{L}$ is a sum of these two losses over a sequence of length T:

$$\mathcal{L} = \sum_{t=0}^{T-1} (\mathcal{L}_{\ell_1}^t + \mathcal{L}_{\text{msg}}^t). \tag{1}$$

4. Evaluation

As a complement to this section, an additional ablation study, an analysis on computational complexity, and additional visual results are all available as Suppl. Material.

4.1. Datasets

To conduct our evaluation, we use in this work three datasets of the state of the art: the SLED dataset [3], the MVSEC dataset [49], and the M3ED dataset [5].

Redefined set	Recordings	Total length
Train	`penno_small_loop_day`; `rittenhouse_day`; `penno_small_loop_night`	8m02s
Val	`horse_day`; `ucity_small_loop_night`	8m59s
Test	`city_hall_day`; `city_hall_night`	9m43s

Table 1. M3ED sets used within this work.

SLED The SLED dataset [3] is a synthetic dataset recorded in 2023 using the CARLA simulator [11]. It contains 20 minutes of perfectly synchronized and calibrated driving data, composed of high spatial definition (1280×720) events and images, point clouds from a 40-channel LiDAR at 10Hz with a maximum range of 200m, and dense ground truth depth maps.

MVSEC The MVSEC dataset [49] was recorded in 2018. It contains 30 minutes of long outdoor driving sequences, with low spatial definition (346×260) events and images, point clouds from a 16-channel LiDAR at 20Hz with a maximum range of 100m, and semi-dense ground truth depth maps. However, the data is loosely synchronized, the calibration is approximate, and the ground truth depth maps are erroneous when there are moving objects in the scene. Yet, it remains the most popular dataset for depth estimation in the event community, making it an interesting benchmark.

M3ED The M3ED dataset [5] was recorded in 2023, and acts as an informal successor to the MVSEC dataset. It is composed (among others) of 110 minutes of outdoor driving sequences, with high spatial definition stereo events (1280×720) and images (1280×800), point clouds from a 64-channel LiDAR at 10Hz with a maximum range of 120m, and sparse ground truth depth maps. However, given the size of the dataset, and since LiDAR data is not provided for the test set, we can not use it as is. To still be able to provide insightful results, we subsampled the dataset and redefined the train/val/test sets as described in Tab. 1.

DSEC We also tried initially to use the popular DSEC dataset [18], but its ground truth depth maps are given at the timestamps of the RGB frames (and are thus not synchronized with the LiDAR point clouds)[1], unfortunately making them incompatible with the training and evaluation of a LiDAR-and-event fusion method like ours.

4.2. Implementation Details

Data representation The event data is split in temporal windows of fixed size $\Delta t = 50$ms, based on the rate of the ground truth of the three datasets in use. The events in each window are accumulated into an Event Volume of shape

[1] https://github.com/uzh-rpg/DSEC/issues/7#issuecomment-1416776152

$(H, W, 4)$, following the formulation of Zhu *et al*. [50]. The LiDAR point clouds are represented as their projection on the event camera's image plane. Pixels without any depth value are set to 0. Both the LiDAR projections and ground truth depth maps are normalized between 0.0 and 1.0, where 1.0 is the maximum LiDAR range in the dataset in use.

Data size The patch size P is set to 16 pixels for high-resolution data (SLED and M3ED), and 12 pixels for low-resolution (MVSEC) to better capture details. We use a standard dimensionality of $D = 1024$. During training, data is randomly cropped to a size of 512×512 pixels for the high-resolution SLED and M3ED datasets, and to 252×252 pixels for the low-resolution MVSEC dataset.

Memories size and initialization Since the role of the central memory is to condense past data, its shape must be the same as the data itself, *i.e*., (N, D). On the contrary, the propagation memory is only a parametric representation of how the LiDAR data should be propagated to match the current events. While its dimensionality is still constrained to D, its number of elements N can be tuned: we empirically chose a size of 128 in this work. As for their initialization, the central memory is initialized with a copy of the two-dimensional positional embedding, while the initial state of the propagation memory is learned.

Training details For training on the three datasets, we use the Adam optimizer [25] with batch size $B = 4$. When training from scratch on the SLED and the M3ED datasets, 100 and 50 epochs are used respectively, the initial learning rate is set to 10^{-4}, and it is decayed by $0.01^{1/99}$ and $0.01^{1/49}$ respectively after each epoch (in order to reach a learning rate of 10^{-6} at the last epoch). When training from scratch on the MVSEC dataset, 20 epochs are used, and the learning rate is set to a constant value of 10^{-4}. When fine-tuning on MVSEC or M3ED, 5 epochs are used, and the learning rate is set to a constant value of 10^{-5}.

Evaluation metrics For all datasets, we use the mean absolute depth error metric [3, 10, 16, 33, 36], but also the standard depth estimation metrics of [13] (AbsRel, RMS, RMSlog, $\delta^{1,2,3}$). Following the convention on the MVSEC dataset [49], results are presented with five cutoff distances (10m, 20m, 30m, half the maximum range, and the maximum range). DELTA is trained on the full depth maps, *i.e*., at the maximum range, and these cutoffs are only applied at test time. For fairness of evaluation, comparisons with ALED [3] are only made on its $\mathrm{D_{bf}}$ depth maps.

4.3. Results on the SLED Dataset

We begin by training DELTA solely on the SLED dataset, and denote this version $\mathrm{DELTA_{SL}}$. Results of $\mathrm{DELTA_{SL}}$ on the testing set of SLED are given in Tab. 2. Compared to the state-of-the-art results of $\mathrm{ALED_{SL}}$ [3] (also trained only on SLED, noted $\mathrm{ALED_S}$ in [3]), a clear improvement can be seen across all metrics. Improvement is particularly important for nearby objects, as the mean depth error at the 10m cutoff is divided nearly by 2 and by 4 for the `Town01` and `Town03` maps respectively. Improvement is less significant at longer ranges, but we argue that close surroundings of the vehicle are of much more importance when driving than distant objects.

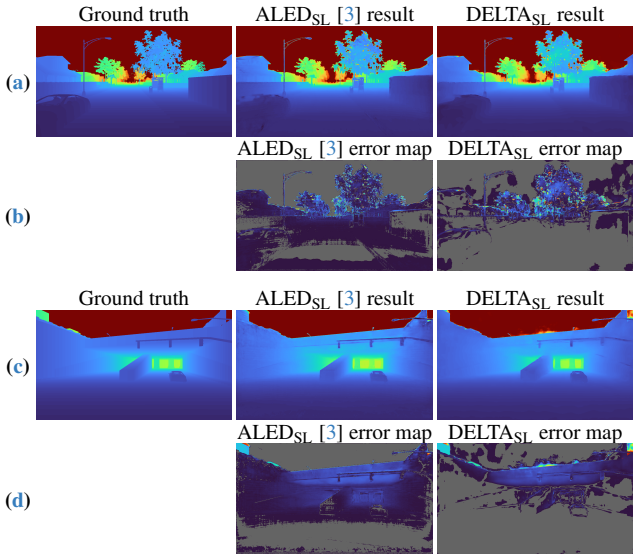

Figure 3. Results on the `Town01_08` ((a), (b)) and `Town03_19` ((c), (d)) sequences of SLED. Rows (a) and (c), left to right: ground truth depth map; result from $\mathrm{ALED_{SL}}$; our result ($\mathrm{DELTA_{SL}}$). Differences between $\mathrm{ALED_{SL}}$ and $\mathrm{DELTA_{SL}}$ are better seen in rows (b) and (d), showing the error maps of $\mathrm{ALED_{SL}}$ and $\mathrm{DELTA_{SL}}$ (where pixels with an error inferior to 0.5m are in gray). *For a better visualization, an enlarged version of this figure is given in the Supplementary Material.*

Visual results are presented in Fig. 3. Looking at the predicted depth maps (rows (a) and (c)), our network is able to infer very accurate results, visually close to the ground truth. Looking at the error maps (rows (b) and (d)), they confirm the observations made on the quantitative results: we commit very small errors at close ranges (especially for the ground) and larger errors at longer ranges, while ALED commits medium to large errors over the whole depth map. These results also highlight that, despite a lack of LiDAR data for the road just in front of the vehicle (at the very bottom of the images), our network can accurately estimate depth for these areas thanks to its attention-based design, which can correlate all the patches of event and LiDAR data (compared to the convolutional-based design of ALED, which only operates on limited neighborhoods, and thus does not achieve good results for these areas).

Map	Cutoff	ALED$_{SL}$ [3]							DELTA$_{SL}$						
		Mean↓	AbsRel↓	RMS↓	RMSlog↓	δ^1↑	δ^2↑	δ^3↑	Mean↓	AbsRel↓	RMS↓	RMSlog↓	δ^1↑	δ^2↑	δ^3↑
Town01	10m	1.24	0.211	8.022	0.478	0.890	0.962	0.978	**0.66**	**0.106**	**6.981**	**0.282**	**0.963**	**0.981**	**0.989**
	20m	2.10	0.232	11.973	0.489	0.885	0.952	0.970	**1.33**	**0.133**	**10.601**	**0.311**	**0.948**	**0.973**	**0.983**
	30m	2.74	0.239	13.831	0.481	0.875	0.945	0.966	**1.91**	**0.147**	**12.437**	**0.317**	**0.932**	**0.965**	**0.979**
	100m	4.26	0.241	17.182	0.468	0.860	0.935	0.959	**3.22**	**0.157**	**15.177**	**0.317**	**0.909**	**0.954**	**0.973**
	200m	**4.53**	0.173	**18.775**	0.405	0.892	0.947	0.966	4.54	**0.119**	19.606	**0.311**	**0.921**	**0.957**	**0.973**
Town03	10m	2.01	0.290	14.900	0.456	0.904	0.952	0.966	**0.54**	**0.082**	**5.656**	**0.184**	**0.958**	**0.973**	**0.986**
	20m	2.87	0.301	17.134	0.516	0.883	0.939	0.961	**1.31**	**0.117**	**9.958**	**0.234**	**0.934**	**0.962**	**0.980**
	30m	3.35	0.292	17.706	0.507	0.874	0.935	0.959	**1.93**	**0.133**	**11.831**	**0.251**	**0.919**	**0.952**	**0.974**
	100m	4.62	0.275	18.840	0.484	0.860	0.928	0.955	**3.40**	**0.146**	**15.119**	**0.265**	**0.897**	**0.941**	**0.968**
	200m	4.87	0.216	20.059	0.438	0.882	0.937	0.960	**4.63**	**0.122**	**19.363**	**0.271**	**0.906**	**0.944**	**0.967**

Table 2. Error metrics of ALED$_{SL}$ [3] and of DELTA$_{SL}$ on the SLED dataset for various cutoff depth distances. Best results are in bold.

4.4. Results on the MVSEC Dataset

For the MVSEC dataset, we propose two variants:

- DELTA$_{MV}$, only trained on MVSEC;
- DELTA$_{SL\to MV}$, after finetuning DELTA$_{SL}$ on MVSEC.

The results of both versions are given in Tabs. 3 and 4, in addition to the results of other methods of the state of the art. Compared to the results of all other methods (except ALED [3]), our method yields consistently lower errors, especially after pre-training on the SLED dataset. Compared to ALED$_{MV}$ (only trained on MVSEC, noted ALED$_{R}$ in [3]), DELTA$_{MV}$ offers a significant improvement, similar to the one observed on the SLED dataset in Sec. 4.3. Compared to ALED$_{SL\to MV}$ (pre-trained on SLED and finetuned on MVSEC, noted ALED$_{S\to R}$ in [3]), DELTA$_{SL\to MV}$ also offers an improvement for close ranges, and remains close for more distant ranges. Comparing DELTA$_{MV}$ and DELTA$_{SL\to MV}$ reveals however that the finetuning is not as efficient as between ALED$_{SL}$ and ALED$_{SL\to MV}$, with the mean depth error only improving slightly in our case. We believe this is due to the large change of resolution between the SLED and MVSEC datasets, which is a well-documented issue with attention-based vision models [14], and which requires in our case a change in patch size and a redefinition of the positional encoding. This makes the finetuning naturally more complex for an attention-based network than for a purely convolutional one. Additionally, as showcased in Fig. 4, fewer ground truth points are available for the close ranges (where our method performs best), skewing the results in favor of ALED (which has more difficulties for close objects, as shown previously in Sec. 4.3).

