GLU-Net: Global-Local Universal Network for Dense Flow and Correspondences

Supplementary Material

Prune Truong Martin Danelljan Radu Timofte
Computer Vision Lab, D-ITET, ETH Zürich, Switzerland
{prune.truong, martin.danelljan, radu.timofte}@vision.ee.ethz.ch

In this supplementary material, we first provide details about the architecture of the different modules of our network GLU-Net in Section 1. We then explain the training procedure in more depth in Section 2. Finally, we present additional qualitative results and more detailed quantitative experiments in Section 3.

1. Architecture details

In this section, we provide additional details about cyclic consistency as a post processing step of the global correlation. We also give a detailed architectural description of the mapping and flow decoders, along with the refinement network. Lastly, we explain in depth the iterative refinement allowed by our adaptive resolution strategy. In the following, a convolution layer or block refers to the composition of a 2D-convolution followed by batch norm [8] and ReLU [14] (Conv-BN-ReLU).

1.1. Cyclic consistency post-processing step for improved global correlation

Since the quality of the correlation layer output is of primary importance for the flow estimation process, we introduce an additional filtering step on the global cost volume to enforce the reciprocity constraint on matches. To encourage matched features to be mutual nearest neighbours, we employ the soft mutual nearest neighbor filtering introduced by [17] and apply it to post-process the global correlation.

The soft mutual nearest neighbor module filters a global correlation $C \in \mathbb{R}^{H\times W\times H\times W}$ into $\hat{C} \in \mathbb{R}^{H\times W\times H\times W}$ such that:

$$\hat{C}(i,j,k,l) = r_t(i,j,k,l) \cdot r_s(i,j,k,l) \cdot C(i,j,k,l) \tag{1}$$

with $r_s(i,j,k,l)$ and $r_t(i,j,k,l)$ the ratios of the score of the particular match $C(i,j,k,l)$ with the best scores along each pair of dimensions corresponding to images I_s and I_t respectively. We present the formula for $r_s(i,j,k,l)$ below, the same applies for $r_t(i,j,k,l)$.

$$r_t(i,j,k,l) = \frac{C(i,j,k,l)}{\max_{ab} C(a,b,k,l)} \tag{2}$$

This cyclic consistency post-processing step does not add any training weights.

1.2. Mapping decoder M_top

In this sub-section, we give additional details of the mapping decoder M_top for the global correlation layer (Eq. 4 and Figure 3 in the paper). We compute a global correlation from the L^2-normalized source and target features. The cost volume is further post-processed by applying channel-wise L^2-normalisation followed by ReLU [14] to strongly down-weight ambiguous matches [16]. Similar to DGC-Net [13], the resulting global correlation layer C is then fed into a correspondence map decoder M_top to estimate a 2D dense correspondence map $\mathbf{m}$ at the coarsest level L_1 of the feature pyramid,

$$\mathbf{m}^1 = M_\text{top}\left(C\left(\frac{F^1_\text{t}}{\|F^1_\text{t}\|}, \frac{F^1_\text{s}}{\|F^1_\text{s}\|}\right)\right) . \tag{3}$$

The outputted mapping estimate is parameterized such that each predicted pixel location in the map belongs to the interval $[-1; 1]$ representing width and height normalized image coordinates. The correspondence map is then re-scaled to image coordinates and converted to a displacement field.

$$\mathbf{w}^1(\mathbf{x}) = \mathbf{m}^1(\mathbf{x}) - \mathbf{x}\,. \tag{4}$$

The decoder M_{top} consists of 5 feed-forward convolutional blocks with a 3×3 spatial kernel. The number of feature channels of each convolutional layers are respectively 128, 128, 96, 64, and 32. The final output of the mapping decoder is the result of a linear 2D convolution, without any activation.

1.3. Flow decoder M

Here, we give additional details of the flow decoder M for the local correlation layers (Eq. 5 and Figure 3 in the paper). At level l, the flow decoder M infers the residual flow $\Delta\tilde{\mathbf{w}}^l$ as,

$$\Delta\tilde{\mathbf{w}}^l = M\left(c\left(F_{\mathrm{t}}^l, \widetilde{F}_{\mathrm{s}}^l; R\right), \mathrm{up}(\mathbf{w}^{l-1})\right)\,. \tag{5}$$

Here, c is a local correlation volume with search radius R and $\widetilde{F}_{\mathrm{s}}^l(\mathbf{x}) = F_{\mathrm{s}}^l\left(\mathbf{x} + \mathrm{up}\left(\mathbf{w}^{l-1}\right)(\mathbf{x})\right)$ is the warped source feature map F_{s} according to the upsampled flow from the previous pyramid level $\mathrm{up}\left(\mathbf{w}^{l-1}\right)$. The complete flow field is then computed as $\tilde{\mathbf{w}}^l = \Delta\tilde{\mathbf{w}}^l + \mathrm{up}\left(\mathbf{w}^{l-1}\right)$.

The flow decoder at level 4 (see Figure 3 of main paper) additionally takes an input $\mathrm{de}_2(f^{l-1})$, obtained by applying a transposed convolution layer de_2 to the features f^{l-1} of the second last layer from the flow decoder M^{l-1}. This additional inputs was first introduced and utilized in PWC-Net [21] at every pyramid level. It enables the decoder of the current level to obtain some information about the correlation at the previous level. In GLU-Net, this additional input to the flow decoder only appears in H-Net since in L-Net, a global correlation and mapping decoder precede the flow decoder.

As for the flow decoder M, we employ a similar architecture to the one in PWC-Net [21]. It consists of 5 convolutional layers with DenseNet connections [5]. The numbers of feature channels at each convolutional layers are respectively 128, 128, 96, 64, and 32, and the spatial kernel of each convolution is 3×3. DenseNet connections are used since they have been shown to lead to significant improvement in image classification [5] and optical flow estimation [21]. The final output of the flow decoder is the result of a linear 2D convolution, without any activation.

