Flow-Guided Online Stereo Rectification for Wide Baseline Stereo (Supplementary Information)

Anush Kumar[1] Fahim Mannan[1] Omid Hosseini Jafari[1] Shile Li[1] Felix Heide[1, 2]
[1]Torc Robotics [2]Princeton University

In this supplemental document, we present additional detail and experiments in support of the findings in the main manuscript. Specifically, we provide details on the differentiable rectification module, stereo network architecture, and we discuss further evaluations of the rotation error, depth quality, and provide additional qualitative results. Finally, we also provide further qualitative examples of the datasets we use for training and evaluation of the proposed method. We also release the proposed datasets at https://light.princeton.edu/online-stereo-recification/.

Contents

1. Differentiable Rectification

An essential component in our model is the differentiable rectification module *DRectify*, which we describe in the following. Being differentiable, this module enables us to train end-to-end while inferring rectified images from the model's pose predictions, which in-turn enables us to employ rectification constraints to the model during training.

To define this operator, we assume two images $I_1, I_2 \in \mathbb{R}^{H\times W\times 3}$ and the relative pose between the sensors as $P = [R|t] \in SE(3)$ and intrinsics $K_1, K_2 \in \mathbb{R}^{3\times 3}$, and we aim to project I_1, I_2 onto a common image plane using rectification rotations R_1, R_2 resulting in images $I_{1_{rect}}, I_{2_{rect}}$. We use differentiable implementations from PyTorch3D and Kornia [5, 10] for conversion between rotation representations such as *matrix2euler, euler2matrix, rodrigues2matrix*

Estimating Rectification Homographies. We first extract our relative rotation and translation estimate R, t from P and transform them to half-rotations as defined below,

$$R_{euler} = matrix2euler(R) \times -0.5, \tag{1}$$

$$R_{half} = euler2matrix(R_{euler}) \tag{2}$$

We then half-rotate our translation vector t to obtain,

$$t_{half} = R_{half}t \tag{3}$$

Here, half-rotations are preferred as we can apply equal and opposite rotations to each image resulting in minimal Field-of-View (FoV) loss as compared to applying a full rotation to only a single image from the stereo pair.

We previously assumed a horizontal baseline, and now define an arbitrary unit vector $u \in \mathbb{R}^{3\times1}$ along the x-axis of t_{half}. Assuming u is our ideal rectified baseline vector, we aim to estimate a rotation that aligns t_{half} in the direction of u. To do so we estimate a vector

$$w = t_{half} \times u \tag{4}$$

normal to the plane containing t_{half}, u where w represents a direction vector implicitly capturing the rotation required to align t_{half} to u. Next, we convert to matrix representation $w_R = rodrigues2matrix(w)$ and compose with R_{half} to obtain the final rectification rotation homographies.

$$R_1 = w_R R_{half}^T, R_2 = w_R R_{half}, t_{rect} = w_R t_{half} \tag{5}$$

This is followed by constructing the rectification projection matrices $P_{1_{rect}}, P_{2_{rect}} \in \mathbb{R}^{3\times4}$ and $Q \in \mathbb{R}^{4\times4}$ which is used to map disparity to depth.

Rectifying the Images. We now we have the rectification rotations R_1, R_2, the new projection matrices $P_{1_{rect}}, P_{2_{rect}}$ and our original intrinsics K_1, K_2 in hand. To simplify notation, we describe this process for a single pixel but this is vectorized in *DRectify* for efficiency and reused for both images. We convert a 2D pixel $p_{old} = (u, v, 1)$ in homogenous coordinates to the normalized camera coordinates using P_{rect}.

$$p_{norm} = (P_{rect})^{-1} p_{old} \tag{6}$$

Next we apply the homography R_{rect} as follows,

$$p_{rect} = R_{rect}^T p_{norm} \tag{7}$$

We then project p_{rect} in camera coordinates to pixels using K

$$p_{new} = K p_{rect} \tag{8}$$

Given the location of our new pixel coordinates, we apply the differentiable *remap()* function from Kornia [5, 8] to assign the pixel value at p_{old} to p_{new}.

2. Network Architecture Details

In this section, we provide additional detail on the network architecture of our method. Table 1a lists the architecture of our feature extractor and feature enhancer. We list our decoder architecture in Table 1b containing the layers and steps involved to produce a valid rotation prediction from our model. We use this rotation estimate to rectify our images using the differentiable rectification module described in Section 1. The differentiable rectification is also mentioned in Table 1a, this step corresponds to rectifying the features to an initial rotation estimates as described in the main paper.

3. Network Architecture Ablations

Table 2 evaluates the proposed model in the absence of key components from the Network Architecture (Figure 2 of the main paper). The experiments validate that the *Feature enhancement* (Figure 2(b) of main paper) play a significant role in the performance, as does *Feature Rectification* (rectifying features to identity rotation matrix as an initial step). We also see a drop in metrics in the absence of the *Cost Volume*, which replaces the cost volume and the decoder (Figure 2(c) of main paper) with aggregation (concatenation) of image features followed by a series of 2D Convolutions and Pooling operations before estimating rotation. We also observe poor performance when trained on lower resolution images (*256x256* for main model and *128x128* for flow estimation), while gaining 60% speedup in inference time. We include *w/o Optical Flow* and the Proposed Model metrics from Table 2(c) in the main paper, for reference.

Layer #	Layer Description		Output Shape
CNN FEATURE EXTRACTOR			
1	Residual Block-1	Conv2d InstanceNorm Skip Connection $\oplus$ ReLU	$64 \times 256 \times 512$
2	Residual Block-2	Conv2d InstanceNorm Skip Connection $\oplus$ ReLU	$96 \times 128 \times 256$
3	Residual Block-3	Conv2d InstanceNorm Skip Connection $\oplus$ ReLU	$128 \times 64 \times 128$
-	*Differentiable Rectification*		$128 \times 64 \times 256$
TRANSFORMER FEATURE ENHANCER			
1	Positional Encoding		$128 \times 64 \times 256$
2	Transformer Block-1	Feed forward Network Self-Attention Block Cross-Attention Block Feed forward Network	128×8192
3	Transformer Block-2	Feed forward Network Self-Attention Block Cross-Attention Block Feed forward Network	128×8192
4	Transformer Block-3	Feed forward Network Self-Attention Block Cross-Attention Block Feed forward Network	128×8192
5	Transformer Block-4	Feed forward Network Self-Attention Block Cross-Attention Block Feed forward Network	128×8192
6	Transformer Block-5	Feed forward Network Self-Attention Block Cross-Attention Block Feed forward Network	128×8192
7	Transformer Block-6	Feed forward Network Self-Attention Block Cross-Attention Block Feed forward Network	128×8192

(a)

Layer #	Layer Description		Output Shape
DECODER			
-	*Correlation Volume*		$64 \times 128 \times 64 \times 128$
1	Decoder Block-1	Conv3d BatchNorm3D	$32 \times 64 \times 64 \times 128$
2	Decoder Block-2	Conv3d BatchNorm3D	$16 \times 64 \times 64 \times 128$
3	Decoder Block-3	Conv3d BatchNorm3D	$8 \times 64 \times 64 \times 128$
4	Decoder Block-4	Conv3d BatchNorm3D	$4 \times 64 \times 64 \times 128$
5	Decoder Block-5	Conv3d BatchNorm3D	$2 \times 64 \times 64 \times 128$
6	Decoder Block-6	Conv3d BatchNorm3D ReLU	$1 \times 64 \times 64 \times 128$
-	Max		$1 \times 64 \times 128$
-	Flatten		1×8192
7	Linear		1×6
	Tanh		
-	*Gram-Schmidt*		3×3

(b)

Table 1. (a) Architecture of the feature extractor and feature enhancement blocks. Here *Differentiable Rectification* corresponds to the module described in Section 1. (b) Architecture for decoder module after the correlation volume is computed using the enhanced features from Table 1a.