Visual results are also presented in Fig. 4, comparing them with those of Cui *et al*. [10] and of ALED$_{SL\to MV}$ [3]. While the method of Cui *et al*. allows for a good sharpness in the image, it fails at producing smooth depth gradients (especially on the ground, where the delimitations between LiDAR scans are clearly visible), and it is limited to the vertical range of the LiDAR sensor. Comparing our results to those of ALED$_{SL\to MV}$, the visualizations reflect the observations made during the quantitative evaluation: our depth maps have a very good accuracy, but those of DELTA$_{SL\to MV}$ are slightly less sharp than those of ALED$_{SL\to MV}$, while those of DELTA$_{MV}$ are much sharper but contain some small defects due to only learning on MVSEC (which contains erroneous ground truth values, as noted in Sec. 4.1). Like all other methods showcasing results on MVSEC [3, 16, 20], the lack of ground truth data for the sky leads to blue blobs in the upper areas of all predictions.

Recording	Cutoff	Events (stereo)	Events & Frames		Events & LiDAR				
		StereoSpike [33]	RAMNet [16]	EvT* [36]	Cui *et al.* [10]	ALED$_{MV}$ [3]	DELTA$_{MV}$	ALED$_{SL\to MV}$ [3]	DELTA$_{SL\to MV}$
outdoor_day_1	10m	0.79	1.39	1.24	1.24	0.91	0.51	**0.50**	**0.50**
	20m	1.47	2.17	1.91	1.28	1.22	0.86	**0.80**	0.81
	30m	1.92	2.76	2.36	4.87	1.43	1.10	**1.02**	1.06
	50m	-	-	-	-	1.67	1.42	**1.31**	1.41
	100m	3.17	-	-	-	1.96	1.73	**1.60**	1.80
outdoor_night_1	10m	**1.38**	2.50	1.45	2.26	1.75	1.55	1.52	1.52
	20m	2.26	3.19	2.10	2.19	2.10	1.94	**1.81**	1.92
	30m	2.97	3.82	2.88	4.50	2.25	2.14	**1.95**	2.16
	50m	-	-	-	-	2.44	2.42	**2.20**	2.46
	100m	4.82	-	-	-	2.73	2.78	**2.54**	2.88
outdoor_night_2	10m	-	1.21	1.48	1.88	1.19	1.16	1.09	**0.99**
	20m	-	2.31	2.13	2.14	1.65	1.59	**1.49**	**1.49**
	30m	-	3.28	2.90	4.67	1.81	1.77	**1.64**	1.75
	50m	-	-	-	-	1.95	1.95	**1.80**	1.98
	100m	-	-	-	-	2.11	2.16	**1.97**	2.23
outdoor_night_3	10m	-	1.01	1.38	1.78	0.85	0.92	0.81	**0.74**
	20m	-	2.34	2.03	1.93	1.25	1.35	**1.16**	1.26
	30m	-	3.43	2.77	4.55	1.42	1.57	**1.33**	1.56
	50m	-	-	-	-	1.57	1.78	**1.51**	1.83
	100m	-	-	-	-	1.73	1.96	**1.66**	2.09

Table 3. Mean depth errors (in meters) on the MVSEC dataset for various cutoff depth distances.

Recording	Metric	Events & Frames			Events & LiDAR			
		RAMNet [16]	HMNet [20]	PCDepth [28]	ALED$_{MV}$ [3]	DELTA$_{MV}$	ALED$_{SL\to MV}$ [3]	DELTA$_{SL\to MV}$
outdoor_day_1	AbsRel↓	0.303	0.230	0.228	0.185	0.118	**0.114**	0.121
	RMS↓	8.526	6.922	6.526	4.947	4.793	**4.574**	4.910
	RMSlog↓	0.424	0.310	0.301	0.259	0.215	**0.200**	0.236
	δ^1↑	0.541	0.717	0.712	0.834	0.890	**0.895**	0.866
	δ^2↑	0.778	0.868	0.867	0.937	0.953	**0.958**	0.954
	δ^3↑	0.877	0.940	0.941	0.970	0.977	**0.980**	**0.980**
outdoor_night_1	AbsRel↓	0.583	0.349	**0.271**	0.310	0.288	0.274	0.289
	RMS↓	13.340	10.818	6.715	6.122	**5.779**	5.791	6.042
	RMSlog↓	0.830	0.543	**0.354**	0.378	0.360	0.360	0.372
	δ^1↑	0.296	0.497	0.632	0.753	0.752	**0.767**	0.726
	δ^2↑	0.502	0.661	0.822	0.861	0.866	**0.870**	0.853
	δ^3↑	0.635	0.784	0.922	0.916	**0.925**	0.922	0.917
outdoor_night_2	AbsRel↓	-	-	-	0.198	0.195	**0.183**	0.184
	RMS↓	-	-	-	4.739	4.636	**4.543**	4.870
	RMSlog↓	-	-	-	0.282	0.281	**0.274**	0.281
	δ^1↑	-	-	-	0.802	0.809	**0.817**	0.785
	δ^2↑	-	-	-	0.911	0.913	**0.916**	0.909
	δ^3↑	-	-	-	0.958	0.959	**0.960**	0.959
outdoor_night_3	AbsRel↓	-	-	-	0.142	0.155	**0.133**	0.145
	RMS↓	-	-	-	3.861	4.053	**3.726**	4.514
	RMSlog↓	-	-	-	0.225	0.239	**0.221**	0.243
	δ^1↑	-	-	-	0.839	0.831	**0.849**	0.803
	δ^2↑	-	-	-	0.934	0.929	**0.937**	0.921
	δ^3↑	-	-	-	**0.973**	0.969	**0.973**	0.968

Table 4. Other error metrics on the MVSEC dataset (with the 100m cutoff).

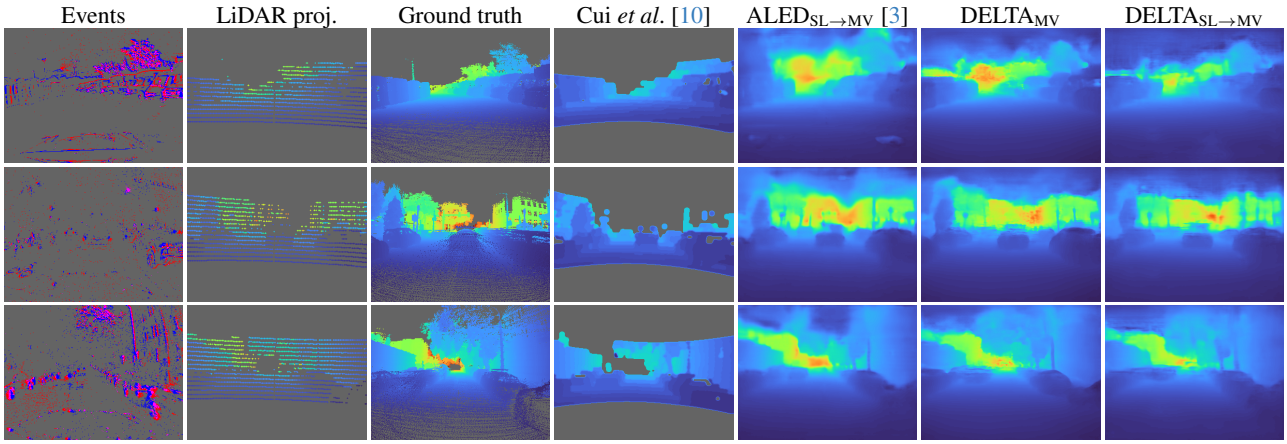

Figure 4. Qualitative results on the MVSEC dataset. Left to right: events; LiDAR projection (with larger points for a better readability); ground truth; results from Cui *et al*. [10]; results from ALED [3]; our results. Top to bottom: `outdoor_day_1`; `outdoor_night_1`; `outdoor_night_2`. *For a better visualization, an enlarged version of this figure is given in the Supplementary Material.*

4.5. Results on the M3ED Dataset

We propose two variants of DELTA on the M3ED dataset:

- DELTA$_{M3}$, only trained on M3ED;
- DELTA$_{SL\rightarrow M3}$, after finetuning DELTA$_{SL}$ on M3ED.

Results are given in Tab. 5, compared with those of ALED [3]. Again, DELTA performs very well on both sequences, with the mean error at the full cutoff distance being close to and lower than 1m for `city_hall_day`, and even lower for `city_hall_night`. This constitutes a significant improvement over ALED, as its mean error is always well over 1m for both recordings. Results on the other metrics also favor DELTA, although in a more even way. However, for both ALED and DELTA, pretraining on SLED brings little to no improvement. This apparent discrepancy can be explained through Fig. 5: the ground truth depth maps of M3ED are very sparse, with a similar density than the one of the LiDAR input, and with very few ground truth values at the edges of the objects. Because of these limitations, both ALED and DELTA showcase depth maps with accurate depth estimations, but give a blob-like appearance to all the objects as they could not learn the notion of edges from the training set. Pretraining on SLED allows for a better identification of these edges, producing sharper depth images, but it also leads to the appearance of artifacts (*e.g.*, on the streetlamp on the left side in Fig. 5).

Therefore, while M3ED remains a dataset of interest in the event-based community, we believe that it may not be the best suited for the training and/or evaluation of an event-and-LiDAR dense depth estimation method like ours.

Recording	Method	Mean↓	AbsRel↓	RMS↓	RMSlog↓	δ^1↑	δ^2↑	δ^3↑
`city_hall_day`	ALED$_{M3}$ [3]	1.41	0.074	2.867	0.120	0.958	0.986	**0.995**
	DELTA$_{M3}$	1.03	**0.054**	2.769	0.120	0.953	0.985	0.994
	ALED$_{SL\rightarrow M3}$ [3]	1.31	0.090	2.616	0.128	0.962	0.987	**0.995**
	DELTA$_{SL\rightarrow M3}$	**0.89**	0.055	**2.258**	**0.113**	**0.968**	**0.989**	**0.995**
`city_hall_night`	ALED$_{M3}$ [3]	1.41	0.068	2.499	0.102	**0.976**	**0.992**	**0.997**
	DELTA$_{M3}$	**0.82**	**0.042**	2.318	**0.099**	0.969	0.990	0.996
	ALED$_{SL\rightarrow M3}$ [3]	1.28	0.085	2.398	0.117	0.971	0.990	0.996
	DELTA$_{SL\rightarrow M3}$	0.85	0.048	**2.243**	0.103	0.974	0.990	0.995

Table 5. Errors of ALED [3] and DELTA on M3ED, without and with pretraining on SLED (with the maximum 120m cutoff). Results of ALED on M3ED were computed for this article.

4.6. Ablation Studies

In this final subsection, we investigate the importance of the two memories in the network, the role of the central cross-attention module CA$_F$, and the information provided by each of the input modalities. For that purpose, we propose five variants of DELTA:

- DELTA^{NPM}, with no propagation memory;
- DELTA^{NCM}, with no central memory;
- DELTA^{NCA}, with no central cross-attention module (replaced by a GRU module for each encoding branch to update the central memory);
- DELTA^{NL}, with no LiDAR input;
- DELTA^{NE}, with no event input.

We give an overview of the structure of these modified networks in the Suppl. Material. For the evaluation, as was done for Sec. 4.3, we trained these variants of DELTA on the SLED dataset, allowing for comparison with the proposed version of the network. The choice of doing this analysis on SLED was motivated by the fact that both the MVSEC and the M3ED datasets suffer from shortcomings in their respective ground truth, which could alter this ablation study.