1.4. Refinement network R

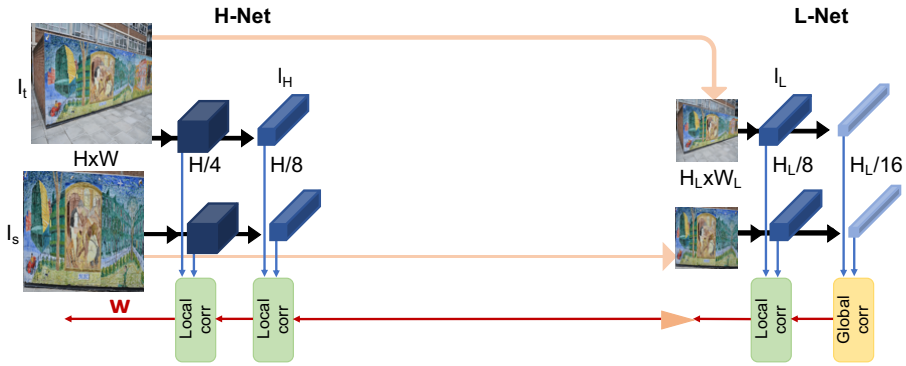
Here, we explain in more details the refinement network R (Figure 3 in the paper). The refinement network aims to refine the pixel-level flow field $\tilde{\mathbf{w}}^l$, thus preventing erroneous flows from being amplified by up-sampling and passing to the next pyramid level. Its architecture is the same than the context network employed in PWC-Net [21]. It is a feed-forward CNN with 7 dilated convolutional layers [20], with varying dilation rates. Dilated convolutions enlarge the receptive field without increasing the number of weights. From bottom to top, the dilation constants are 1, 2, 4, 8, 16, 1, and 1. The spatial kernel is set to 3×3 for all convolutional layers.

1.5. Details about Local-net, Global-Net and GLOCAL-Net

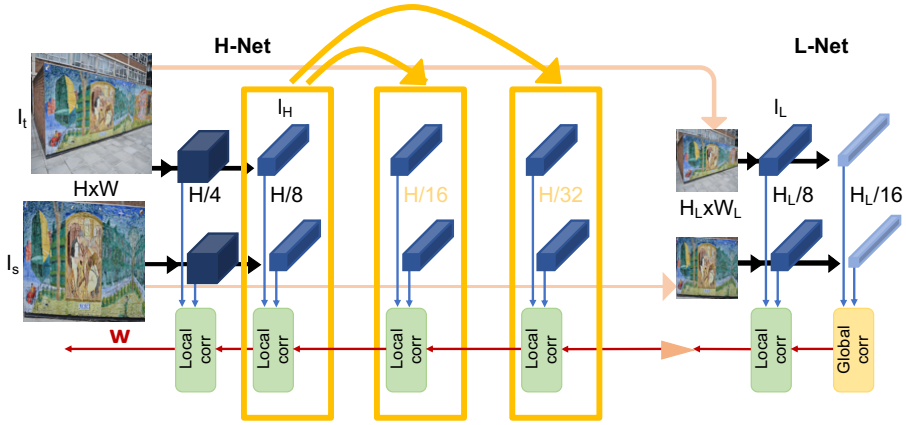
In Figure 2 of the main paper, we introduced Local-Net, Global-Net and GLOCAL-Net to investigate the differences between architectures based on local correlation layers, a global correlation layer or a combination of the two, respectively. All three networks are composed of three pyramid levels and use the same feature extractor backbone VGG-16 [2]. The mapping and flow decoders have the same architecture as those used for GLU-Net and described above. For Global-Net, the pyramid levels following the global correlation level employ concatenation of the target and warped source feature maps, as suggested in DGC-Net [13]. They are fed to the flow estimation decoder along with the up-sampled flow from the previous resolution. Finally, Global-Net and GLOCAL-Net are both restricted to a pre-determined input resolution $H_L \times W_L$ due to their global correlation at the coarsest pyramid level. On the other hand, Local-Net, which only relies on local correlations, can take input images of any resolutions.

1.6. Iterative refinement

Here we provide more details about the iterative refinement procedure described in Section 3.3 in the paper. For high-resolution images, the upscaling factor between the finest pyramid level, l_L, of L-Net and the coarsest, l_H, of H-Net (see Figure 1) can be significant. Our adaptive resolution strategy allows additional refinement steps of the flow estimate between



(a) GLU-Net without iterative refinement.



(b) GLU-Net with iterative refinement between L-Net and H-Net.

Figure 1. Schematic representation of iterative refinement. The features and weights of l_H level of H-Net are iteratively applied at intermediate resolutions between L-Net and H-Net.

those two levels during inference, thus improving the accuracy of the estimated flow, without training any additional weights. This is performed by recursively applying the l_H layer weights at intermediate resolutions obtained by down-sampling the source and target feature maps from l_H.

Particularly, we apply iterative refinement if the ratio between the resolutions of the l_H and l_L levels is larger than three. We then iteratively perform refinements at intermediate resolutions, obtained by a reduction of factor 2 from l_H in each step, until the ratio between the resolution of the coarsest intermediate level and the resolution of l_L is smaller than 2.

In more details, we construct a local correlation layer from the source and target feature maps of level l_H down-sampled to the desired intermediate resolution. We then apply the weights of the level l_H decoder to the local correlation, therefore obtaining an intermediate refinement of the flow field. This process is illustrated in Figure 1, where the gap between l_L and l_H here allows for two additional flow field refinements.

2. Training details

Here, we provide additional details about the training procedure and the training dataset.

2.1. Loss

We freeze the weights of the feature extractor during training. Let θ denote the learnable parameters of the network. Let $\mathbf{w}_\theta^l = (\mathbf{w_x}^l, \mathbf{w_y}^l) \in \mathbb{R}^{H_l \times W_l \times 2}$ denote the flow field estimated by the network at the l^{th} pyramid level. $\mathbf{w}_{\mathrm{GT}}^l$ refers to the corresponding dense flow ground-truth, computed from the random warp. We employ the multi-scale training loss, first

introduced in FlowNet [4],

$$\mathcal{L}(\theta) = \sum_{l=L_1}^{L} \alpha_l \sum_{\mathbf{x}} \left\| \mathbf{w}_\theta^l(\mathbf{x}) - \mathbf{w}_{\mathrm{GT}}^l(\mathbf{x}) \right\| + \gamma \left\| \theta \right\| , \tag{6}$$

where α_l are the weights applied to each pyramid level and the second term of the loss regularizes the weights of the network. We do not apply any mask during training, which means that occluded regions (that do not have visible matches) are included in the training loss. Since the image pairs are related by synthetic transformations, these regions do have a correct ground-truth flow value.