Ablation	**MAE**	**SIFT Offset** *(pixels)*	**SuperGlue Offset** *(pixels)*	**Vertical Flow** *(pixels)*	**Inference Time** *(msec)*
w/o Feature Enhancement	0.15	8.47	6.88	6.82	41
w/o Feature Rectification	0.14	8.32	6.88	6.75	79
w/o Optical Flow	0.13	7.68	5.53	5.67	86
w/o Cost Volume	0.06	3.92	0.98	1.33	44
Proposed Model	**0.05**	**3.55**	**0.64**	**1.003**	86
w/ Reduced Image Resolution	0.08	3.98	1.07	1.39	**36**

Table 2. We provide additional ablation experiments on the Carla dataset. The experimennts validate the need for key components in the network architecture.

4. Additional Rotation Errors Evaluations

Next, we provide additional evaluations that evaluate the rotation error as a metric typically evaluated to assess camera pose estimation approaches. Table 3 reports the errors in rotation measured in degrees along x (R_X), y (R_y) and z (R_z) axes of rotation. These errors are computed as the mean absolute error between the ground truth relative rotation and the predicted relative rotation, after conversion to axis angle representation. On the real datasets, Semi-Truck Highway and KITTI [7], our method consistently outperforms other methods. We attribute this to our method being directly trained to reduce the rotation error. We report these metrics here since it is standard practice in general camera pose estimation approaches. On the Carla dataset, featuring extreme pose variations, still performs best overall while other pose estimation methods are in a competitive range.

5. Additional Stereo Depth Evaluations

In this section, we provide further evaluation of the effect of different rectification methods on downstream stereo depth estimation, in the presence of pose variation and de-calibration. We used two publicly available off-the-shelf stereo models (HITNet [14] and DLNR [15]) for this evaluation. Given left and right input images (I_L and I_R), we first retrieve the depth map using ground-truth calibration in the following steps. We rectify I_L and I_R using ground-truth calibration parameters and get I_L^{*Rect} and I_R^{*Rect}. Then, we pass I_L^{*Rect} and I_R^{*Rect} to the stereo model and compute the depth map Y^{*Rect}. We finally unrectify the depth map, yielding the reference depth Y^* with ground truth calibration. For a given calibration method M, we then estimate depth by rectifying I_L and I_R using calibration parameters of M and get I_L^{MRect} and I_R^{MRect}. We pass I_L^{MRect} and I_R^{MRect} to the identical stereo model from above and compute the depth map Y^{MRect}, which we also unrectify to yield the estimated depth map Y^M for method M.

Given the predicted depth value of a pixel y_i from Y^M and the depth value of a pixel y_i^* on Y^* with ground truth calibration, we evaluate the following metrics:

- Mean Absolute Error (MAE):

$$\frac{1}{N}\sum_i |y_i^* - y_i| \tag{9}$$

- Root Mean Squared Error (RMSE):

$$\sqrt{\frac{1}{N}\sum_i (y_i^* - y_i)^2} \tag{10}$$

- Scale Invariant Logarithmic error (SILog):

$$\frac{1}{N}\sum_i d_i^2 - \frac{1}{N^2}\left(\sum_i d_i\right)^2, \tag{11}$$

  where $d_i = \log y_i - \log y_i^*$.
- Absolute Relative Error percent (AbsRel):

$$\frac{1}{N}\left|\frac{y_i^* - y_i}{y_i^*}\right| * 100 \tag{12}$$

- Accuracy with threshold thr: percentage (%) of y_i s.t.

$$\max\left(\frac{y_i^*}{y_i}, \frac{y_i}{y_i^*}\right) = \delta < thr, \tag{13}$$

  where $thr \in \{1.25, 1.25^2, 1.25^3\}$.

We evaluate the metrics in binned $[0, 80]m$, $[0-100]m$, $[0-200]m$, and $[0-300]m$ depth ranges. The proposed method consistently outperforms existing methods on the Semi-Truck Highway and KITTI [7] (Tables 5 and 6) and compares favorably on the CARLA dataset (Table 7).

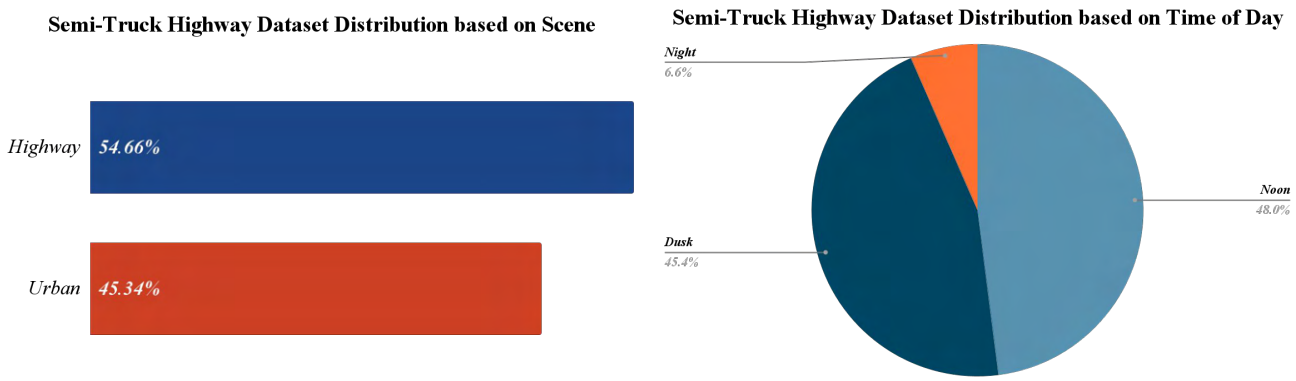

Figure 1. **Semi-Truck Highway Data Distribution.** (Left) Summarises the distribution of data samples in *Highway Scenes* v/s *Urban Scenes*. (Right) Summarises the distribution of data based on the time of day, to capture the variation in natural light throughout the dataset.

6. Additional Dataset Details

We include additional details and examples from our two proposed datasets, the **Semi-Truck Highway** dataset in Fig. 2, and our custom **CARLA** dataset in Fig. 3. The examples highlight the diversity in scenes, lighting and environment that were described in the main manuscript.

The **Semi-Truck Highway** captures wide-baseline calibration deterioration naturally in a long-haul trucking operation setting. The dataset was captured with four front-facing 8MP 20-bit RCCB HDR sensors (AR0820) with 30 degrees horizontal field of view lenses. The cameras were mounted on a single rigid bar placed on a truck at a height of approximately 3m from the ground, and the cameras were distributed over a 2m baseline with the baseline varying between 0.6m and 0.7m. The mounting plate for the cameras were custom-made to ensure that they are attached rigidly and there are no significant orientation differences between each pair. Offline calibration was performed before and after each capture drive to ensure consistency in the offline calibration parameters. Calibration was performed in two stages: lab-based offline intrinsic parameter estimation and on-site calibration using charts with clearly detectable patterns. Calibration captures were done while the vehicle was static, either in neutral or the engine turned off, to reduce any artifacts due to camera vibration and rolling-shutter effects. A total of 52 hours of data were collected in urban areas and highways under varying illumination conditions. Fig. 1 shows the scene and illumination distribution. Here, we include the distribution of data based on the scene (Highway, Urban) and the time of day which we split into (noon, dusk and night). We provide additional metrics on the Semi-Truck Highway dataset based on the time of day evaluations in Table 4. We provide a breakdown of the metrics based on time of day on our proposed method and our baseline methods, as seen in from the table our method performs well in challenging low light scenarios (Dusk and Night) with comparable metrics to daytime scenes (Noon).