Results for the first three variants are presented in Tab. 6. As can be seen, **(1)** the base variant DELTA$_{SL}$ produces the best results, except for the maximum 200m cutoff range, where it is only slightly beaten by the DELTA$_{SL}^{NCM}$ variant. **(2)** The version without a propagation memory DELTA$_{SL}^{NPM}$ has constantly an additional error of around 0.2m to 0.4m,

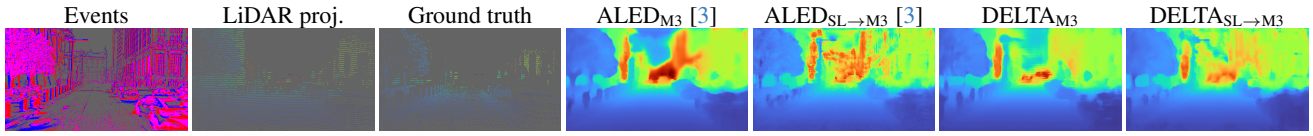

Figure 5. Qualitative results for the `city_hall_day` sequence from M3ED. The size of points was increased for both the LiDAR projection and the ground truth. *For a better visualization, an enlarged version of this figure is given in the Supplementary Material.*

Cutoff	DELTA$_{SL}$	DELTA$_{SL}^{NPM}$	DELTA$_{SL}^{NCM}$	DELTA$_{SL}^{NCA}$
10m	**0.60**	0.85 (+0.25)	0.91 (+0.31)	0.95 (+0.35)
20m	**1.32**	1.64 (+0.32)	1.71 (+0.39)	1.80 (+0.48)
30m	**1.92**	2.24 (+0.32)	2.21 (+0.29)	2.40 (+0.48)
100m	**3.32**	3.73 (+0.41)	3.53 (+0.21)	3.85 (+0.53)
200m	4.58	5.01 (+0.43)	**4.49** (−0.09)	5.03 (+0.45)

Table 6. Absolute and relative mean depth errors (in meters) on the full testing set of SLED, for alternative versions of DELTA (No Propagation Memory, No Central Memory, No Cross-Attention).

highlighting the importance of temporally propagating the LiDAR data with the events. **(3)** The version without the central memory DELTA$_{SL}^{NCM}$ also performs worse, albeit with an additional error that diminishes the greater the cutoff distance is, beating DELTA$_{SL}$ by 0.09m at the maximum cutoff range. It should be noted however that the car which the sensors are mounted on in the SLED dataset rarely stops, and if it does, it is for very short periods of time. As explained in Sec. 3.2, since the main role of the central memory is to help provide accurate results in these cases, it only serves here its secondary role of being a stabilization medium, which is still valuable given the numerical results. **(4)** The version without the central cross-attention module DELTA$_{SL}^{NCA}$ is the worst performing variant, as it has the largest error across all cutoff ranges. As such, the central cross-attention module CA$_F$ is crucial for encoding the cross-relations between the encoded LiDAR and event data before updating the central memory.

Finally, results for the DELTA^{NL} and DELTA^{NE} variants are given in Fig. 6. They highlight the complementarity of the two inputs: the events help identify the edges of the objects but not their depth (DELTA^{NL}), while the LiDAR helps identify the depth but not the edges of the objects, especially those above or under its narrow vertical FOV (DELTA^{NE}).

5. Conclusion and Discussions

In this article, a new attention-based network for fusing projected LiDAR data and temporal windows of events to construct dense depth maps was presented. Thanks to the introduction of a propagation memory between cross-attentions, DELTA is able to extrapolate LiDAR with events at higher rate for an optimal fusion. A GRU is also added between the encoding and decoding stages, allowing for a short-term central memory and more robust outputs. A thorough evaluation including ablation studies was conducted on three datasets of the state of the art to demonstrate the relevance of these propositions. On the synthetic SLED dataset, large improvements are achieved, with for instance the mean error being reduced up to four times for short ranges when compared to the state of the art. On MVSEC and M3ED, despite limitations in the ground truth data of these datasets, DELTA remains very competitive, being close to or achieving the state of the art, with low errors across all cutoff ranges. We believe in particular that our new state-of-the-art architecture can serve as a basis for other applications or for fusion with other sensors.

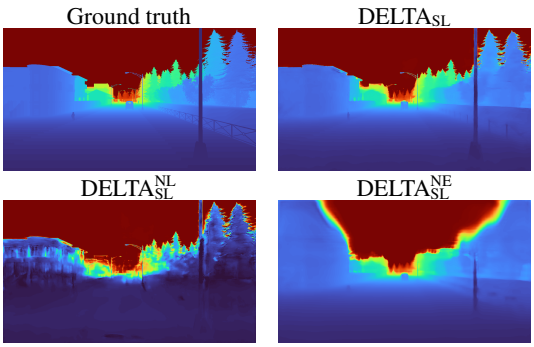

Figure 6. Qualitative results for the DELTA^{NL} and DELTA^{NE} variants, on the `Town01_19` sequence of SLED.

In hindsight, some modifications could still be brought to this work in order to further improve its overall performance. **(1)** As shown throughout this article, the reduction of errors over short ranges is sometimes done at the cost of lower precision at longer ranges. Adapting our training procedure by introducing variable weights for every cutoff distance could be a solution to make sure all of them are optimized equally. **(2)** We chose to use the event volume [50], as it is a standard, compact, and high-performing representation of event data. While it allows DELTA to achieve very accurate results, this representation can lead to some information being lost in case of very fast motion. Novel representations [1, 15, 51] could be examined for these cases. **(3)** Finally, projecting the LiDAR data allows for a pixel-wise fusion with the event data, but makes the three-dimensional structure of the point cloud disappear. Working directly in 3D like [10] could therefore constitute an interesting research opportunity.

Acknowledgment

This work was supported in part by the Hauts-de-France Region and in part by the SIVALab Joint Laboratory (Renault Group – Université de technologie de Compiègne (UTC) – Centre National de la Recherche Scientifique (CNRS)). This work was also supported by The Royal Physiographic Society in Lund.

References

[1] R. W. Baldwin, Ruixu Liu, Mohammed Almatrafi, Vijayan K. Asari, and Keigo Hirakawa. Time-ordered recent event (TORE) volumes for event cameras. *IEEE TPAMI*, 45: 2519–2532, 2021. 8

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

[3] Vincent Brebion, Julien Moreau, and Franck Davoine. Learning to estimate two dense depths from LiDAR and event data. In *Image Analysis - 22nd Scandinavian Conference, SCIA 2023, Sirkka, Finland, April 18-21, 2023, Proceedings, Part II*, pages 517–533. Springer, 2023. 2, 3, 4, 5, 6, 7, 8

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[5] Kenneth Chaney, Fernando Cladera Ojeda, Ziyun Wang, Anthony Bisulco, M. A. Hsieh, Christopher M. Korpela, Vijay R. Kumar, Camillo Jose Taylor, and Kostas Daniilidis. M3ED: Multi-robot, multi-sensor, multi-environment event dataset. *CVPRW*, pages 4016–4023, 2023. 2, 4

[6] Stefano Chiavazza, Svea Marie Meyer, and Yulia Sandamirskaya. Low-latency monocular depth estimation using event timing on neuromorphic hardware. *CVPRW*, pages 4071–4080, 2023. 2

[7] Hoonhee Cho, Jegyeong Cho, and Kuk-Jin Yoon. Learning adaptive dense event stereo from the image domain. *CVPR*, pages 17797–17807, 2023. 2

[8] 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 *Conference on Empirical Methods in Natural Language Processing*, 2014. 4

[9] Mathieu Cocheteux, Julien Moreau, and Franck Davoine. MULi-Ev: Maintaining unperturbed LiDAR-event calibration. *CVPRW*, pages 4579–4586, 2024. 2

[10] Mingyue Cui, Yuzhang Zhu, Yechang Liu, Yun-Meng Liu, Gang Chen, and Kai Huang. Dense depth-map estimation based on fusion of event camera and sparse LiDAR. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022. 2, 5, 6, 7, 8, 1, 4

[11] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 4

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3, 1

[13] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 5

[14] Kerr Fitzgerald, Meng Law, Jarrel C. Y. Seah, Jennifer Tang, and Bogdan J. Matuszewski. Multi-resolution fine-tuning of vision transformers. In *Annual Conference on Medical Image Understanding and Analysis*, 2022. 6

[15] Daniel Gehrig, Antonio Loquercio, Konstantinos G. Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. *ICCV*, pages 5632–5642, 2019. 8

[16] Daniel Gehrig, Michelle Rüegg, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction. *IEEE Robotics and Automation Letters*, 6:2822–2829, 2021. 2, 3, 5, 6

[17] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. *CVPR*, pages 13884–13893, 2023. 2

[18] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6: 4947–4954, 2021. 2, 4

[19] Suman Ghosh and Guillermo Gallego. Multi-event-camera depth estimation and outlier rejection by refocused events fusion. *Advanced Intelligent Systems*, 4, 2022. 2

[20] Ryuhei Hamaguchi, Yasutaka Furukawa, Masaki Onishi, and Ken Sakurada. Hierarchical neural memory network for low latency event processing. *CVPR*, pages 22867–22876, 2023. 2, 6

[21] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events. *3DV*, pages 534–542, 2020. 2

[22] Jianhao Jiao, Feiyi Chen, He Wei, Jin Wu, and Mingming Liu. LCE-Calib: Automatic LiDAR-frame/event camera extrinsic calibration with a globally optimal solution. *ArXiv*, abs/2303.09825, 2023. 2

[23] Uday Kamal, Saurabh Dash, and S. Mukhopadhyay. Associative memory augmented asynchronous spatiotemporal representation learning for event-based perception. In *ICLR*, 2023. 2

[24] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-time 3D reconstruction and 6-DoF tracking with an event camera. In *ECCV*, 2016. 2

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 5

[26] Boyang Li, Hao Meng, Yuzhang Zhu, Rihui Song, Mingyue Cui, Gang Chen, and Kai Huang. Enhancing 3-D LiDAR point clouds with event-based camera. *IEEE Transactions on Instrumentation and Measurement*, 70:1–12, 2021. 2

[27] Zhihao Li, M. Salman Asif, and Zhan Ma. Event transformer. *ArXiv*, abs/2204.05172, 2022. 2

[28] Haotian Liu, Sanqing Qu, Fan Lu, Zongtao Bu, Florian Roehrbein, Alois Knoll, and Guang Chen. PCDepth: Pattern-based complementary learning for monocular depth estimation by best of both worlds. *ArXiv*, abs/2402.18925, 2024. 2, 6

[29] Xu Liu, Jianing Li, Xiaopeng Fan, and Yonghong Tian. Event-based monocular dense depth estimation with recurrent transformers. *ArXiv*, abs/2212.02791, 2022. 2

[30] Yeongwoo Nam, Sayed Mohammad Mostafavi Isfahani, Kuk-Jin Yoon, and Jonghyun Choi. Stereo depth from events cameras: Concentrate and focus on the future. *CVPR*, pages 6104–6113, 2022. 2

[31] Urbano Miguel Nunes, Laurent Udo Perrinet, and Sio-Hoi Ieng. Time-to-contact map by joint estimation of up-to-scale inverse depth and global motion using a single event camera. *ICCV*, 2023. 2

[32] Yansong Peng, Yueyi Zhang, Zhiwei Xiong, Xiaoyan Sun, and Feng Wu. GET: Group event transformer for event-based vision. *ICCV*, 2023. 2

[33] Ulysse Rançon, Javier Cuadrado-Anibarro, Benoit R. Cottereau, and Timothée Masquelier. Stereospike: Depth learning with a spiking neural network. *IEEE Access*, 10: 127428–127439, 2021. 2, 5, 6

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing, 2015. 3

[35] Alberto Sabater, Luis Montesano, and Ana Cristina Murillo. Event transformer. A sparse-aware solution for efficient event data processing. *CVPRW*, pages 2676–2685, 2022. 2

[36] Alberto Sabater, Luis Montesano, and Ana Cristina Murillo. Event transformer+. A multi-purpose solution for efficient event data processing. *ArXiv*, abs/2211.12222, 2022. 2, 5, 6

[37] Mario A. V. Saucedo, Akash Patel, Rucha Sawlekar, Akshit Saradagi, Christoforos Kanellakis, Ali akbar Aghamohammadi, and George Nikolakopoulos. Event camera and lidar based human tracking for adverse lighting conditions in subterranean environments. *ArXiv*, abs/2304.08908, 2023. 2

[38] Stephan Schraml, Ahmed Nabil Belbachir, Nenad Milosevic, and Peter Schön. Dynamic stereo vision system for real-time tracking. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 1409–1412, 2010. 2

[39] Stephan Schraml, Ahmed Nabil Belbachir, and Horst Bischof. An event-driven stereo system for real-time 3-D 360° panoramic vision. *IEEE Transactions on Industrial Electronics*, 63:418–428, 2016. 2

[40] Rihui Song, Zhihua Jiang, Yanghao Li, Yunxiao Shan, and Kai Huang. Calibration of event-based camera and 3D LiDAR. *2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, pages 289–295, 2018. 2

[41] Kevin Ta, David Brüggemann, Tim Brödermann, Christos Sakaridis, and Luc Van Gool. L2E: Lasers to events for 6-DoF extrinsic calibration of lidars and event cameras. *ICRA*, pages 11425–11431, 2023. 2

[42] Zachary Teed and Jun Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020. 4, 1

[43] Yi Tian and J. Andrade-Cetto. Event transformer flownet for optical flow estimation. In *BMVC*, 2022. 2

[44] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. DeMoN: Depth and motion network for learning monocular stereo. *CVPR*, pages 5622–5631, 2016. 4

[45] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[46] Zuowen Wang, Yuhuang Hu, and Shih-Chii Liu. Exploiting spatial sparsity for event cameras with visual transformers. *ICIP*, pages 411–415, 2022. 2

[47] David Weikersdorfer, David B. Adrian, Daniel Cremers, and Jörg Conradt. Event-based 3D SLAM with a depth-augmented dynamic vision sensor. *ICRA*, pages 359–364, 2014. 2

[48] Wenming Weng, Yueyi Zhang, and Zhiwei Xiong. Event-based video reconstruction using transformer. *ICCV*, pages 2543–2552, 2021. 2

[49] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay R. Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters*, 3:2032–2039, 2018. 2, 4, 5

[50] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. *CVPR*, pages 989–997, 2019. 2, 5, 8

[51] Nikola Zubić, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. From chaos comes order: Ordering event representations for object recognition and detection. *ICCV*, 2023. 8

DELTA: Dense Depth from Events and LiDAR using Transformer's Attention

Supplementary Material

6. Enlarged Views of the Results on SLED, MVSEC, and M3ED

As described in the main article, an enlarged version of Fig. 3 is given in Fig. 7 (this version also includes the input LiDAR and event data), an enlarged version of Fig. 4 is given in Fig. 8, and an enlarged version of Fig. 5 is given in Fig. 9.