For our adaptive resolution strategy, we down-sample and scale the ground truth from original resolution $H \times W$ to $H_L \times W_L$ in order to obtain the ground truth flow fields for L-Net. Similarly to FlowNet [4] and PWC-Net [21], we down-sample the ground truth from the base resolution to the different pyramid resolutions without further scaling, so as to obtain the supervision signals at the different levels.

2.2. Dataset

To use the full potential of our GLU-Net, training should be performed on high-resolution images. We create the training dataset following the procedure in DGC-Net [13], but enforcing the condition of high resolution. We use the same $40,000$ synthetic transformations (affine, thin-plate and homographies), but apply them to our higher resolution images collected from the DPED [7], CityScapes [3] and ADE-20K [24] datasets. Indeed, DPED images are very large, however the DPED training dataset is composed of only approximately 5000 sets of images taken by four different cameras. We use the images from two cameras, resulting in around $10,000$ images. CityScapes additionally adds about $23,000$ images. We complement with a random sample of ADE-20K images with a minimum resolution of 750×750.

2.3. Implementation details

As a preprocessing step, the training images are mean-centered and normalized using mean and standard deviation of ImageNet dataset [11]. For all local correlation layers, we employ a search radius $R = 4$. For the training of Global-Net and GLOCAL-Net, which both have a pre-determined fixed input image resolution of $(H_L \times W_L = 256 \times 256)$, we use a batch size of 32 while we train LOCAL-Net, which can take any input image, with batches of size 16. We set the initial learning rate to 10^{-2} and gradually decrease it during training. The weights in the training loss are set to be $\alpha_1 = 0.32, \alpha_2 = 0.08, \alpha_3 = 0.02$.

Our final network GLU-Net is trained with a batch size of 16 and the learning rate initially equal to 10^{-4}. The weights in the training loss are set to be $\alpha_1 = 0.32, \alpha_2 = 0.08, \alpha_3 = 0.02, \alpha_4 = 0.01$. Our system is implemented using Pytorch [15] and our networks are trained using Adam optimizer [10] with learning rate decay of 0.0004.

3. Detailed results

Here, we first provide additional details on the run-time computation in Section 3.1. Then, in Section 3.2, we evaluate the influence of the training dataset on the evaluation results. We then present additional qualitative and more detailed quantitative results on subsequently the geometric matching, the semantic matching and the optical flow tasks in respectively Sections 3.3, 3.4 and 3.5. Finally, we expose additional ablation experiments in Section 3.6.

3.1. Run time

We compare the run time of our method with state-of-the-art approaches over the HP-240 images in Table 1. The timings have been obtained on the same desktop with an NVIDIA GTX 1080 Ti GPU. The HP-240 images are of size 240×240, which corresponds to the pre-determined input resolution of DGC-Net. For PWC-Net, LiteFlowNet and GLU-Net, the images are resized to 256×256 before being passed through the networks. We do not consider this resizing in the estimated time. They all output a flow at a quarter resolution the input image. We up-scale to the image resolution 240×240 with bilinear interpolation. This up-scaling operation is included in the estimated time.

	PWC-Net	LiteFlowNet	DGC-Net	GLU-Net (Ours)
Run-time [ms]	38.51	45.10	138.30	**38.10**

Table 1. Run time of our methods compared to optical-flow competitors PWC-Net and LiteFlowNet as well as geometric matching competitor DGC-Net, averaged over 295 image pairs of HP-240.

	HP-240x240			HP			KITTI-2012		KITTI-2015	
	AEPE	PCK-1px [%]	PCK-5px [%]	AEPE	PCK-1px [%]	PCK-5px [%]	AEPE-all	F1-all [%]	AEPE-all	F1-all [%]
DGC-Net (*tokyo*)	9.07	50.01	77.40	33.26	12.00	58.06	8.50	32.38	14.97	50.98
DGC-Net† (*DPED-CityScape-ADE*)	9.12	43.09	79.35	33.47	9.19	56.02	7.96	34.35	14.33	50.35
GLU-Net (*DPED-CityScape-ADE*)	**7.40**	**59.92**	**83.47**	**25.05**	**39.55**	**78.54**	**3.34**	**18.93**	**9.79**	**37.52**

Table 2. Effect of the training dataset on the evaluation results of DGC-Net and comparison to GLU-Net. The training dataset is indicated in parenthesis.

Our network GLU-Net obtains similar run time than PWC-Net and is three times faster than DGC-Net. This is due to the fact that PWC-Net, LiteFlowNet and GLU-Net outputs a flow at a quarter image resolution whereas DGC-Net refines the estimated flow field with two additional pyramid levels until the fixed resolution of 240×240.

3.2. Training dataset

Since DGC-Net is our main competitor, for a fair comparison, we additionally trained DGC-Net on our training dataset *DPED-CityScape-ADE*, using the training code provided by the authors, which resulted in DGC-Net†. In Table 2, we summarize the results of DGC-Net trained on both *DPED-CityScape-ADE* or *tokyo* and evaluated on geometric matching datasets HP-240 and HP as well as optical flow datasets KITTI-2012 and KITTI-2015. It seems that the training dataset in this case only has a small effect. Since both datasets were created by applying the same synthetic transformations, this support the fact that geometric transformation and displacement statistics are more important for generalization properties than image content [12, 19, 22].

3.3. Geometric matching

We provide the detailed results on HP and ETH3D datasets, as well as extensive additional qualitative examples. We also analyse the performance of our network with respect to rotation and scaling.

3.3.1 Results on HPatches dataset

Detailed results obtained by different models on the various view-points of the HP and HP-240 datasets are presented in Table 3. It corresponds to Table 1 of the main paper, that only provides the average over all viewpoint IDs. Note that increasing view-point IDs lead to increasing geometric transformations due to larger changes in viewpoint. We outperform all other methods for each viewpoint ID on both low resolution (HP-240) and high-resolution images (HP). Particularly, our network permits to gain a lot of accuracy (in the order of 3 to 4 times higher for PCK-1 on HP) as compared to DGC-Net. Additional qualitative examples are shown in Figure 4.