Our custom **CARLA** dataset was created to maximize the diversity of poses. While our real data were captured across multiple capture campaigns to maximize the scene and illumination diversity, pose diversity is still hard to achieve in practice. To this end, we used CARLA to synthetically generate different pose variations while having a setup close to the Truck setup. For the synthetic dataset we used the same 8MP resolution cameras with a 30 degrees field of view but in RGB format. The cameras are placed at a height of 2.5m from the ground and a baseline of 0.8m between each pair. For the scenes, we used Town01 to Town06 with random waypoints. For each waypoint, the cameras poses are randomly perturbed before rendering the scene.

7. Additional Qualitative Images

To provide further qualitative insight into the effectiveness of our proposed method, we add additional qualitative images from all three datasets we evaluate on. We overlay the left-right rectified stereo pairs and include the corresponding stereo depth inferred by HITNet [14] on these images. For KITTI [7], we see our model compares favorably as reported in Fig. 4. Our evaluation on the semi-truck highway datasets highlights the effects of wide baseline calibration deterioration, see Fig. 5. The results on CARLA dataset show our model ability to handle large pose variations while also displaying the challenge this dataset proposes when comparing the results from the other methods, see Fig. 6. The proposed method handles significant

Dataset	Method	R_x *(degrees)*	R_y *(degrees)*	R_z *(degrees)*
(a) **Semi-Truck Highway**	SIFT + LO-RANSAC [3, 9]	6.64	5.37	10.15
	SuperGlue + MAGSAC [1, 12]	0.79	0.46	0.18
	LOFTR + MAGSAC [1, 13]	0.69	3.05	0.50
	RPNet [6]	0.62	1.16	0.49
	DirectionNet [2]	0.37	3.6	0.47
	ViTPose [11]	0.15	0.16	0.11
	Ours (*w/o OF*)	0.03	0.05	0.04
	Ours (*w/ OF*)	**0.02**	**0.02**	**0.01**
(b) **KITTI**[7]	SIFT + LO-RANSAC [3, 9]	2.18	1.51	2.66
	SuperGlue + MAGSAC [1, 12]	0.19	0.11	0.03
	LOFTR + MAGSAC [1, 13]	0.10	0.10	0.02
	RPNet [6]	0.08	0.28	0.03
	DirectionNet [2]	0.08	0.19	0.12
	ViTPose [11]	0.05	0.09	0.06
	Ours (*w/o OF*)	**0.02**	0.05	**0.004**
	Ours (*w/ OF*)	0.03	**0.003**	0.008
(c) **CARLA**	SIFT + LO-RANSAC [3, 9]	3.32	1.80	2.85
	SuperGlue + MAGSAC [1, 12]	0.29	**0.11**	0.03
	LOFTR + MAGSAC [1, 13]	0.45	0.14	0.14
	RPNet [6]	0.55	0.66	0.66
	DirectionNet [2]	1.10	0.80	0.76
	ViTPose [11]	0.12	0.16	0.29
	Ours (*w/o OF*)	0.55	0.53	0.52
	Ours (*w/ OF*)	**0.03**	0.23	**0.11**

Table 3. Quantitative Evaluation on the (a) Semi-Truck Highway, (b) KITTI [7] and (c) Carla Datasets. Included in this table are the Rotation errors along all three axes of rotation. We compute the MAE between the predicted rotation estimates and the ground truth rotation. R_x, R_y and R_z correspond to rotations about x, y, and z-axis respectively.

pose variations in a diverse set of scenarios and lighting conditions, as also validated by the quantitative evaluations in Section 4 and Section 5.

8. Acknowledgments

We would like to thank AISEE and National Research Council of Canada for supporting this project and Torc Robotics for providing us with data and resources.

Dataset	Method	MAE	SIFT Offset *(pixels)*	SuperGlue Offset *(pixels)*	Vertical Flow *(pixels)*
	Unrectified	0.26	8.00	2.56	4.52
	GT (Offline Calibration)	-	2.93	0.79	0.69
	SIFT + LO-RANSAC [3, 9]	0.32	25.68	26.50	17.74
	SuperGlue + MAGSAC [1, 12]	0.16	13.73	11.01	11.35
(a) **Noon**	LOFTR + MAGSAC [1, 13]	0.17	14.12	11.34	11.49
	RPNet [6]	0.20	11.15	8.39	8.66
	DirectionNet [2]	0.33	7.68	5.36	5.76
	ViTPose [11]	0.07	4.39	2.29	2.80
	Ours	**0.018**	**2.70**	**0.59**	**1.09**
	Unrectified	0.21	7.64	2.23	4.89
	GT (Offline Calibration)	-	3.17	0.76	0.71
	SIFT + LO-RANSAC [3, 9]	0.30	28.68	34.56	18.32
	SuperGlue + MAGSAC [1, 12]	0.13	16.31	12.82	12.94
(b) **Dusk**	LOFTR + MAGSAC [1, 13]	0.13	16.6	13.2	12.95
	RPNet [6]	0.17	12.88	9.52	9.79
	DirectionNet [2]	0.25	8.78	6.13	6.53
	ViTPose [11]	0.05	4.47	2.07	2.63
	Ours	**0.013**	**2.88**	**0.56**	**1.13**
	Unrectified	0.15	14.11	9.10	9.83
	GT (Offline Calibration)	-	2.70	0.71	0.41
	SIFT + LO-RANSAC [3, 9]	0.16	22.17	20.76	17.39
	SuperGlue + MAGSAC [1, 12]	0.12	17.47	13.97	14.22
(c) **Night**	LOFTR + MAGSAC [1, 13]	0.11	19.42	15.74	16.32
	RPNet [6]	0.12	15.89	12.61	12.59
	DirectionNet [2]	0.16	11.14	9.20	9.82
	ViTPose [11]	0.07	5.71	4.0	4.03
	Ours	**0.010**	**2.64**	**0.68**	**0.77**

Table 4. Quantitative Evaluation on the (a) Noon, (b) Dusk and (c) Night scenes from the Semi-Truck Highway Dataset. Evaluation on unrectified images and ground truth images are also included.

Figure 2. **Semi-Truck Highway Samples** These are scenes picked at random from our proposed real dataset. As seen the scenes are predominantly highway and urban coupled with different lighting conditions based on the time of day.