7. Alternative Versions of DELTA

Alternative versions of our DELTA network are given in Figs. 10 to 15. Versions of the network illustrated in Figs. 10 to 14 are used as part of the ablation study in Sec. 4.6 of the main article, while the version illustrated in Fig. 15 is used as part of Sec. 8 of this Supplementary Material.

8. Ablation Study on Encoding Heads

In addition to the ablation studies conducted in the main article, we propose here an additional variant of the network, DELTA$^{\text{NEH}}$, showcased in Fig. 15. Here, the convolutional encoding heads are replaced by a more direct splitting into patches, as originally done in the Vision Transformer [12]. To compensate for the reduced number of parameters in the network, we add a third layer of self-attention modules. At the end of the decoding, the patches are simply grouped back to an image-based format, and a final small convolutional head reduces the number of channels and smooths the resulting depth maps.

Results of DELTA$^{\text{NEH}}$ on the SLED dataset are shown in Tab. 7. As can be seen, DELTA$_{\text{SL}}^{\text{NEH}}$ does not perform well at all, even worse than all the variants showcased in the main article. As visually illustrated in Fig. 16, DELTA$_{\text{SL}}^{\text{NEH}}$ produces depth maps with large errors (especially for the tunnel in the bottom row), and where the junction between the patches remains visible, creating numerous artifacts. While a simple splitting into patches can be conducted for the Vision Transformer, as classification is the end task, here we require a dense reconstruction at the end, *i.e.*, we need to keep information about the structure of the scene. Therefore, in our case, computing the patches using a convolutional head allows for a better re-grouping of the patches at the end of the network, by allowing the decoding head to be guided by the corresponding data from the encoding head through our use of the convex upsampling module of [42].

9. Computational Complexity

We report in Tab. 8 several metrics of the computational complexity of DELTA, computed on a single NVIDIA L40 GPU. For high- (1280×720), mid- (640×480), and low-resolution (346×260) data, DELTA has a mean inference rate of 6.3Hz, 20.5Hz, and 47.8Hz respectively. Compared to the method of Cui *et al*. [10] with its reported output rate of 56Hz on the MVSEC dataset, our method is only 1.17 times slower, but for a much better accuracy as shown in Tab. 3 of the main article. Compared to ALED, despite the significant increase in the number of parameters due to the use of attention modules, DELTA requires a similar amount of FLOPS and of GPU memory, allowing its deployment on standard consumer-grade GPUs. Inference times of ALED are of course smaller, and while we can not exactly call our method real-time, we want to remind the reader here that the focus of our work was set on accuracy rather on real-time compatibility. As such, inference time and/or memory usage could be further reduced, as we are not using advanced optimizations like `torch.compile()`, and as we believe the DELTA architecture could be slightly revised to reduce its number of parameters while keeping a similar accuracy. Implementation on specialized hardware could also be considered for real-time robotic applications, but is beyond the scope of this work.

Map	Cutoff	DELTA$_{\text{SL}}$	DELTA$_{\text{SL}}^{\text{NEH}}$
Town01	10m	**0.66**	1.58 (+0.92)
	20m	**1.33**	2.89 (+1.56)
	30m	**1.91**	3.90 (+1.99)
	100m	**3.22**	6.73 (+3.51)
	200m	**4.54**	9.85 (+5.41)
Town03	10m	**0.54**	2.13 (+1.59)
	20m	**1.31**	3.17 (+1.86)
	30m	**1.93**	3.89 (+1.96)
	100m	**3.40**	6.14 (+2.74)
	200m	**4.63**	9.23 (+4.60)

Table 7. Absolute and relative mean depth errors (in meters) on the SLED dataset, for the base version of DELTA and the "No Encoding Head" variant shown in Fig. 15.

10. Additional Visual Results on the SLED Dataset

Additional qualitative results on the SLED dataset are given in Figs. 17 to 20. We showcase in Figs. 17 and 18 scenes with accurate estimations, but also some small and larger failure cases in Figs. 19 and 20.

Model	Resolution (with padding)	Dataset(s)	Patch size	Nb. param.	FLOPS	Inference time	Max. GPU mem.
DELTA	1280 × 720	SLED, M3ED	16	180.9M	1786.1B	157.9ms ± 2.0ms	10.64 GB
	640 × 480	DSEC	16	180.9M	596.4B	48.7ms ± 0.4ms	4.68 GB
	346 × 260 (348 × 264)	MVSEC	12	181.2M	300.4B	20.9ms ± 0.2ms	3.09 GB
ALED	1280 × 720	SLED, M3ED	/	26.2M	1546.0B	91.2ms ± 16.8ms	7.02 GB
	640 × 480	DSEC	/	26.2M	515.3B	26.9ms ± 5.0ms	2.74 GB
	346 × 260 (352 × 264)	MVSEC	/	26.2M	155.9B	7.5ms ± 1.4ms	1.36 GB

Table 8. Computational complexity metrics (number of parameters, FLOPS, mean inference time, maximum GPU memory usage) for DELTA and ALED, for both high-, mid-, and low-resolution data.

11. Additional Visual Results on the MVSEC Dataset

Additional qualitative results on the MVSEC dataset are given in Fig. 21, showing the quality of the results for both day and night scenes despite the sparse and low-resolution input event and LiDAR data.

12. Additional Visual Results on the M3ED Dataset

Additional qualitative results on the M3ED dataset are given in Figs. 22 and 23, where the sparsity of the ground truth depth maps (especially compared to the density of the LiDAR data) and the blob-like appearance of the objects in the predictions can be better observed.

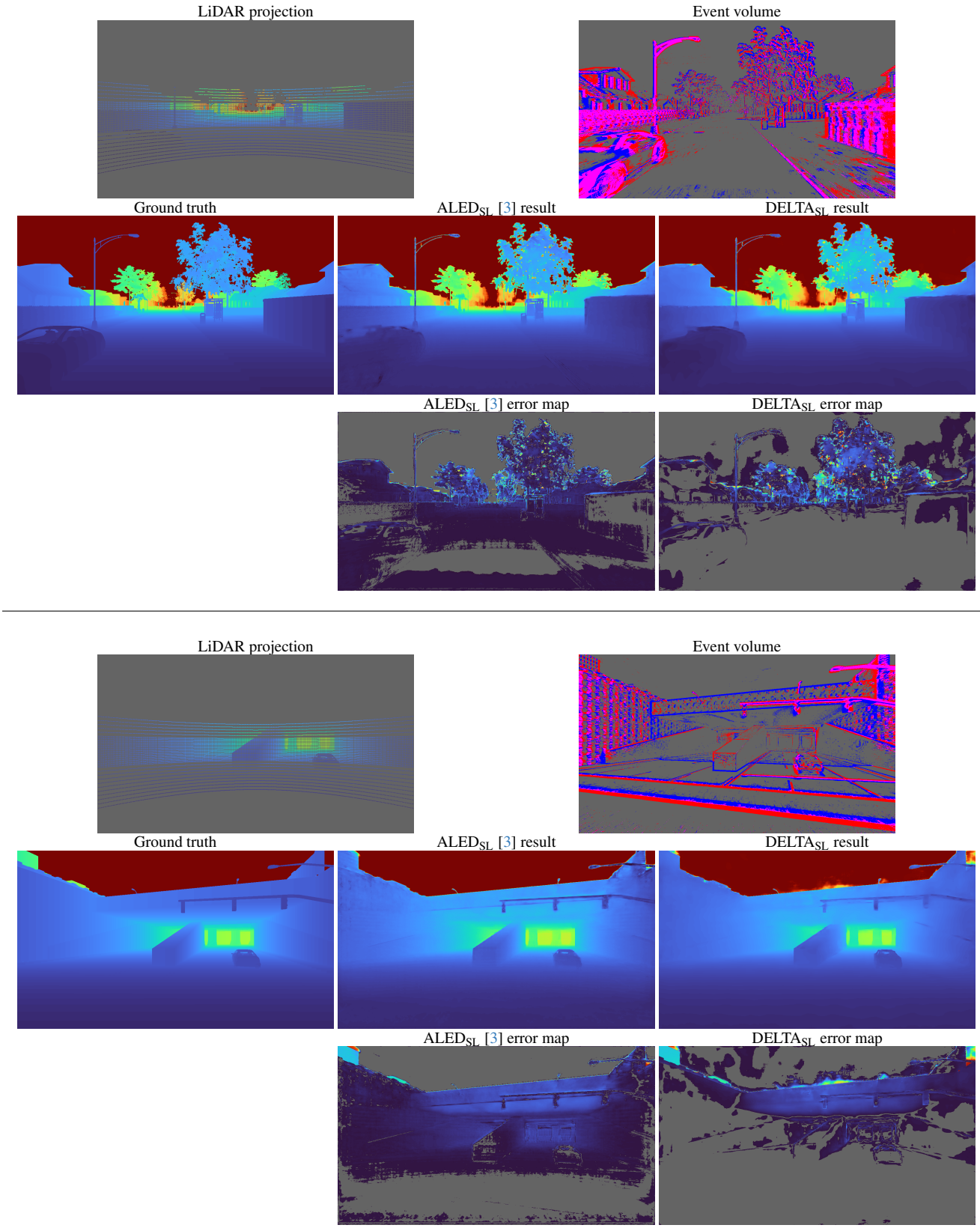

Figure 7. Comparison on the `Town01_08` (top) and `Town03_19` (bottom) sequences of SLED (enlarged version of Fig. 3).