We additionally present the PCK curves computed over the different viewpoints of HP, as a function of the relative distance threshold. We do not set a pixel-level thresholds for the PCK curves since HP image pairs have different resolutions

		HP-240						HP					
		I	II	III	IV	V	all	I	II	III	IV	V	all
LiteFlowNet	AEPE	6.99	16.78	19.13	25.27	28.89	19.41	36.69	102.17	113.58	154.97	186.82	118.85
	PCK-1px [%]	50.06	28.93	25.87	23.22	13.72	28.36	34.86	12.95	10.35	6.93	4.47	13.91
	PCK-5px [%]	82.14	59.62	56.92	51.04	38.59	57.66	63.99	32.88	28.99	18.52	13.85	31.64
PWC-Net	AEPE	5.74	17.69	20.46	27.61	36.97	21.68	23.93	76.33	91.30	124.22	164.91	96.14
	PCK-1px [%]	43.55	20.35	18.60	14.17	8.27	20.99	31.56	12.10	10.83	7.09	4.12	13.14
	PCK-5px [%]	80.06	57.08	53.89	45.70	34.22	54.19	68.79	38.51	36.38	25.24	16.76	37.14
DGC-Net	AEPE	1.74	5.88	9.07	12.14	16.50	9.07	5.71	20.48	34.15	43.94	62.01	33.26
	PCK-1px [%]	70.29	53.97	52.06	41.02	32.74	50.01	20.92	12.88	12.85	7.66	5.67	12.00
	PCK-5px [%]	93.70	82.43	77.58	71.53	61.78	77.40	78.88	63.37	60.21	48.83	38.99	58.06
DGC-Net†	AEPE	1.90	5.65	9.42	11.39	17.26	9.11	6.04	21.60	32.87	41.82	65.03	33.47
	PCK-1px [%]	60.88	47.88	46.01	34.87	25.80	43.09	15.81	9.86	9.84	6.17	4.29	9.19
	PCK-5px [%]	93.47	84.04	80.28	74.93	63.76	79.35	75.44	62.16	59.58	46.71	36.21	56.02
GLU-Net (Ours)	AEPE	**0.59**	**4.05**	**7.64**	**9.82**	**14.89**	**7.40**	**1.55**	**12.66**	**27.54**	**32.04**	**51.47**	**25.05**
	PCK-1px [%]	**87.89**	**67.49**	**62.31**	**47.76**	**34.14**	**59.92**	**61.72**	**42.43**	**40.57**	**29.47**	**23.55**	**39.55**
	PCK-5px [%]	**99.14**	**92.39**	**85.87**	**78.10**	**61.84**	**83.47**	**96.15**	**84.35**	**79.46**	**73.80**	**58.92**	**78.54**

Table 3. Details of AEPE and PCK evaluated over each view-point ID of HP and HP-240 datasets.

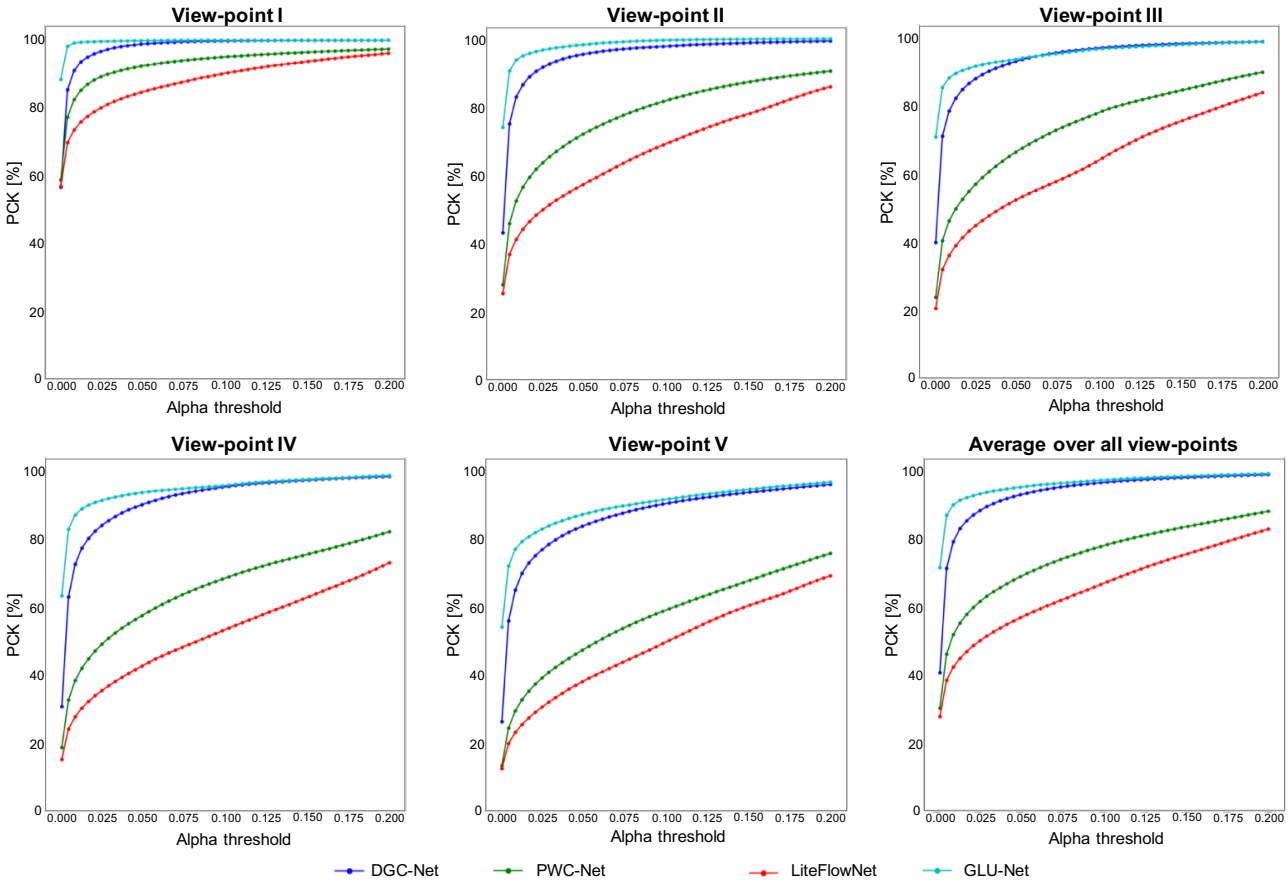


Figure 2. PCK curves obtained by state-of-the-art methods and GLU-Net over the different view-points of HP.

in general. GLU-Net achieves better accuracy (better PCK) for all thresholds compared to PWC-Net [21], LiteFLowNet [6] and DGC-Net [13]. Importantly, GLU-Net obtains significantly better PCK for low thresholds.