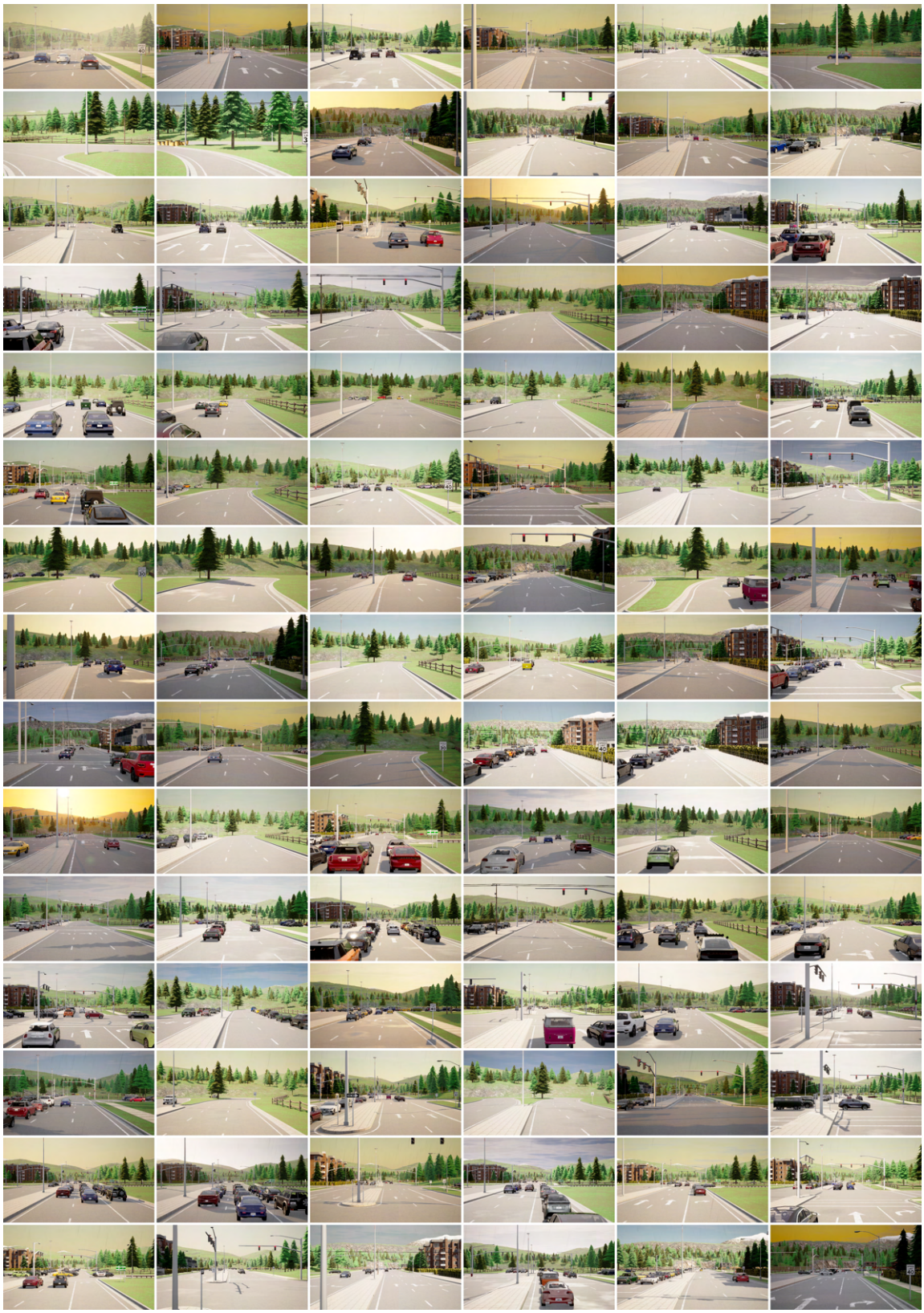

Figure 3. **Carla Dataset Samples** These are scenes picked at random from our proposed CARLA[4] simulated dataset. As seen the scenes are more urban while the lighting variations are more significant. Additionally we also have similar scenes with different lighting conditions in this dataset.

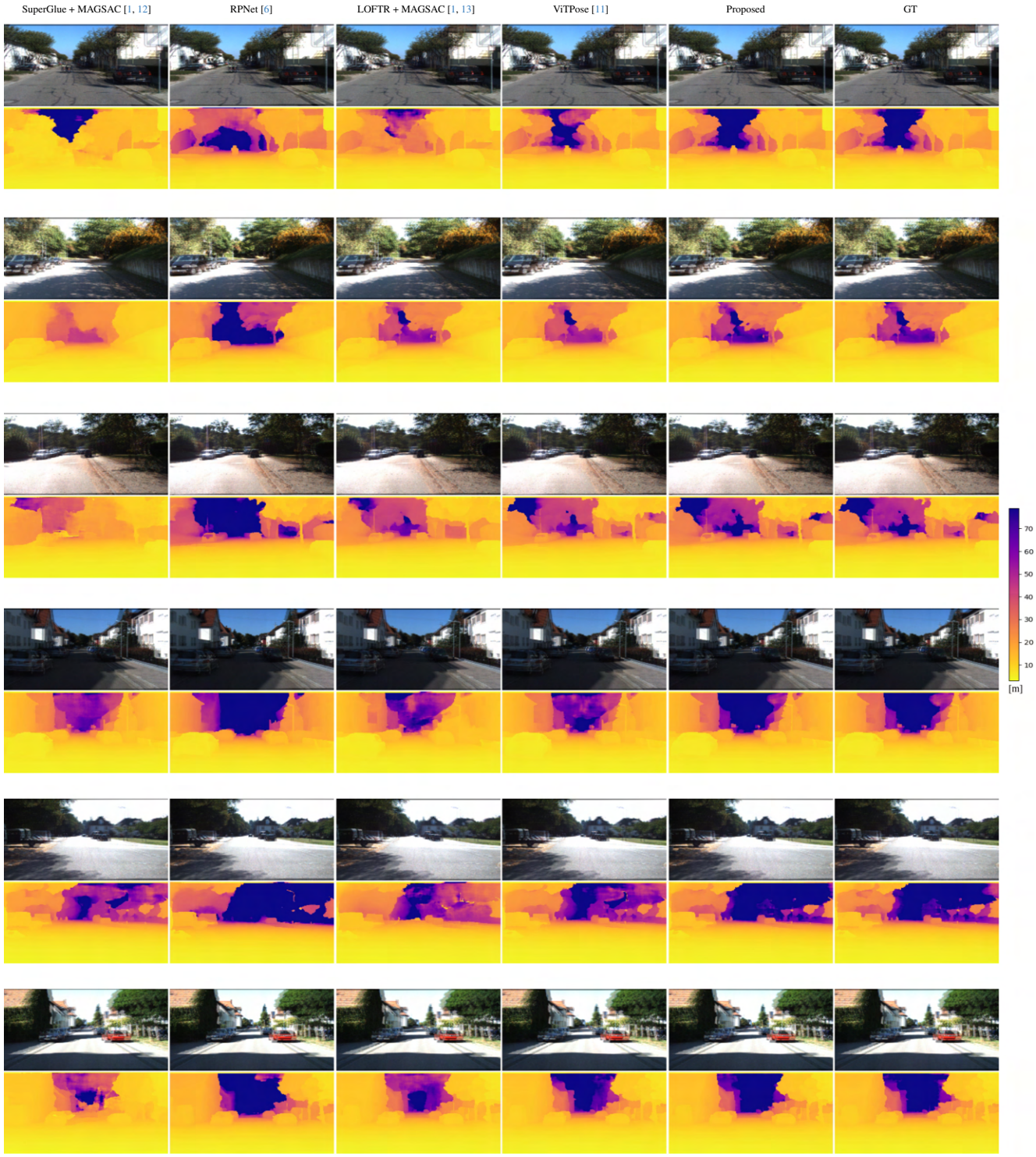

Figure 4. **KITTI [7] Qualitative Assessment**. We overlay here the rectified left-right stereo pairs. Each column represents a different rectification method. To visually evaluate the rectification quality, focus on an object in the scene and compare the vertical disparity. Every row is accompanied by the corresponding depth inferred from HITNet [14].