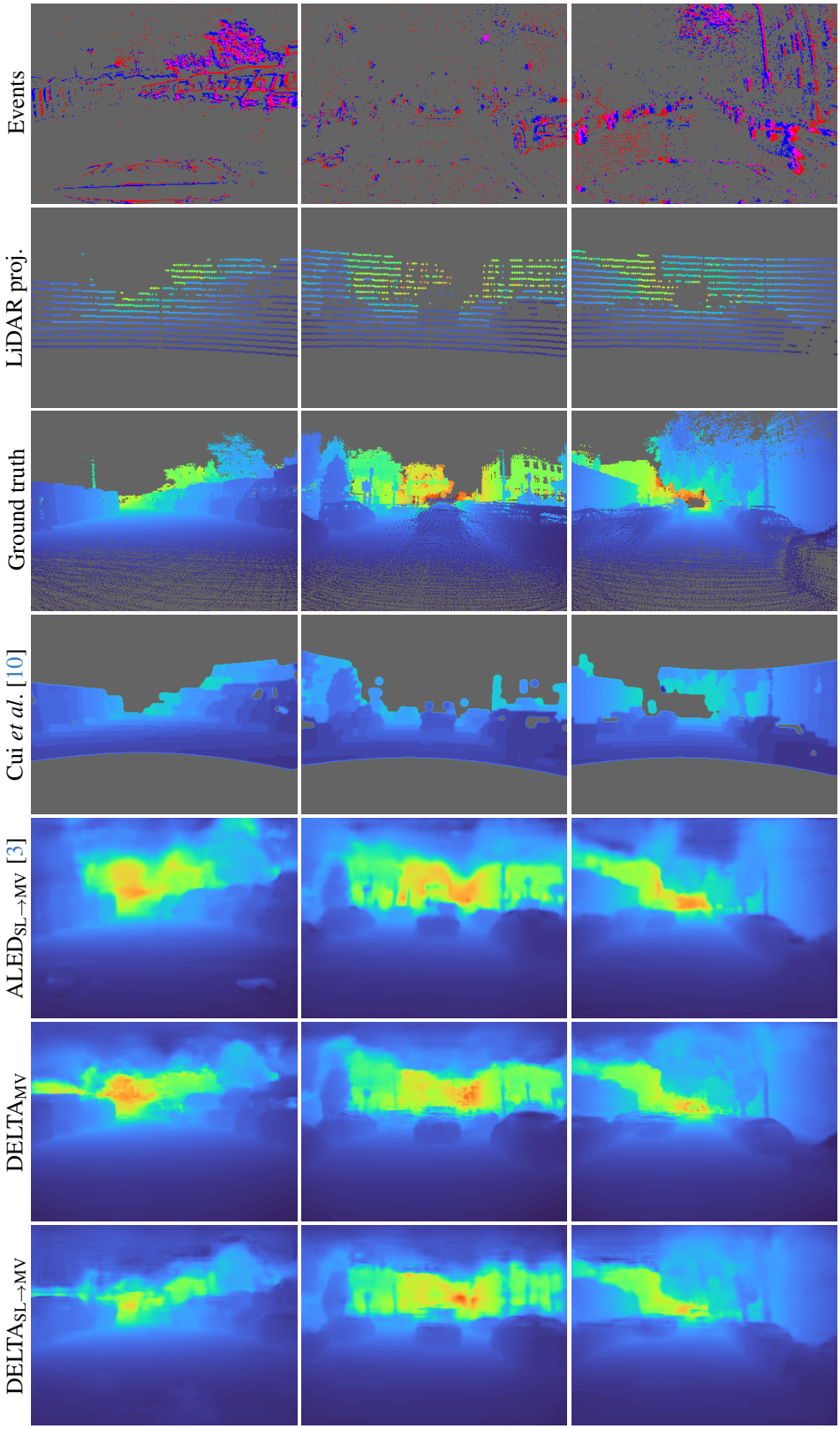

Figure 8. Comparison on the `outdoor_day_1`, `outdoor_night_1`, and `outdoor_night_2` sequences of MVSEC (enlarged version of Fig. 4).

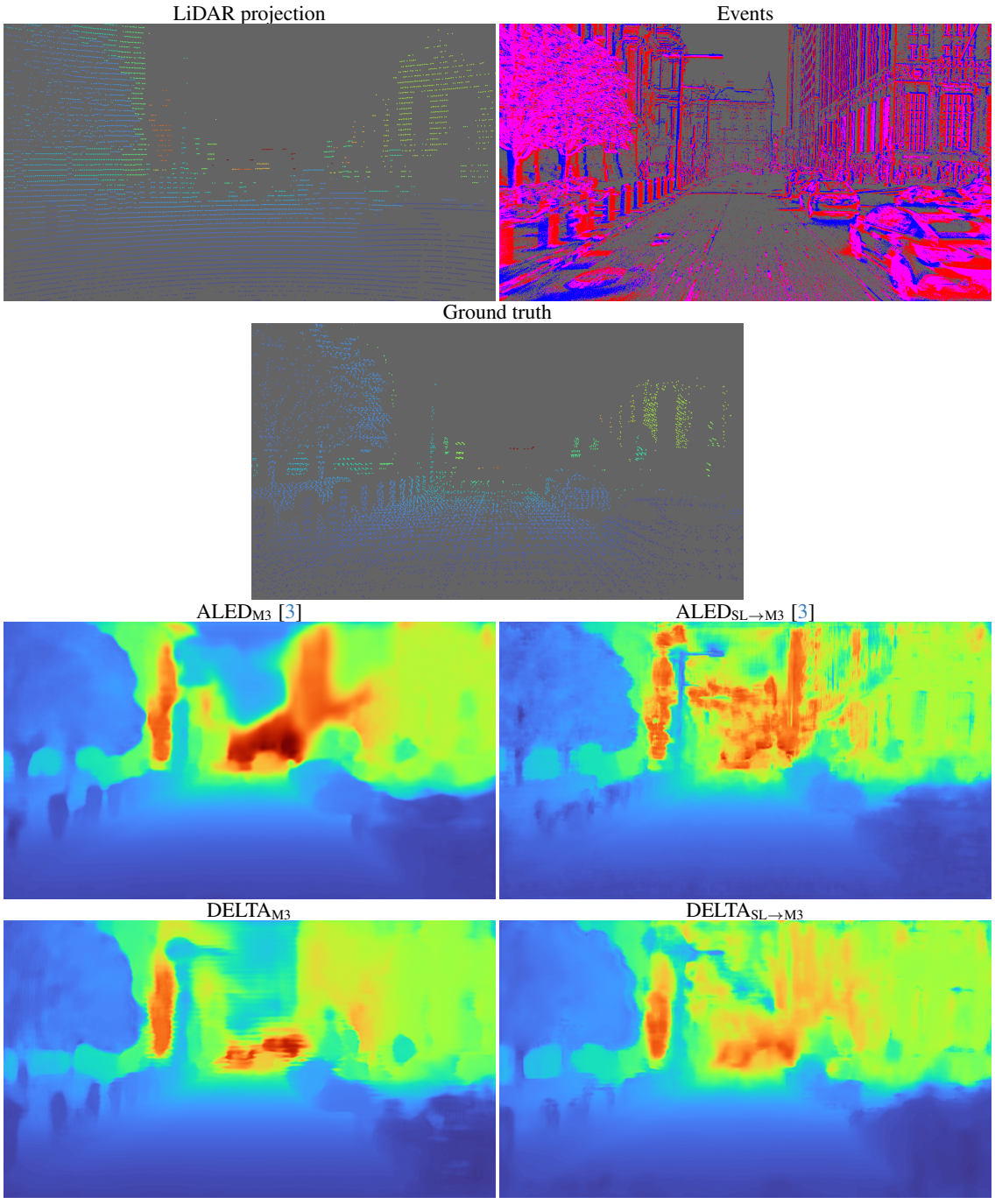

Figure 9. Comparison on the `city_hall_day` sequence of M3ED (enlarged version of Fig. 5).

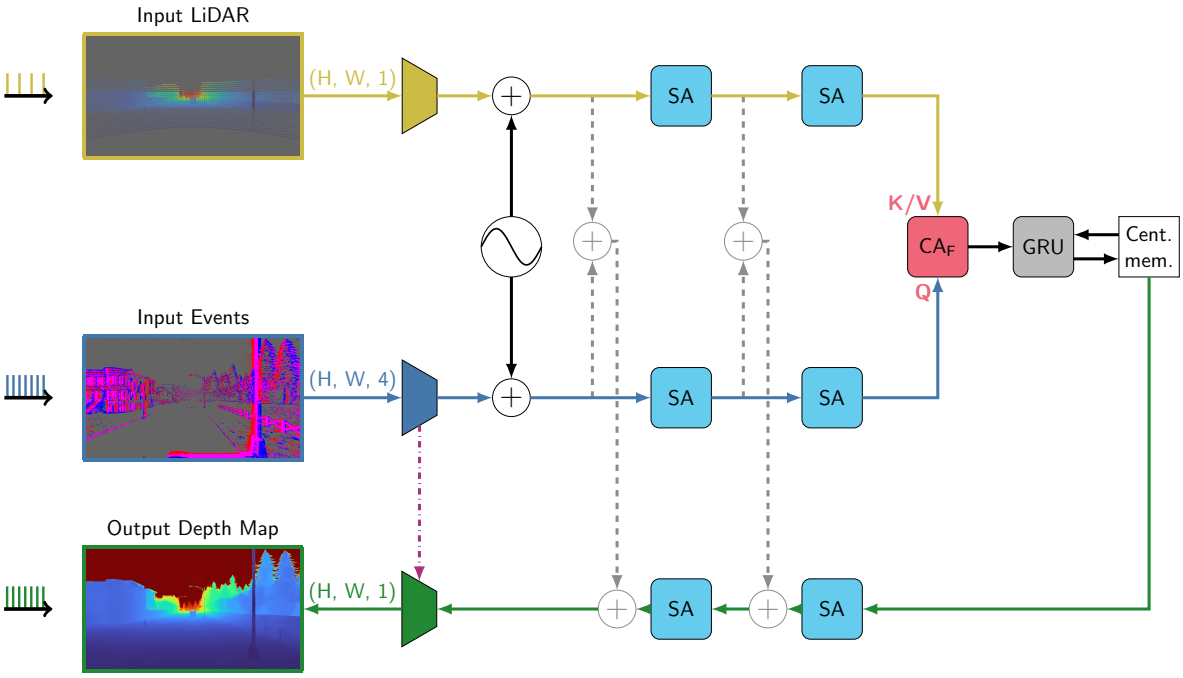

Figure 10. The alternative architecture without propagation memory, DELTA^{NPM}.

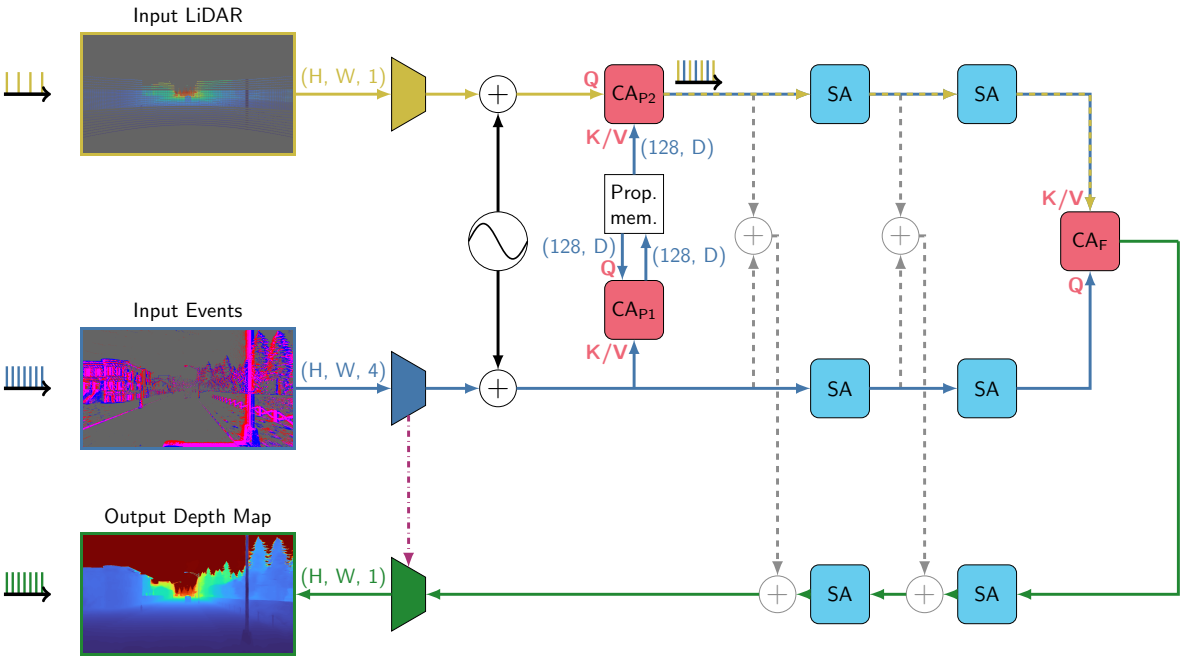

Figure 11. The alternative architecture without central memory, DELTA^{NCM}.