3.3.2 Results on ETH3D

In the main paper, Figure 5, we quantitatively evaluated our approach over pairs of ETH3D images sampled from consecutive frames at different intervals. In Table 4, we give the corresponding detailed evaluation metrics (AEPE and PCK) obtained by PWC-Net, LiteFlowNet, DGC-Net, DGC-Net† and GLU-Net.

Here, we additionally provide qualitative examples of the different networks and GLU-Net applied to pairs of images at different intervals in Figure 6. It is visible that while optical flow methods achieve good results for low intervals, the warped source images according to their outputted flows get worst when increasing the intervals between image pairs. On the other hand, our model produces flow fields of constant quality.

Qualitative results: We additionally use ETH3D images to demonstrate the superiority of our approach to deal with extreme viewpoint changes on the one hand, and radical illumination and appearance variations on the other hand.

In addition to the medium resolution images evaluated previously, ETH3D [18] also provides several additional scenes taken with high-resolution cameras, acquiring images at 24 Megapixel (6048×4032). Since the images of a sequence are taken by a unique camera, consecutive pairs of images show only little lightning variations, however they are related by *very wide view-point changes*. As there are no ground-truth correspondences provided along with the images, we only evaluate qualitatively on consecutive pairs of images. The original images of 6048×4032 are down-samled by a factor of 2 for practical purposes. We show quantitative results over a few of those images in Figure 3. GLU-Net is capable of handling very large motions, where the other methods partly (DGC-Net) or completely fail (PWC-Net and LiteFlowNet).

On the other hand, our network can also handle large appearances changes due to variation in illumination or due to

		LiteFlowNet	PWC-Net	DGC-Net	DGC-Net†	GLU-Net (Ours)
	AEPE	**1.77**	1.84	2.53	2.80	2.06
interval = 3	PCK-1px [%]	**58.88**	54.14	31.50	25.71	47.47
	PCK-5px [%]	**92.65**	92.44	88.34	86.29	91.03
	AEPE	2.68	**2.18**	3.321	3.64	2.61
interval = 5	PCK-1px [%]	**53.64**	47.02	25.23	19.88	40.22
	PCK-5px [%]	**90.53**	90.53	83.07	80.85	87.74
	AEPE	6.13	**3.27**	4.212	4.70	3.54
interval = 7	PCK-1px [%]	**46.97**	39.69	20.90	15.86	34.41
	PCK-5px [%]	86.29	**86.88**	78.17	75.31	84.06
	AEPE	13.01	5.64	5.38	5.64	**4.28**
interval = 9	PCK-1px [%]	**39.54**	32.61	17.61	13.12	30.25
	PCK-5px [%]	78.34	**81.01**	73.58	70.35	80.58
	AEPE	29.72	14.39	6.81	7.16	**5.65**
interval = 11	PCK-1px [%]	**31.12**	26.15	14.88	11.15	26.54
	PCK-5px [%]	65.94	71.74	69.09	65.31	**76.61**
	AEPE	52.45	27.52	9.04	8.91	**7.59**
interval = 13	PCK-1px [%]	**24.82**	21.30	12.83	9.34	23.45
	PCK-5px [%]	54.94	63.07	64.10	60.24	**72.16**
	AEPE	74.99	43.44	12.25	12.46	**10.82**
interval = 15	PCK-1px [%]	19.90	17.03	10.69	7.82	**20.48**
	PCK-5px [%]	46.19	54.25	58.52	54.49	**67.64**

Table 4. Metrics evaluated over scenes of ETH3D with different intervals between consecutive pairs of images (taken by the same camera). Note that those results are the average over the different sequences of ETH3D dataset. Small AEPE and high PCK are better.

Figure 3. Qualitative examples of state-of-the-art methods applied to very high-resolution images of different scenes of ETH3D. The presented image pairs show substantial view-point changes, and thus *very large motions*.

the use of different optics. For this purpose, we utilize additional examples of pairs of images from ETH3D taken by *two different cameras* simultaneously. The camera of the first images has a field-of-view of 54 degrees while the other camera has a field of view of 83 degrees. They capture images at a resolution of 480×752 or 514×955 depending on the scenes and on the camera. The exposure settings of the cameras are set to automatic for all datasets, allowing the device to adapt to illumination changes. Qualitative examples of state-of-the-art methods and GLU-Net applied to such pairs of images are presented in Figure 5. GLU-Net is robust to changes in lightning conditions as well as to artifacts. While DGC-Net [13] obtains satisfactory results, the warped image according to its outputted flow is often blurry whereas we always obtain sharp, almost perfect warped source images.

Figure 4. Qualitative examples of different state-of-the-art algorithms and our GLU-Net applied to HP images. The source images are warped according to the flow fields outputted by the different networks. The warped source images should resemble the target images. Our method GLU-Net is robust to drastic view-point changes.