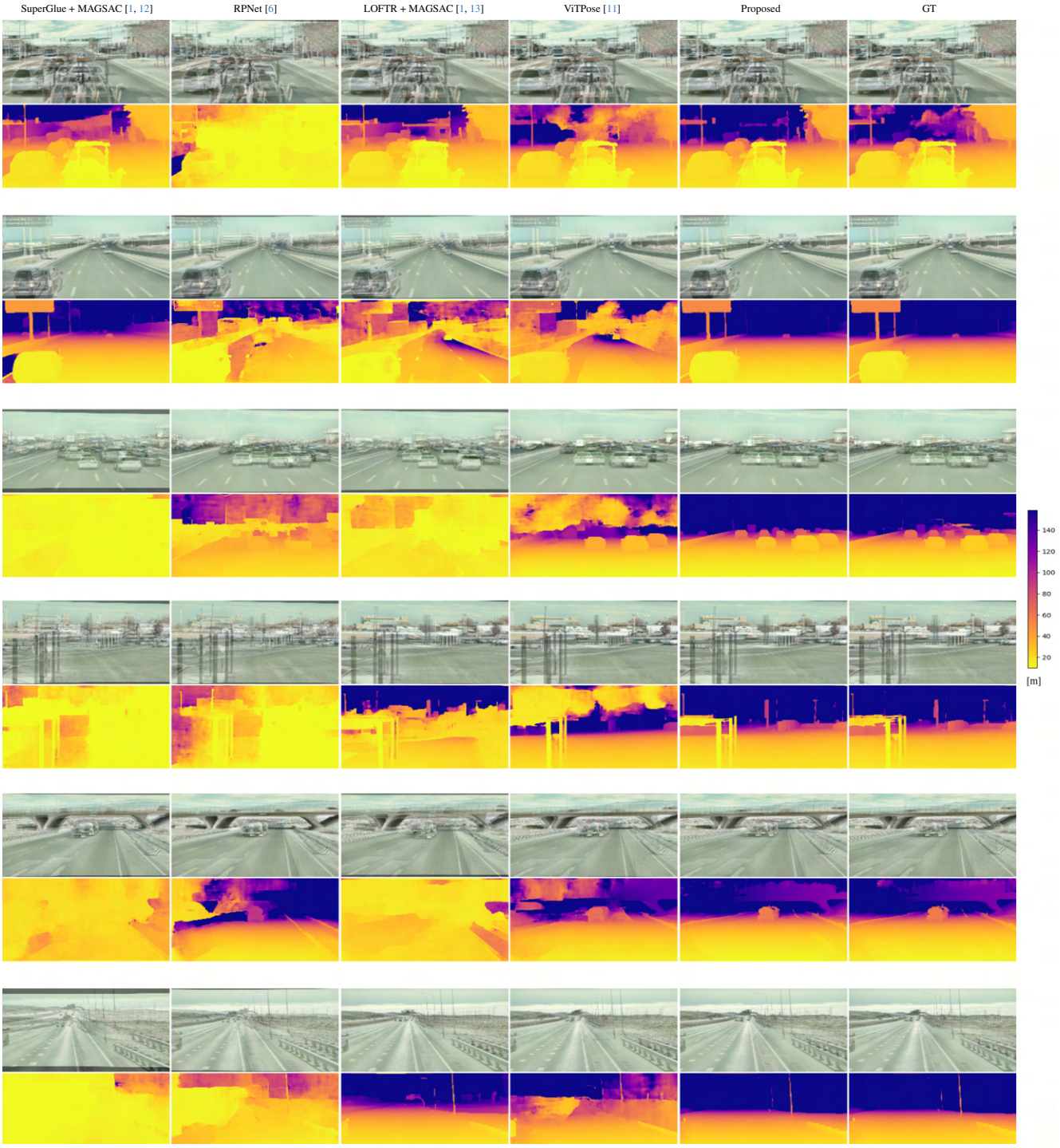

Figure 5. **Semi-Truck Highway Qualitative Assessment**. We overlay here the rectified left-right stereo pairs. Each column represents a different rectification method. To visually evaluate the rectification quality, focus on an object in the scene and compare the vertical disparity. Every row is accompanied by the corresponding depth inferred from HITNet [14].

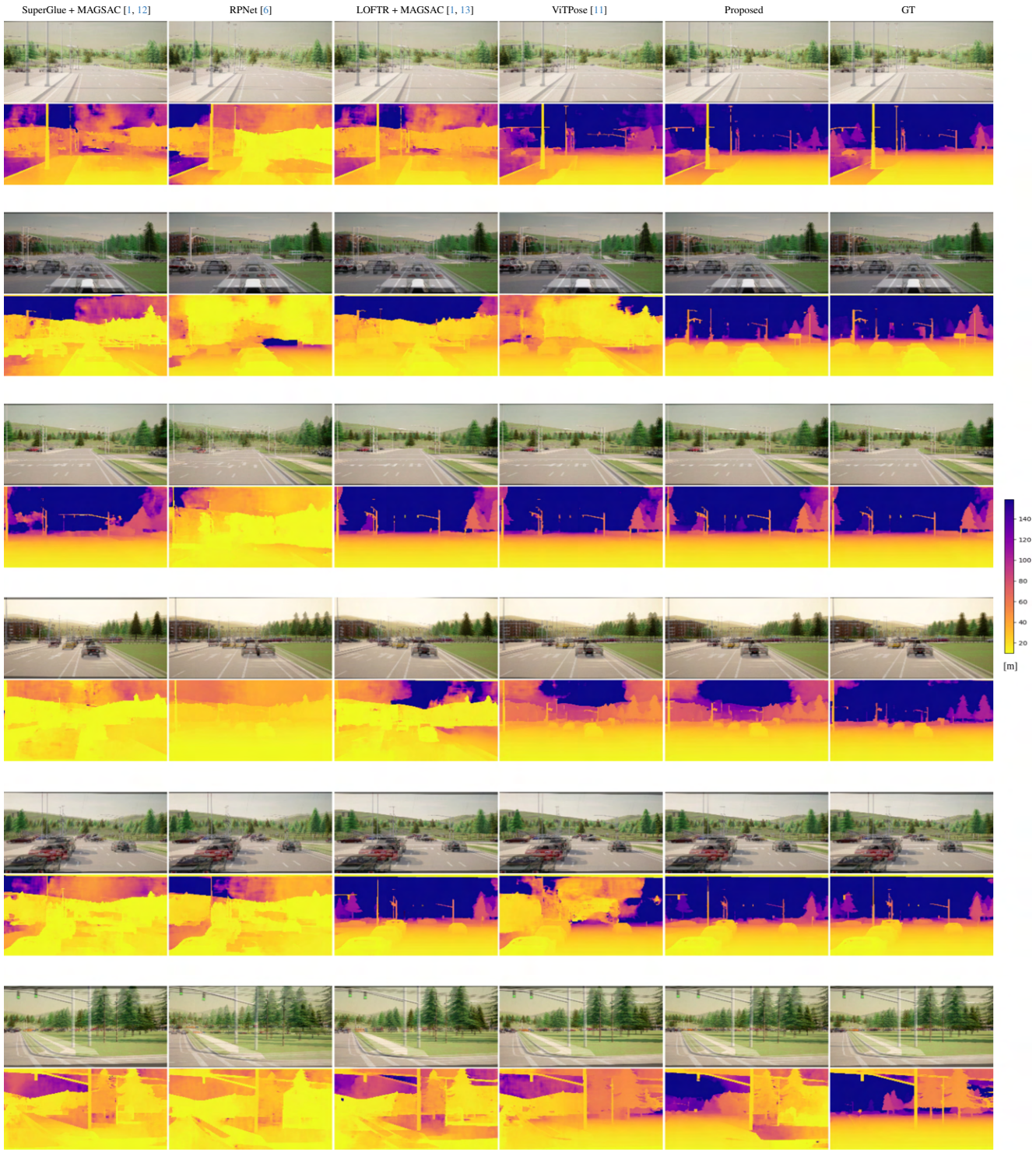

Figure 6. **CARLA Qualitative Assessment**. We overlay here the rectified left-right stereo pairs. Each column represents a different rectification method. To visually evaluate the rectification quality, focus on an object in the scene and compare the vertical disparity. Every row is accompanied by the corresponding depth inferred from HITNet [14].