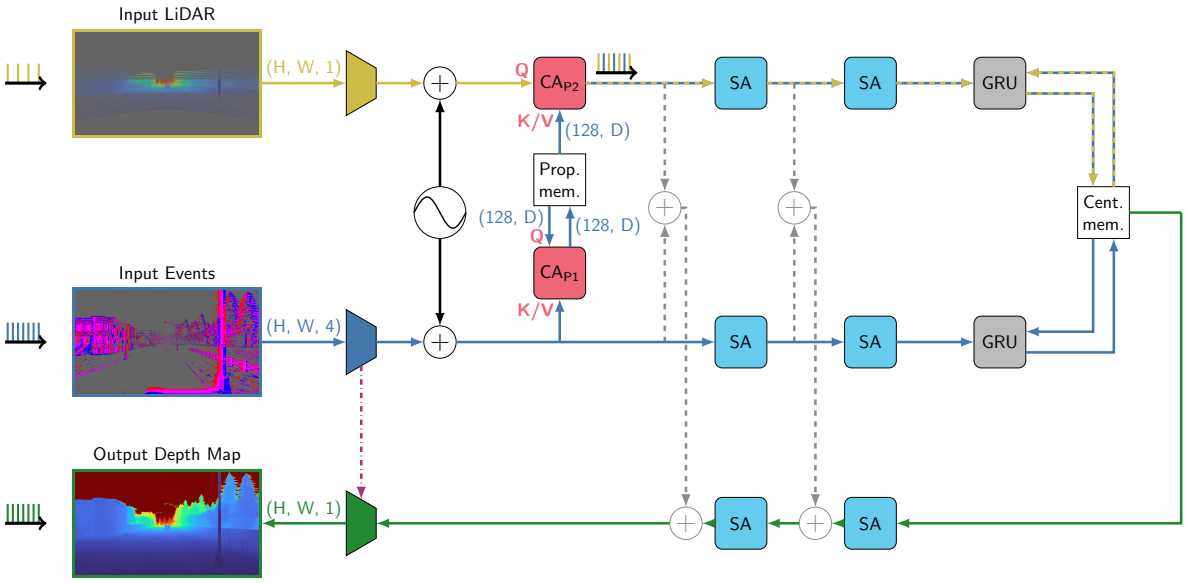

Figure 12. The alternative architecture without the central cross-attention, DELTA^{NCA}.

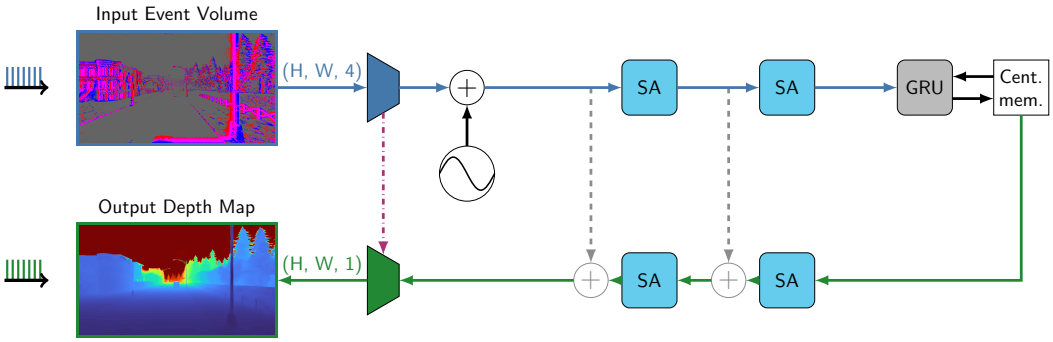

Figure 13. The alternative architecture without the LiDAR input, DELTA^{NL}.

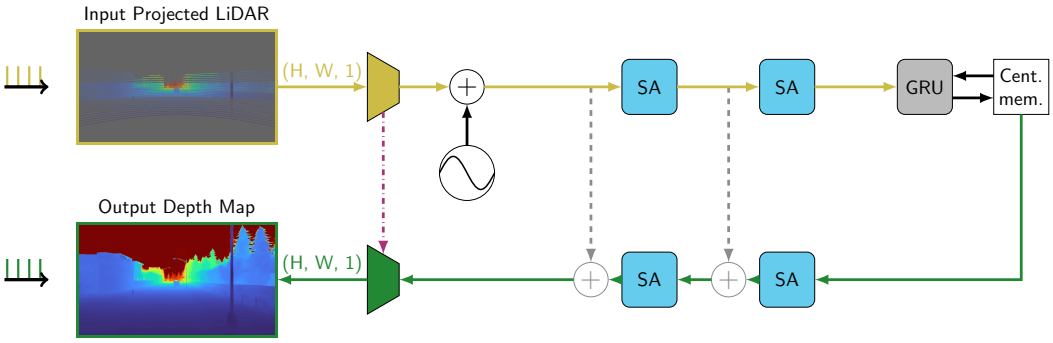

Figure 14. The alternative architecture without the event input, DELTA^{NE}.

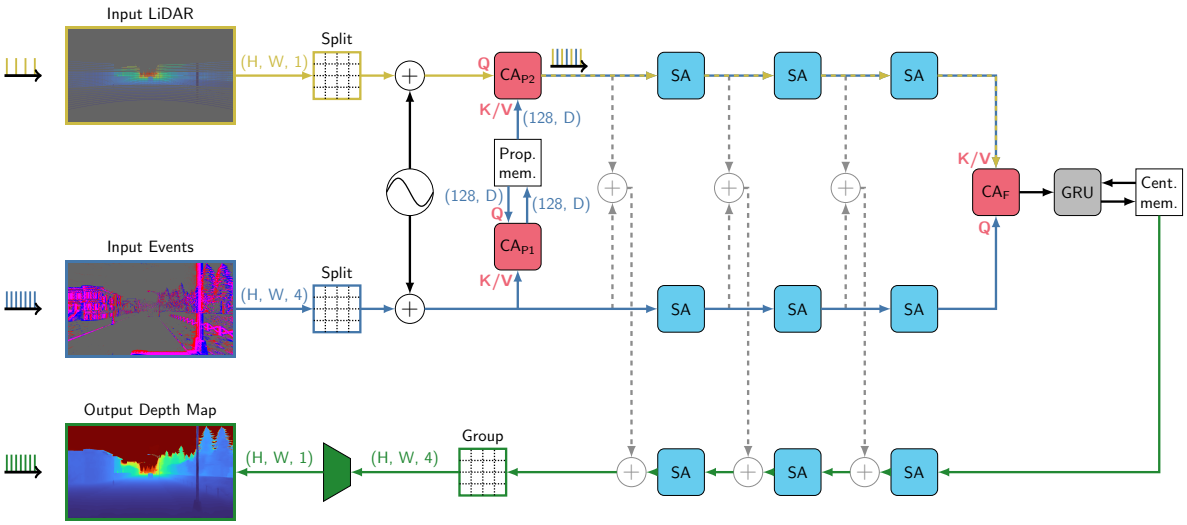

Figure 15. The alternative architecture where the convolutional encoding heads are replaced by a simple splitting into patches, DELTA$^{\text{NEH}}$.

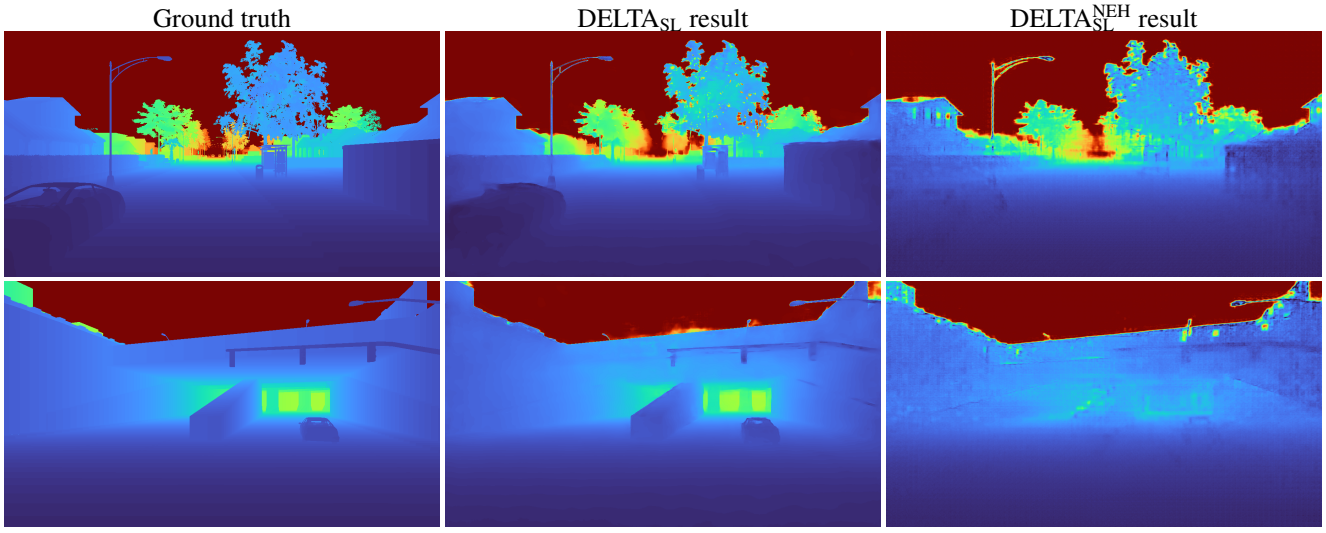

Figure 16. Results on the `Town01_08` (top) and `Town03_19` (bottom) sequences of SLED, for DELTA$_{SL}$ and DELTA$_{SL}^{NEH}$. Zoom on the numerical version may be required to better see the individual patches and artifacts for DELTA$_{SL}^{NEH}$.

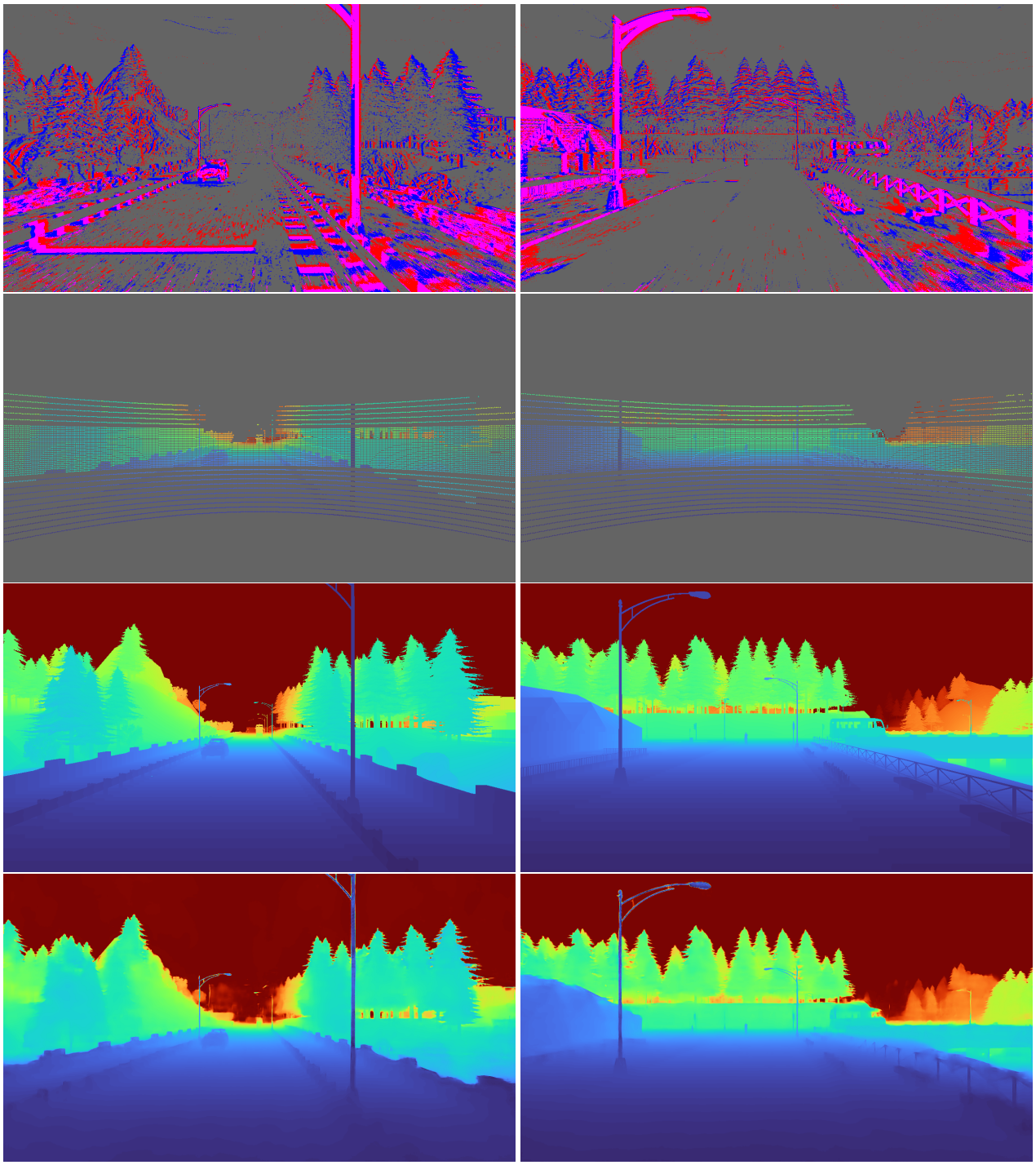

Figure 17. Additional results on the SLED dataset, on sequences `Town01_03` and `Town01_05`. From top to bottom: events, LiDAR projection, ground truth, our results.