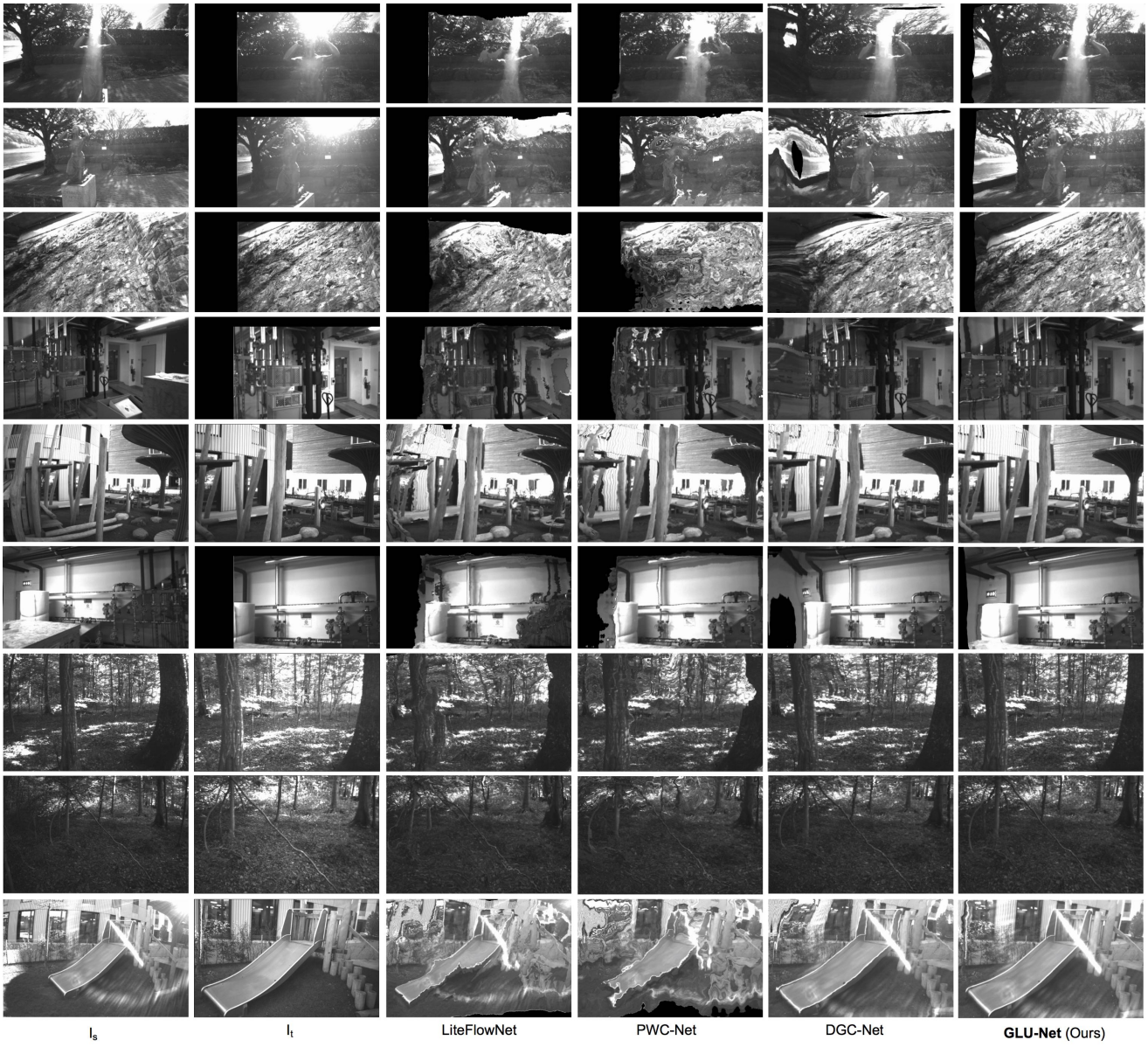


Figure 5. Qualitative examples of ETH3D pairs of images taken *simultaneously by two different cameras*. The two cameras have different field-of-views and sometimes different resolutions. Pairs of images experience drastic differences in lightning conditions. The source images are warped according to the flow fields outputted by different state-of-the-art networks and by our GLU-Net. The warped source images should resemble the target images.