Stereo Model	Depth Range	Method	lower is better				higher is better		
			MAE	RMSE	SILog	absrel	d<1.25	d<1.25^2	d<1.25^3
DLNR [15]	$[0, 80]m$	DirectionNet [2]	22.81	27.57	42.33	61.98	5.98	13.58	23.69
		LOFTR + MAGSAC [1, 13]	20.70	25.75	49.53	54.87	20.82	31.61	39.99
		RPNet [6]	21.79	26.86	52.79	58.81	11.80	24.02	34.72
		SIFT + LO-RANSAC [3, 9]	23.99	28.74	59.35	64.91	10.56	19.79	27.89
		SuperGlue + MAGSAC [1, 12]	20.97	26.05	50.26	55.70	19.73	30.46	38.95
		ViTPose [11]	10.44	15.31	34.50	26.64	62.93	72.14	77.85
		Proposed	**3.19**	**5.96**	**10.42**	**10.05**	**91.98**	**95.29**	**97.15**
	$[0, 100]m$	DirectionNet [2]	26.52	32.98	46.21	63.47	5.60	12.72	22.17
		LOFTR + MAGSAC [1, 13]	24.31	31.06	53.48	56.70	19.49	29.83	37.94
		RPNet [6]	25.43	32.20	56.40	60.47	10.97	22.37	32.45
		SIFT + LO-RANSAC [3, 9]	27.79	34.16	62.81	66.37	9.86	18.53	26.23
		SuperGlue + MAGSAC [1, 12]	24.61	31.41	54.23	57.57	18.49	28.71	36.87
		ViTPose [11]	13.03	19.62	38.49	28.91	60.07	69.58	75.58
		Proposed	**4.27**	**8.07**	**12.03**	**12.12**	**90.94**	**94.60**	**96.54**
	$[0, 200]m$	DirectionNet [2]	39.91	53.79	57.19	66.86	4.84	10.95	19.09
		LOFTR + MAGSAC [1, 13]	37.33	51.43	64.13	60.84	16.92	26.12	33.50
		RPNet [6]	38.58	52.77	66.49	64.26	9.34	19.02	27.76
		SIFT + LO-RANSAC [3, 9]	41.34	54.74	72.22	69.78	8.45	15.95	22.71
		SuperGlue + MAGSAC [1, 12]	37.72	51.88	64.85	61.73	16.05	25.10	32.47
		ViTPose [11]	23.24	37.34	49.41	34.79	53.42	63.28	69.77
		Proposed	**8.26**	**16.63**	**16.21**	**17.65**	**88.34**	**93.16**	**95.50**
	$[0, 300]m$	DirectionNet [2]	47.28	66.22	61.68	67.91	4.64	10.49	18.28
		LOFTR + MAGSAC [1, 13]	44.57	63.72	68.39	62.13	16.25	25.11	32.27
		RPNet [6]	45.90	65.19	70.68	65.44	8.92	18.16	26.52
		SIFT + LO-RANSAC [3, 9]	48.70	66.93	76.05	70.86	8.07	15.28	21.78
		SuperGlue + MAGSAC [1, 12]	44.99	64.18	69.12	62.97	15.42	24.15	31.27
		ViTPose [11]	29.34	48.72	53.83	36.92	51.41	61.23	67.78
		Proposed	**10.67**	**22.38**	**17.81**	**19.78**	**87.30**	**92.62**	**95.16**
HITNet [14]	$[0, 80]m$	DirectionNet [2]	21.86	26.83	48.91	63.74	6.80	15.24	26.00
		LOFTR + MAGSAC [1, 13]	18.31	23.70	50.62	50.27	25.80	39.03	48.61
		RPNet [6]	19.78	25.36	58.72	55.66	15.69	30.73	42.99
		SIFT + LO-RANSAC [3, 9]	22.79	27.75	66.73	64.86	12.38	23.85	34.53
		SuperGlue + MAGSAC [1, 12]	18.53	23.96	51.55	50.84	24.60	37.86	47.68
		ViTPose [11]	7.54	11.82	27.96	19.87	71.98	82.70	87.76
		Proposed	**1.71**	**3.94**	**7.82**	**5.01**	**96.27**	**98.01**	**98.86**
	$[0, 100]m$	DirectionNet [2]	25.19	32.01	53.50	65.61	6.47	14.48	24.73
		LOFTR + MAGSAC [1, 13]	21.50	28.77	55.55	52.36	24.48	37.28	46.65
		RPNet [6]	23.10	30.58	63.64	57.80	14.80	29.06	40.84
		SIFT + LO-RANSAC [3, 9]	26.31	33.03	71.40	66.88	11.72	22.60	32.86
		SuperGlue + MAGSAC [1, 12]	21.74	29.06	56.50	52.91	23.35	36.15	45.70
		ViTPose [11]	9.50	15.43	31.70	21.65	69.56	80.75	86.10
		Proposed	**2.38**	**5.54**	**9.04**	**6.09**	**95.61**	**97.58**	**98.52**
	$[0, 200]m$	DirectionNet [2]	37.95	54.05	66.91	69.98	5.75	12.87	21.96
		LOFTR + MAGSAC [1, 13]	33.94	50.62	69.59	57.82	21.66	33.32	42.06
		RPNet [6]	35.75	52.68	77.57	63.18	12.95	25.52	36.05
		SIFT + LO-RANSAC [3, 9]	39.51	55.10	84.43	71.55	10.33	20.02	29.26
		SuperGlue + MAGSAC [1, 12]	34.20	50.92	70.52	58.20	20.69	32.28	41.16
		ViTPose [11]	18.05	32.62	43.18	27.00	63.42	75.36	81.44
		Proposed	**6.02**	**14.56**	**13.32**	**10.66**	**93.54**	**96.14**	**97.41**
	$[0, 300]m$	DirectionNet [2]	47.22	71.78	73.97	71.85	5.46	12.21	20.84
		LOFTR + MAGSAC [1, 13]	43.19	68.59	76.80	60.62	20.60	31.73	40.15
		RPNet [6]	44.88	70.37	84.58	65.49	12.26	24.18	34.23
		SIFT + LO-RANSAC [3, 9]	49.22	73.17	91.17	73.76	9.79	19.03	27.86
		SuperGlue + MAGSAC [1, 12]	43.48	68.87	77.74	60.85	19.68	30.74	39.27
		ViTPose [11]	25.39	48.54	49.56	30.19	60.74	72.69	79.06
		Proposed	**9.53**	**24.04**	**15.99**	**14.20**	**92.53**	**95.47**	**96.83**

Table 5. Stereo depth evaluation on **Semi-Truck Highway** dataset using **DLNR-Middlebury** and **HITNet** stereo models.