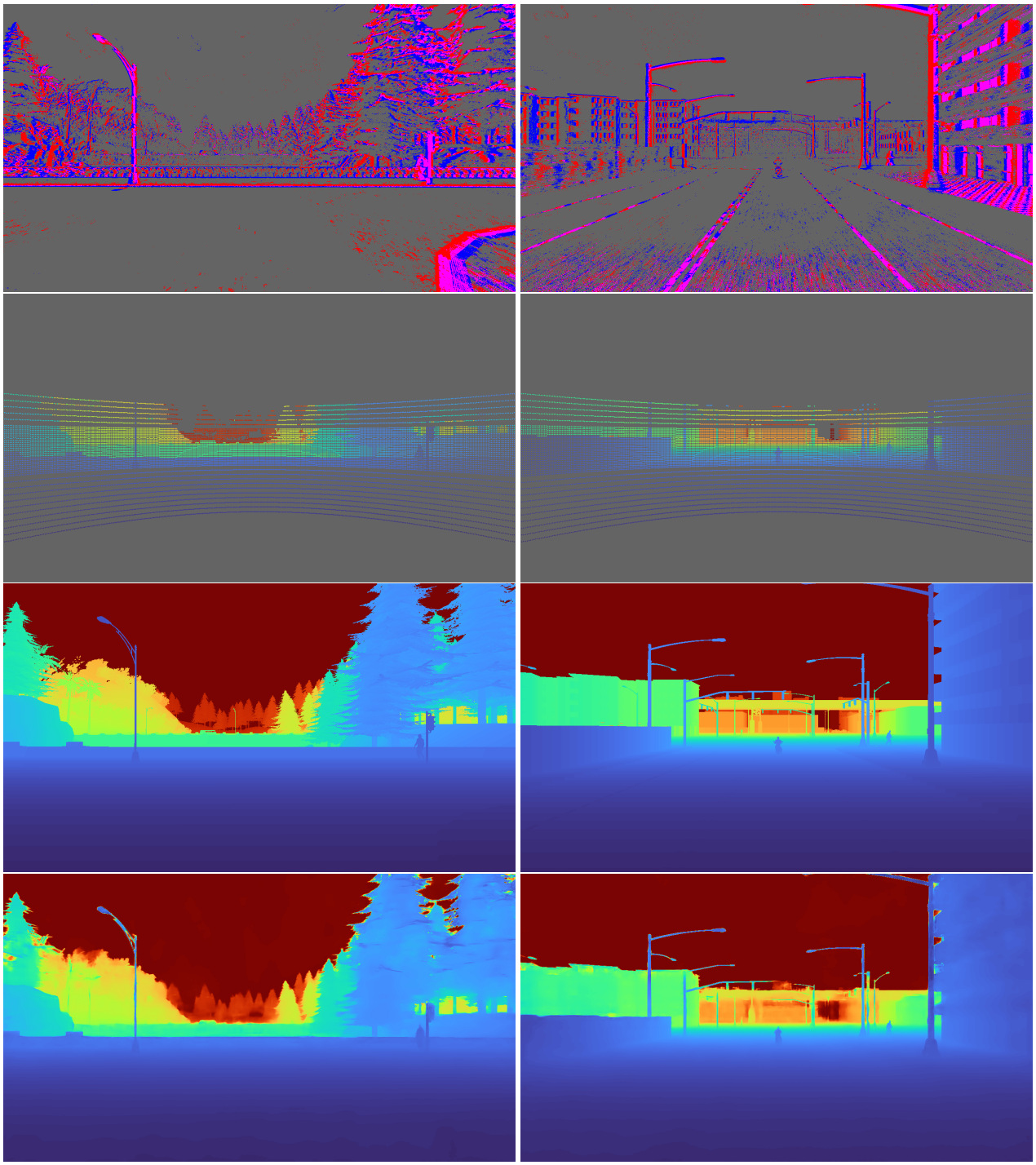

Figure 18. Additional results on the SLED dataset, on sequences `Town01_18` and `Town03_02`. From top to bottom: events, LiDAR projection, ground truth, our results.

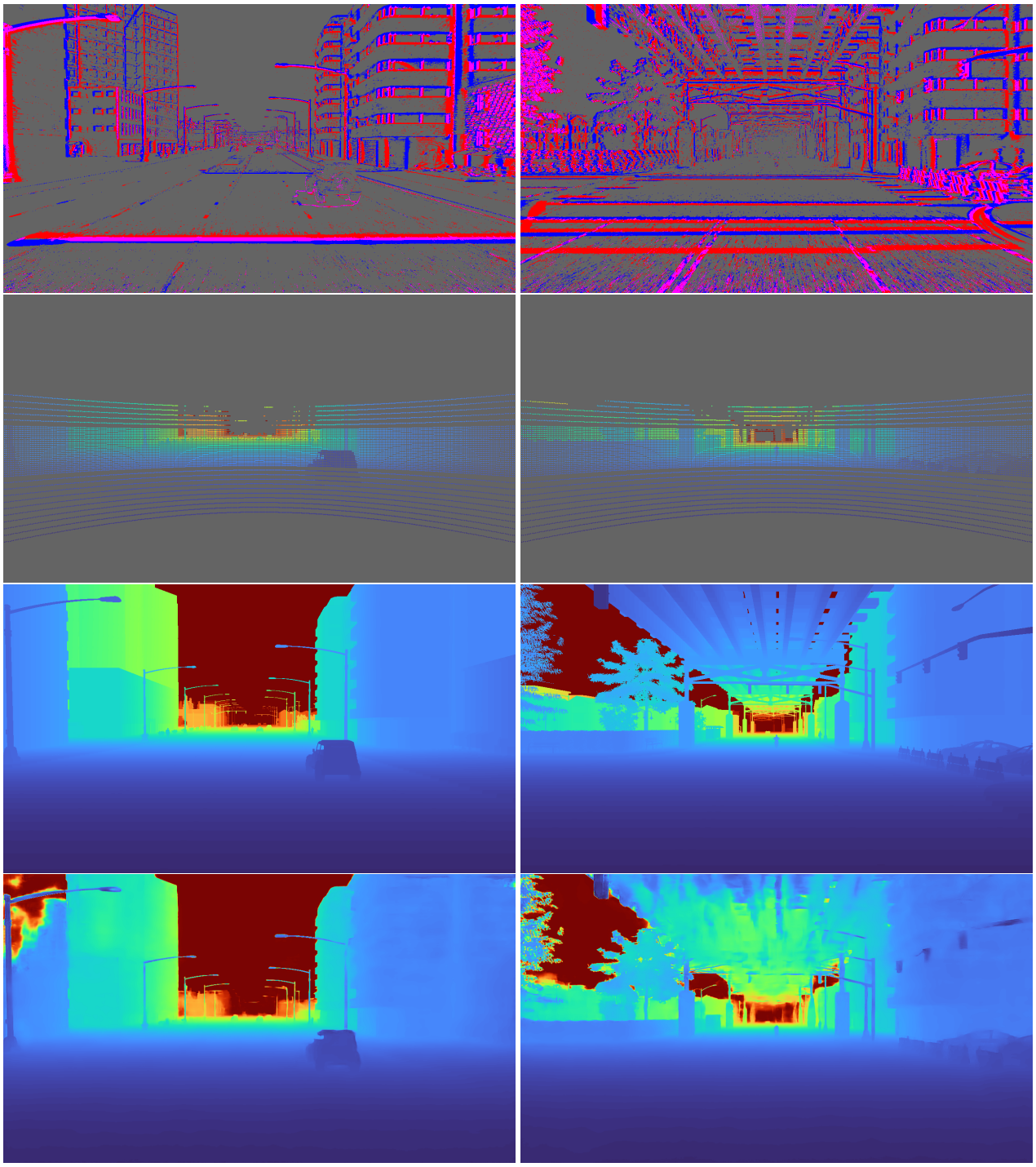

Figure 19. Additional results on the SLED dataset, on sequences `Town03_06` and `Town03_13`. From top to bottom: events, LiDAR projection, ground truth, our results. Shown here are two cases where DELTA_{SL} displays moderate to large errors for objects in the upper part of the depth maps (where no LiDAR data is available), like the building on the top left for the left column, and the suspended railway on the top for the right column.

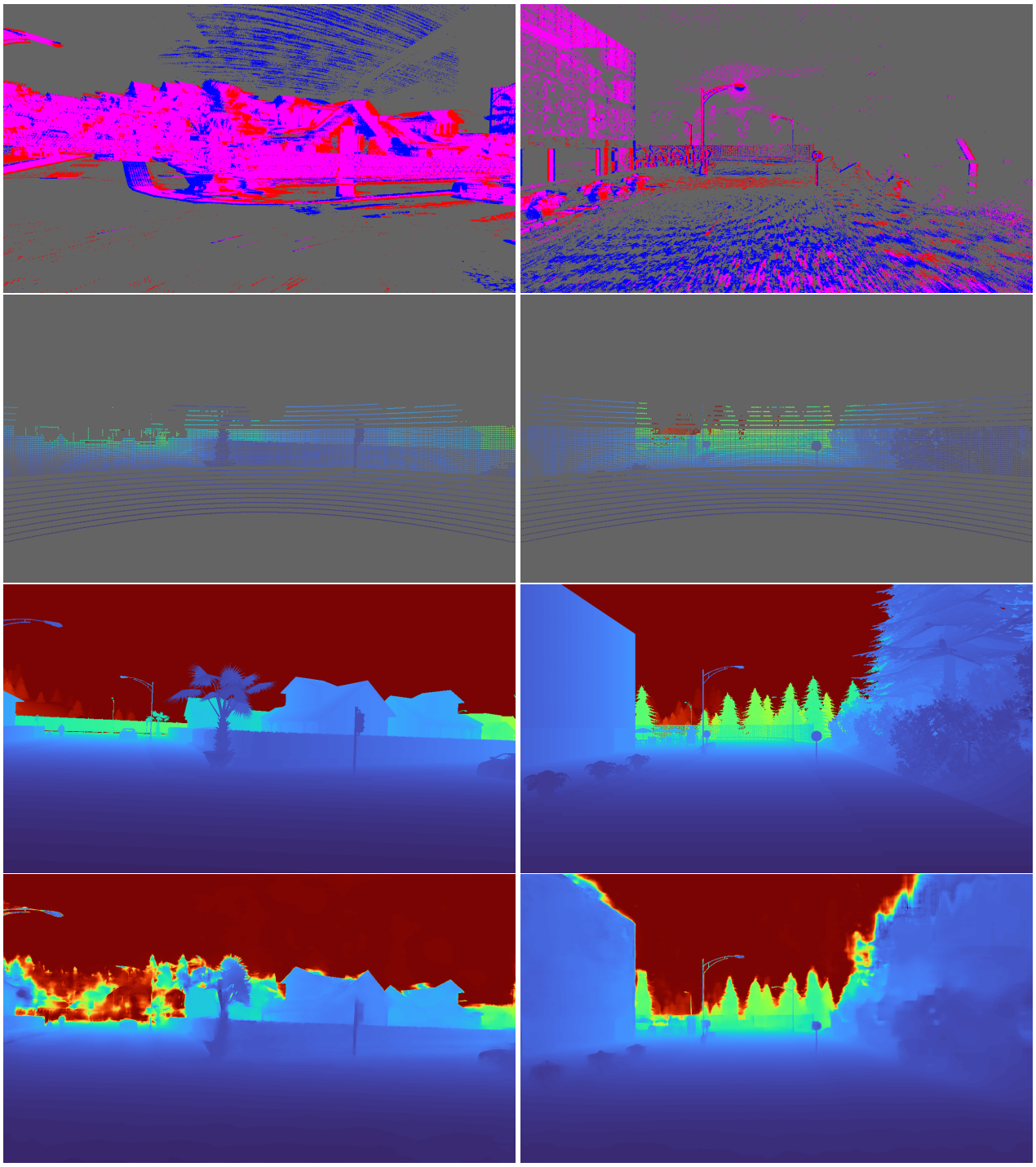

Figure 20. Additional results on the SLED dataset, on sequences `Town01_08` and `Town01_11`. From top to bottom: events, LiDAR projection, ground truth, our results. Shown here are two failure cases where DELTA$_{SL}$ displays large errors. Left: due to a high-speed sharp turn, a very high quantity of events is produced in the time window of accumulation, leading to information being lost in the event volume, and thus leading to an inaccurate depth estimation for background objects. Right: due to the limitations of the event camera in the CARLA simulator, dark objects in a night scene like the trees in the middle and on the right of the scene are not captured in the event stream of the SLED dataset, resulting in blurry depth estimations for these objects.