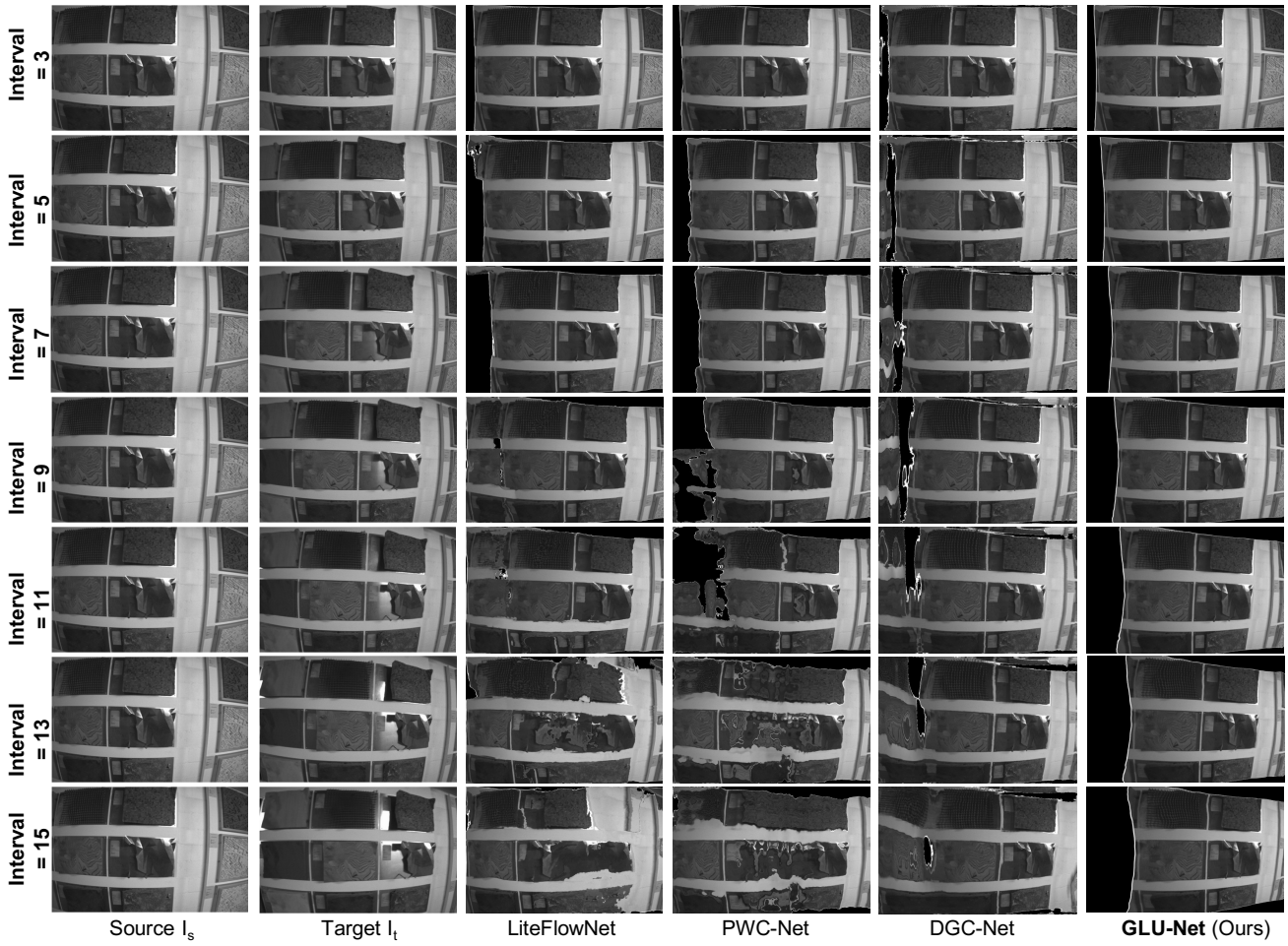


Figure 6. Qualitative examples of multiple networks and our GLU-Net applied to pairs of ETH3D dataset *taken at different intervals by the same camera*. The source images are warped according to the flow fields outputted by the different networks. The warped source images should resemble the target images. Optical flow methods obtain good qualitative results for low intervals (3 and 5) but largely degrade on bigger intervals. On the contrary, GLU-Net has a steady performance over all intervals.

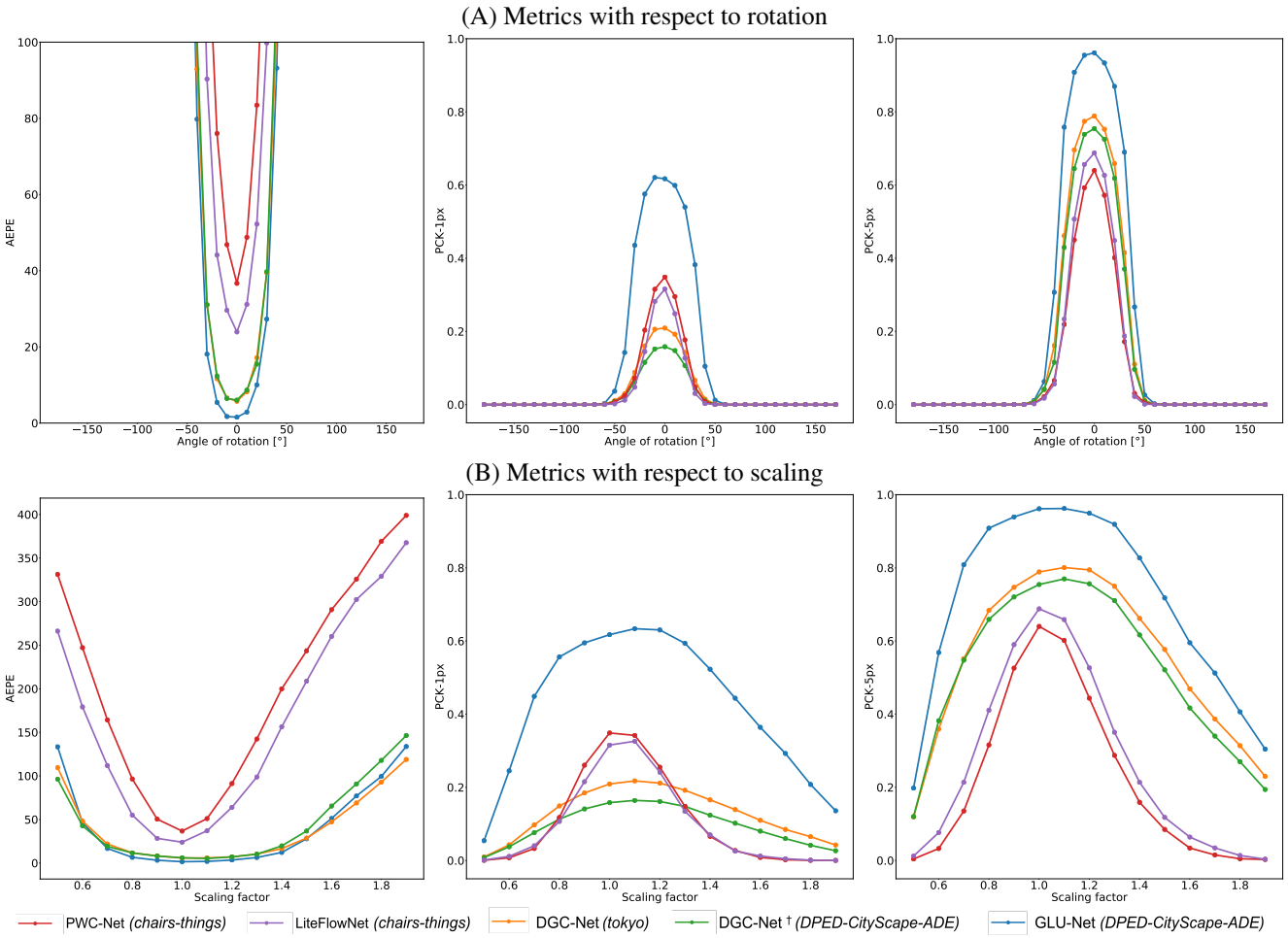


Figure 7. Quantitative results (AEPE, PCK-1px and PCK-5px) over the first viewpoint of HP, for different rotation and scaling factors applied to the target images and ground-truth flow fields. The training datasets are indicated in parenthesis for each model.