Stereo Model	Depth Range	Method	lower is better				higher is better		
			MAE	RMSE	SILog	absrel	d<1.25	d<1.25^2	d<1.25^3
DLNR [15]	$[0, 80]m$	DirectionNet [2]	5.12	9.11	27.80	18.61	75.58	88.10	92.23
		LOFTR + MAGSAC [1, 13]	3.34	6.10	17.50	12.13	87.02	92.19	94.11
		RPNet [6]	5.86	10.00	27.73	21.85	70.67	89.10	94.18
		SIFT + LO-RANSAC [3, 9]	15.78	20.84	64.55	74.01	4.86	9.56	14.11
		SuperGlue + MAGSAC [1, 12]	5.41	9.02	28.76	21.03	72.24	79.03	82.45
		ViTPose [11]	2.10	4.22	10.01	7.55	95.21	98.39	99.30
		Proposed	**0.97**	**2.59**	**8.94**	**3.50**	**96.88**	**98.91**	**99.53**
	$[0, 100]m$	DirectionNet [2]	5.89	10.88	29.21	19.45	74.41	87.10	91.65
		LOFTR + MAGSAC [1, 13]	3.81	7.21	18.30	12.53	86.03	91.87	93.89
		RPNet [6]	6.74	12.04	29.41	22.87	69.46	87.67	93.51
		SIFT + LO-RANSAC [3, 9]	16.93	23.07	66.15	74.39	4.77	9.39	13.86
		SuperGlue + MAGSAC [1, 12]	6.05	10.45	29.87	21.48	71.46	78.52	82.05
		ViTPose [11]	2.43	5.10	10.60	7.89	94.43	98.20	99.20
		Proposed	**1.15**	**3.10**	**9.31**	**3.66**	**96.63**	**98.79**	**99.47**
	$[0, 200]m$	DirectionNet [2]	8.14	17.35	32.21	21.51	72.72	85.19	90.19
		LOFTR + MAGSAC [1, 13]	4.94	10.52	20.04	13.20	84.66	91.10	93.48
		RPNet [6]	9.51	20.01	33.13	25.53	67.72	85.27	91.20
		SIFT + LO-RANSAC [3, 9]	19.23	28.45	69.11	74.99	4.65	9.17	13.55
		SuperGlue + MAGSAC [1, 12]	7.47	14.27	31.98	22.15	70.41	77.70	81.43
		ViTPose [11]	3.33	8.00	11.98	8.55	92.94	97.68	98.93
		Proposed	**1.60**	**4.70**	**9.98**	**3.96**	**96.22**	**98.57**	**99.34**
	$[0, 300]m$	DirectionNet [2]	9.44	22.09	33.49	22.47	72.42	84.78	89.66
		LOFTR + MAGSAC [1, 13]	5.47	12.42	20.64	13.40	84.39	90.84	93.30
		RPNet [6]	11.11	25.93	34.77	26.77	67.44	84.85	90.60
		SIFT + LO-RANSAC [3, 9]	20.13	31.06	70.03	75.17	4.63	9.12	13.49
		SuperGlue + MAGSAC [1, 12]	8.11	16.36	32.71	22.32	70.21	77.49	81.24
		ViTPose [11]	3.79	9.78	12.50	8.80	92.63	97.42	98.80
		Proposed	**1.82**	**5.66**	**10.17**	**4.05**	**96.09**	**98.51**	**99.30**
HITNet [14]	$[0, 80]m$	DirectionNet [2]	4.58	8.21	17.65	16.84	78.86	93.92	97.69
		LOFTR + MAGSAC [1, 13]	2.61	5.02	13.72	9.72	91.09	95.77	97.14
		RPNet [6]	5.94	9.95	16.05	22.42	69.20	92.84	99.12
		SIFT + LO-RANSAC [3, 9]	15.84	21.24	80.87	78.53	6.50	12.74	18.48
		SuperGlue + MAGSAC [1, 12]	4.16	7.54	23.79	16.19	78.18	86.37	89.91
		ViTPose [11]	1.78	3.59	8.40	6.45	97.70	99.19	99.61
		Proposed	**0.66**	**2.21**	**7.47**	**2.70**	**98.63**	**99.41**	**99.70**
	$[0, 100]m$	DirectionNet [2]	5.60	10.67	19.55	18.25	77.17	92.25	96.97
		LOFTR + MAGSAC [1, 13]	3.07	6.28	14.70	10.14	89.90	95.44	96.96
		RPNet [6]	7.35	13.15	18.59	24.47	67.37	89.99	98.00
		SIFT + LO-RANSAC [3, 9]	17.12	23.79	83.16	79.59	6.38	12.49	18.15
		SuperGlue + MAGSAC [1, 12]	4.78	9.11	25.22	16.69	77.24	85.75	89.49
		ViTPose [11]	2.12	4.60	9.05	6.80	96.67	99.07	99.56
		Proposed	**0.81**	**2.82**	**7.92**	**2.87**	**98.46**	**99.34**	**99.67**
	$[0, 200]m$	DirectionNet [2]	10.56	24.85	26.92	23.79	74.22	88.33	93.30
		LOFTR + MAGSAC [1, 13]	4.83	12.08	18.04	11.25	87.70	93.97	96.18
		RPNet [6]	14.09	31.19	28.09	32.54	64.51	85.18	91.79
		SIFT + LO-RANSAC [3, 9]	21.05	33.29	88.71	83.34	6.15	12.06	17.54
		SuperGlue + MAGSAC [1, 12]	6.96	15.72	29.39	17.94	75.49	84.18	88.24
		ViTPose [11]	3.50	9.62	11.57	7.82	94.18	98.24	99.19
		Proposed	**1.54**	**6.22**	**9.60**	**3.49**	**97.83**	**99.01**	**99.47**
	$[0, 300]m$	DirectionNet [2]	15.19	39.74	31.83	28.03	73.42	87.25	91.84
		LOFTR + MAGSAC [1, 13]	6.46	18.01	20.37	11.86	86.98	93.23	95.54
		RPNet [6]	20.48	50.30	34.56	38.90	63.78	84.04	90.12
		SIFT + LO-RANSAC [3, 9]	23.96	41.75	91.36	86.18	6.08	11.92	17.33
		SuperGlue + MAGSAC [1, 12]	8.87	22.19	31.88	18.61	74.90	83.52	87.59
		ViTPose [11]	4.76	14.77	13.43	8.42	93.36	97.49	98.75
		Proposed	**2.37**	**9.90**	**10.81**	**3.93**	**97.32**	**98.75**	**99.30**

Table 6. Stereo depth evaluation on **KITTI** dataset using **DLNR-Middlebury** and **HITNet** stereo models.