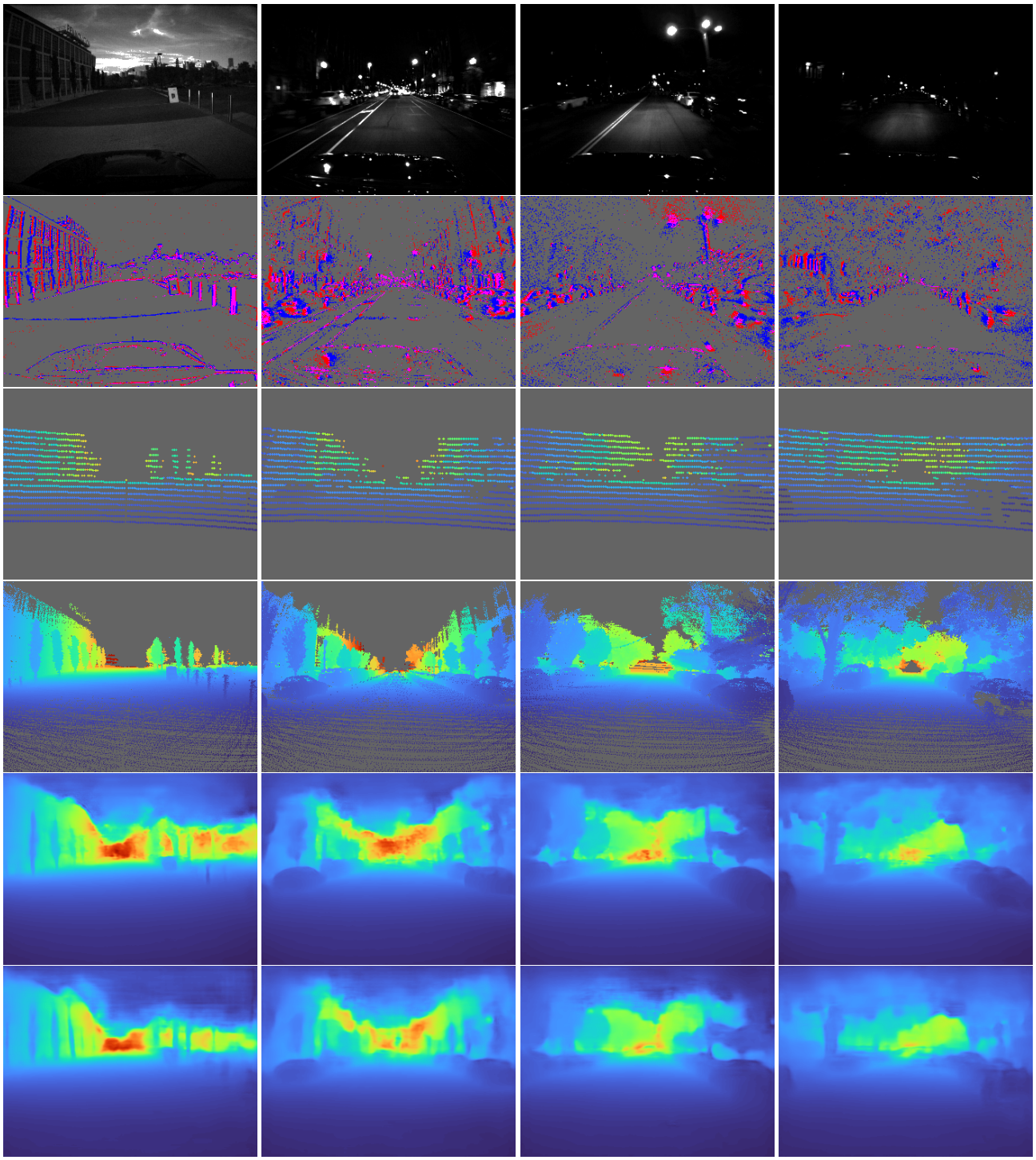

Figure 21. Additional results on the MVSEC dataset. Sequences shown, from left to right: `outdoor_day_1`; `outdoor_night_1`; `outdoor_night_2`; `outdoor_night_3`. From top to bottom: reference image of the scene; events; LiDAR projection (with size of points increased for a better visibility); ground truth; our results (DELTA_{MV}, $\text{DELTA}_{\text{SL}\rightarrow\text{MV}}$).

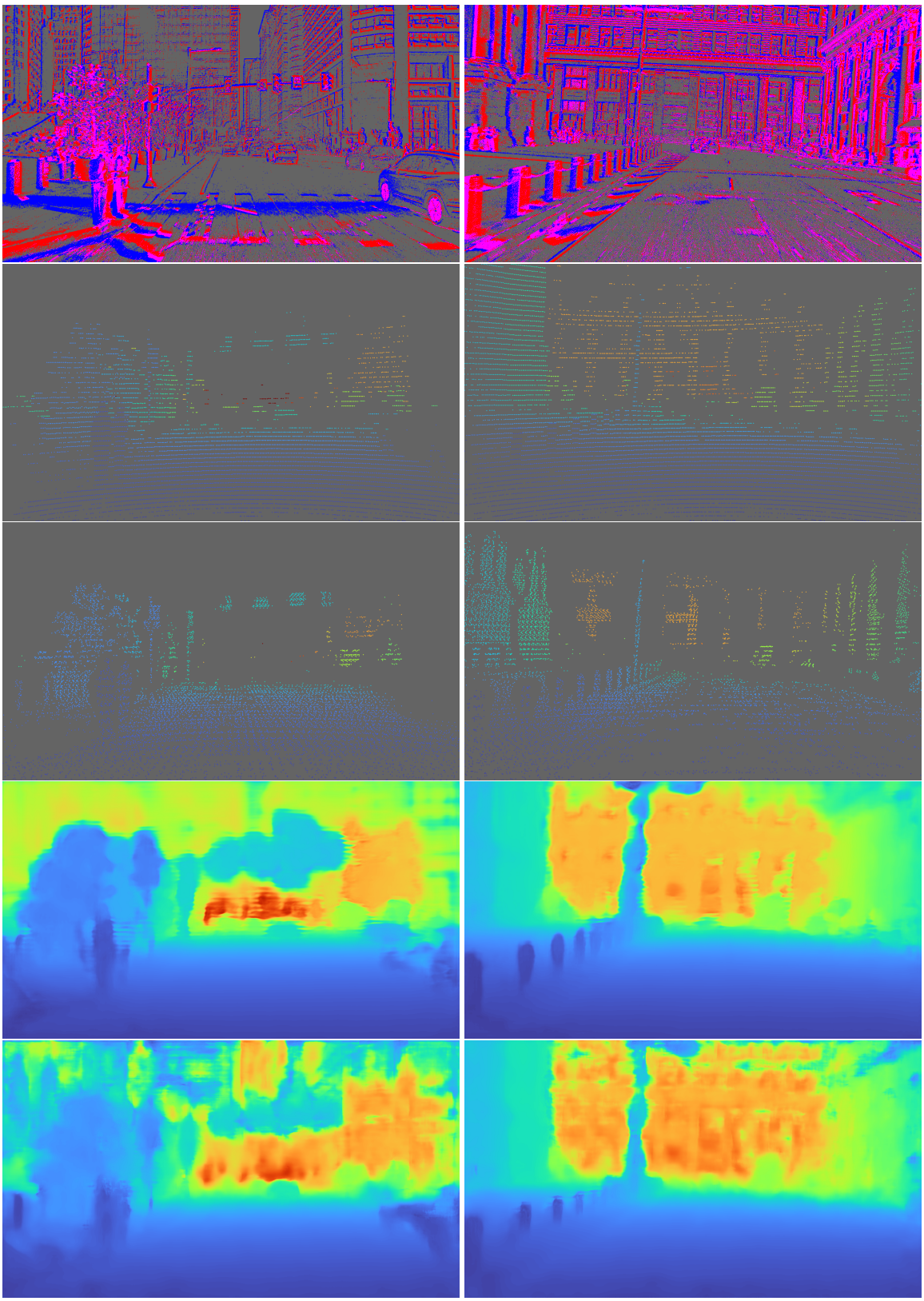

Figure 22. Additional results on the M3ED dataset, for the `city_hall_day` sequence. From top to bottom: events, LiDAR projection, ground truth, our results (DELTA_{M3}, $\text{DELTA}_{\text{SL}\rightarrow\text{M3}}$).

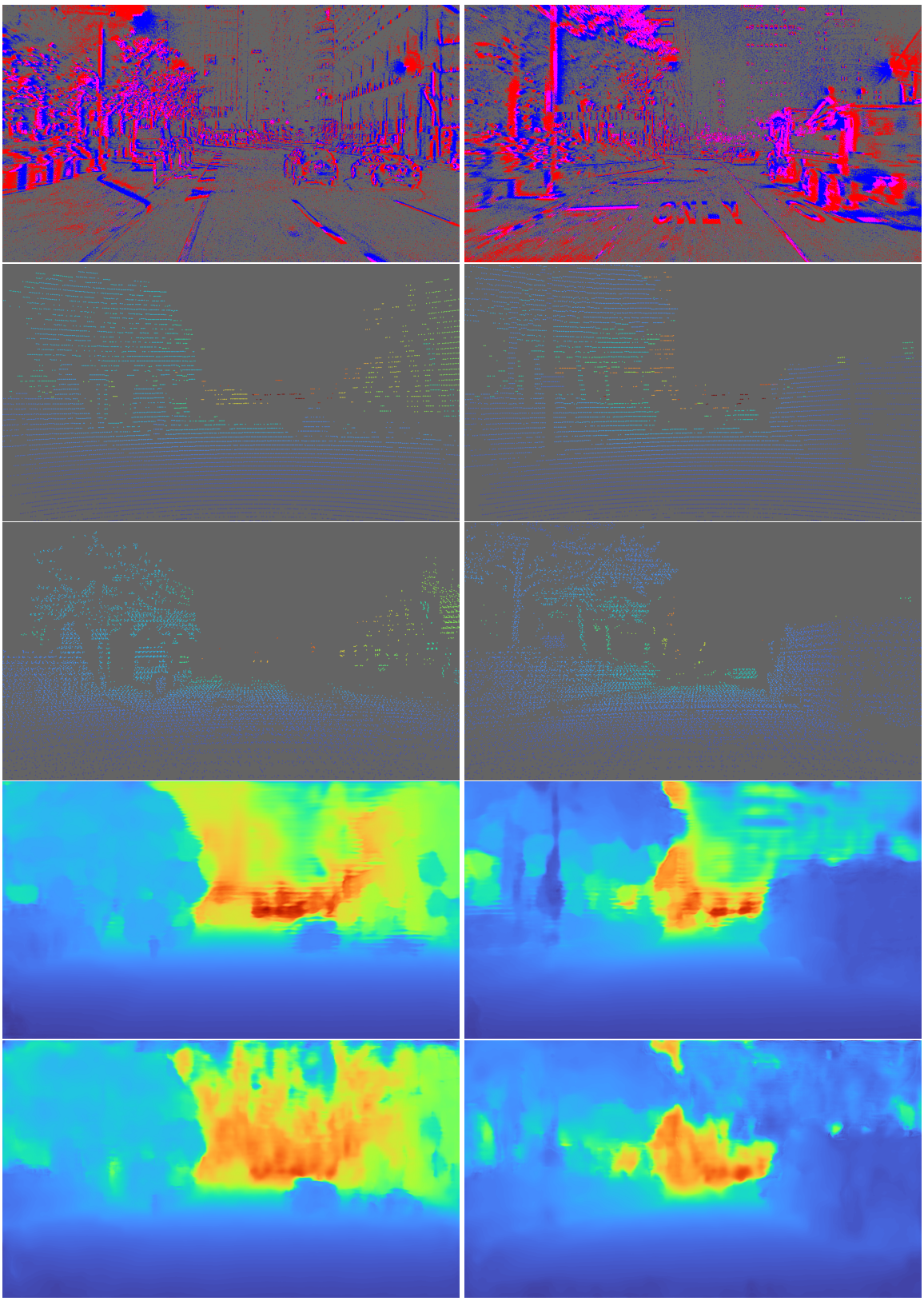

Figure 23. Additional results on the M3ED dataset, for the `city_hall_night` sequence. From top to bottom: events, LiDAR projection, ground truth, our results (DELTA_{M3}, $\text{DELTA}_{\text{SL}\rightarrow\text{M3}}$).