Stereo Model	Depth Range	Method	lower is better				higher is better		
			MAE	RMSE	SILog	absrel	d<1.25	d<1.25^2	d<1.25^3
DLNR [15]	$[0, 80]m$	DirectionNet [2]	18.19	24.41	61.00	57.79	15.15	26.98	36.52
		LOFTR + MAGSAC [1, 13]	8.98	13.85	35.24	26.41	62.29	69.60	74.44
		RPNet [6]	16.08	22.44	61.23	49.54	24.01	39.87	50.73
		SIFT + LO-RANSAC [3, 9]	20.97	26.74	60.47	67.25	9.10	16.80	23.81
		SuperGlue + MAGSAC [1, 12]	10.37	15.65	41.00	30.81	55.93	64.58	70.18
		ViTPose [11]	5.91	10.43	27.90	16.20	77.74	86.42	90.08
		Proposed	**4.06**	**6.95**	**12.78**	**11.40**	**86.48**	**96.55**	**98.75**
	$[0, 100]m$	DirectionNet [2]	22.82	31.75	66.42	59.88	14.00	24.99	33.90
		LOFTR + MAGSAC [1, 13]	11.89	18.85	39.83	28.30	60.28	67.47	72.26
		RPNet [6]	20.52	29.58	67.22	51.96	22.25	37.06	47.32
		SIFT + LO-RANSAC [3, 9]	25.86	34.25	65.31	68.96	8.41	15.51	22.03
		SuperGlue + MAGSAC [1, 12]	13.61	21.12	45.78	32.89	53.50	62.20	67.67
		ViTPose [11]	8.25	14.82	32.68	18.08	74.44	83.98	88.10
		Proposed	**5.63**	**9.84**	**15.00**	**12.75**	**83.13**	**95.01**	**98.10**
	$[0, 200]m$	DirectionNet [2]	44.50	64.16	83.53	66.13	10.98	19.70	26.93
		LOFTR + MAGSAC [1, 13]	26.66	42.65	54.69	34.71	53.28	60.78	65.34
		RPNet [6]	41.61	61.45	85.68	59.15	17.71	29.69	38.22
		SIFT + LO-RANSAC [3, 9]	48.48	67.35	80.78	73.93	6.68	12.34	17.58
		SuperGlue + MAGSAC [1, 12]	30.39	47.67	61.71	39.86	45.75	54.81	60.10
		ViTPose [11]	20.53	35.95	47.10	24.51	63.72	75.87	81.47
		Proposed	**15.42**	**26.55**	**23.25**	**18.55**	**71.27**	**87.05**	**94.05**
	$[0, 300]m$	DirectionNet [2]	51.54	76.85	87.36	67.24	10.54	18.89	25.85
		LOFTR + MAGSAC [1, 13]	32.33	53.78	58.69	36.17	51.76	59.27	63.90
		RPNet [6]	48.67	74.27	89.76	60.53	17.02	28.53	36.75
		SIFT + LO-RANSAC [3, 9]	55.86	80.43	84.82	74.97	6.37	11.74	16.79
		SuperGlue + MAGSAC [1, 12]	36.36	59.21	65.75	41.34	44.24	53.21	58.58
		ViTPose [11]	25.58	46.14	50.50	26.14	61.57	73.71	79.70
		Proposed	**21.17**	**38.11**	**26.93**	**21.21**	**68.52**	**83.67**	**91.48**
HITNet [14]	$[0, 80]m$	DirectionNet [2]	17.12	24.14	76.69	59.66	24.13	43.34	57.70
		LOFTR + MAGSAC [1, 13]	8.22	13.30	37.37	26.28	66.75	77.07	83.51
		RPNet [6]	15.70	22.60	68.43	54.86	31.28	52.84	66.95
		SIFT + LO-RANSAC [3, 9]	20.36	26.56	73.35	66.74	13.15	24.55	34.35
		SuperGlue + MAGSAC [1, 12]	9.33	14.84	42.44	30.49	62.13	75.11	83.14
		ViTPose [11]	5.96	10.66	26.60	18.11	78.09	88.05	92.57
		Proposed	**4.20**	**6.98**	**11.77**	**12.11**	**85.68**	**96.34**	**98.72**
	$[0, 100]m$	DirectionNet [2]	21.80	31.58	86.46	63.39	22.50	40.37	53.82
		LOFTR + MAGSAC [1, 13]	11.04	18.19	43.54	28.56	64.62	74.55	80.78
		RPNet [6]	20.12	29.71	78.13	58.51	29.22	49.40	62.71
		SIFT + LO-RANSAC [3, 9]	25.23	34.12	81.95	69.07	12.18	22.72	31.86
		SuperGlue + MAGSAC [1, 12]	12.52	20.33	49.67	33.12	59.39	72.12	79.85
		ViTPose [11]	8.22	14.96	31.76	20.06	74.98	85.61	90.44
		Proposed	**5.73**	**9.79**	**13.67**	**13.47**	**82.46**	**94.87**	**98.12**
	$[0, 200]m$	DirectionNet [2]	42.64	63.28	111.36	73.26	18.00	32.58	43.80
		LOFTR + MAGSAC [1, 13]	24.46	40.41	60.06	35.28	57.86	67.74	73.38
		RPNet [6]	40.40	60.96	102.74	68.72	23.49	40.12	51.44
		SIFT + LO-RANSAC [3, 9]	46.91	66.43	104.37	75.28	9.81	18.35	25.80
		SuperGlue + MAGSAC [1, 12]	28.15	45.71	69.46	40.63	51.04	63.54	70.79
		ViTPose [11]	19.33	34.46	45.81	26.06	65.41	78.51	84.44
		Proposed	**15.55**	**26.78**	**21.50**	**19.66**	**70.71**	**87.02**	**94.28**
	$[0, 300]m$	DirectionNet [2]	51.89	80.65	119.13	78.33	17.20	31.03	41.69
		LOFTR + MAGSAC [1, 13]	31.18	53.96	65.94	37.93	55.89	65.74	71.48
		RPNet [6]	49.73	78.58	110.64	74.08	22.44	38.27	49.03
		SIFT + LO-RANSAC [3, 9]	55.52	82.09	111.41	77.38	9.33	17.40	24.52
		SuperGlue + MAGSAC [1, 12]	35.54	60.29	75.77	43.50	48.98	61.15	68.51
		ViTPose [11]	25.32	46.70	50.48	28.48	62.90	75.90	82.46
		Proposed	**22.65**	**40.81**	**25.78**	**23.34**	**67.52**	**82.94**	**91.17**

Table 7. Stereo depth evaluation on **CARLA** dataset using **DLNR-Middlebury** and **HITNet** stereo models.

References

[1] Dániel Baráth, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1301–1309, 2019. 6, 7, 10, 11, 12, 13, 14, 15

[2] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3257–3267, 2021. 6, 7, 13, 14, 15

[3] Ondřej Chum, Jiri Matas, and Josef Kittler. Locally optimized ransac. In *DAGM-Symposium*, 2003. 6, 7, 13, 14, 15

[4] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 9

[5] D. Ponsa E. Rublee E. Riba, D. Mishkin and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Winter Conference on Applications of Computer Vision*, 2020. 1, 2

[6] Sovann En, Alexis Lechervy, and Frédéric Jurie. Rpnet: an end-to-end network for relative camera pose estimation. In *ECCV Workshops*, 2018. 6, 7, 10, 11, 12, 13, 14, 15

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 4, 5, 6, 10

[8] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *ArXiv*, abs/1506.02025, 2015. 2

[9] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004. 6, 7, 13, 14, 15

[10] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 1

[11] C. Rockwell, Justin Johnson, and David F. Fouhey. The 8-point algorithm as an inductive bias for relative pose prediction by vits. *2022 International Conference on 3D Vision (3DV)*, pages 1–11, 2022. 6, 7, 10, 11, 12, 13, 14, 15

[12] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. *CoRR*, abs/1911.11763, 2019. 6, 7, 10, 11, 12, 13, 14, 15

[13] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8918–8927, 2021. 6, 7, 10, 11, 12, 13, 14, 15

[14] Vladimir Tankovich, Christian Häne, S. Fanello, Yinda Zhang, Shahram Izadi, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14357–14367, 2020. 4, 5, 10, 11, 12, 13, 14, 15

[15] Haoliang Zhao, Huizhou Zhou, Yongjun Zhang, Jie Chen, Yitong Yang, and Yong Zhao. High-frequency stereo matching network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1327–1336, 2023. 4, 13, 14, 15