3.3.3 Rotation and scaling

We additionally measured the performance of our GLU-Net compared to state-of-the-art networks with respect to increasing rotation and scaling factors. To do so, we used the 59 pairs of the ViewPoint I of the HP [1] dataset as base images and applied increasingly high rotation and scaling factors to the target and ground-truth flow fields. In Figure 7, we plot the metrics (AEPE, PCK-1px and PCK-5px) obtained by GLU-Net, DGC-Net, DGC-Net†, PWC-Net and LiteFlowNet with respect to increasing applied rotation or scaling factors. For both rotation and scaling, while GLU-Net obtains similar AEPE than DGC-Net, its accuracy (PCK-1px and PCK-5px) is significantly above that of DGC-Net.

It must also be noted that GLU-Net is particularly robust and accurate for rotations up to +/- 50 degrees and scaling factors comprised between 0.8 and 1.4. This corresponds to the extent of geometric transformations present in the training dataset. Therefore, for improved robustness to larger rotations or scaling, image pairs experiencing such transformations should be additionally included in the training set.

3.4. Semantic correspondences

In Figure 10, we present additional qualitative results on the TSS [23] dataset of our universal network (GLU-Net) and its modified version (Semantic-GLU-Net), which includes NC-Net [17] and feature concatenation [9].

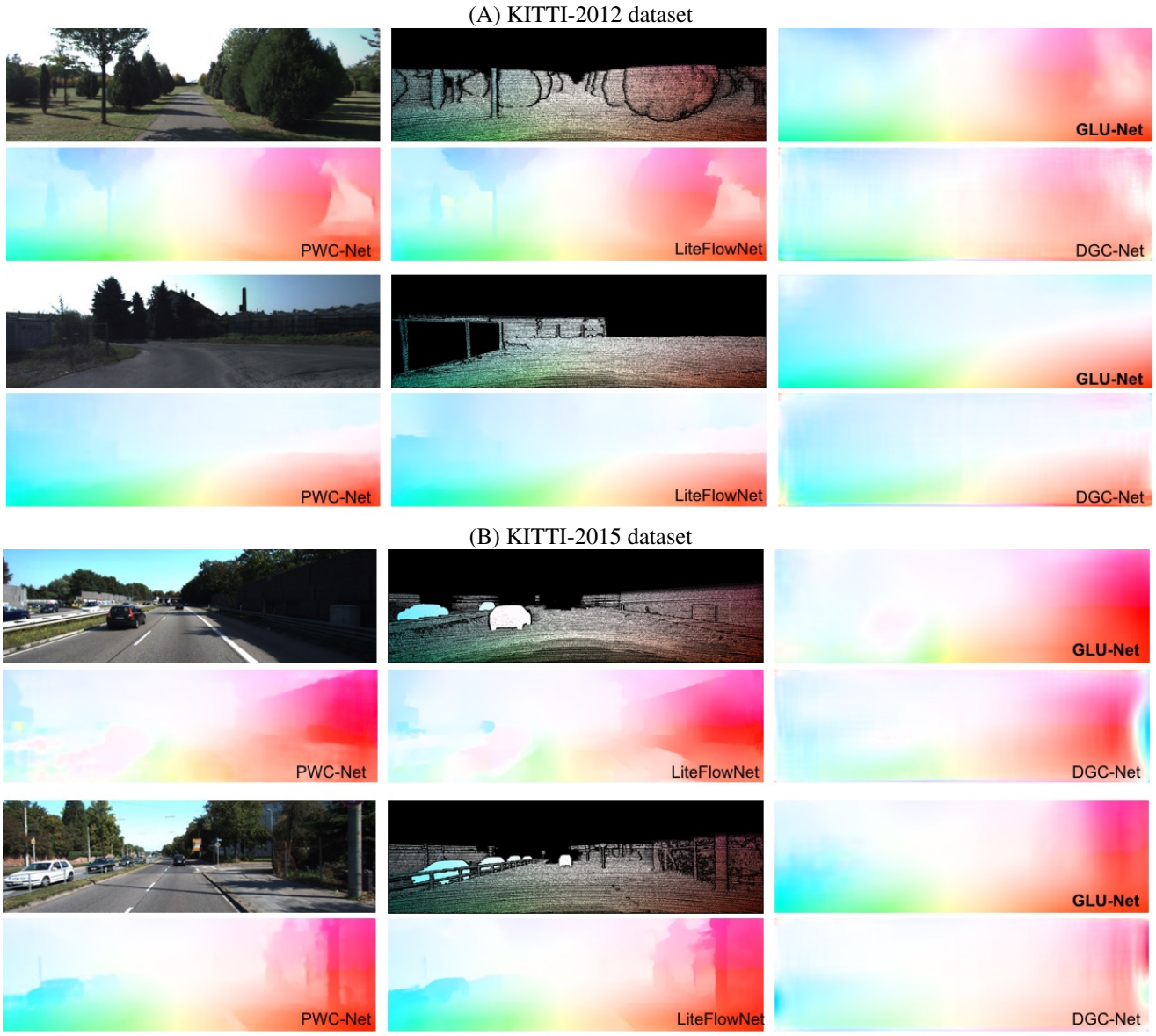


Figure 8. Representation of the flow fields estimated by state-of-the-art methods and GLU-Net applied to images of : (A) KITTI 2012 dataset, that is restricted to static scenes; (B) KITTI 2015 dataset, which comprises dynamic scenes.

3.5. Optical flow

Additional qualitative results: In Figure 8, we present additional qualitative examples of the estimated flow fields obtained by our method and competitors on the KITTI datasets. While our approach GLU-Net lacks accuracy at the object boundaries compared to the optical flow methods, our results are substantially better than those of DGC-Net, which is trained on the same kind of synthetic geometric transformations. As already stated, improved results, particularly at the object boundaries, could be obtained by including optical flow data with independently moving objects in the training set.

Supplementary analysis of the optical flow results: According to Table 3 and Figure 5 of the main paper, our GLU-Net obtains better AEPE than the optical flow methods PWC-Net and LiteFlowNet on the KITTI datasets (Table 3), but worst AEPE on the first intervals of ETH3D (Figure 5). The reasons for this behavior are explained below. As observed in Figure 9, while KITTI and ETH3D pairs for small intervals show similar average displacement, the KITTI datasets have a much wider distribution of displacements due to moving objects and the fast camera forward motion. Besides, our GLU-Net performs

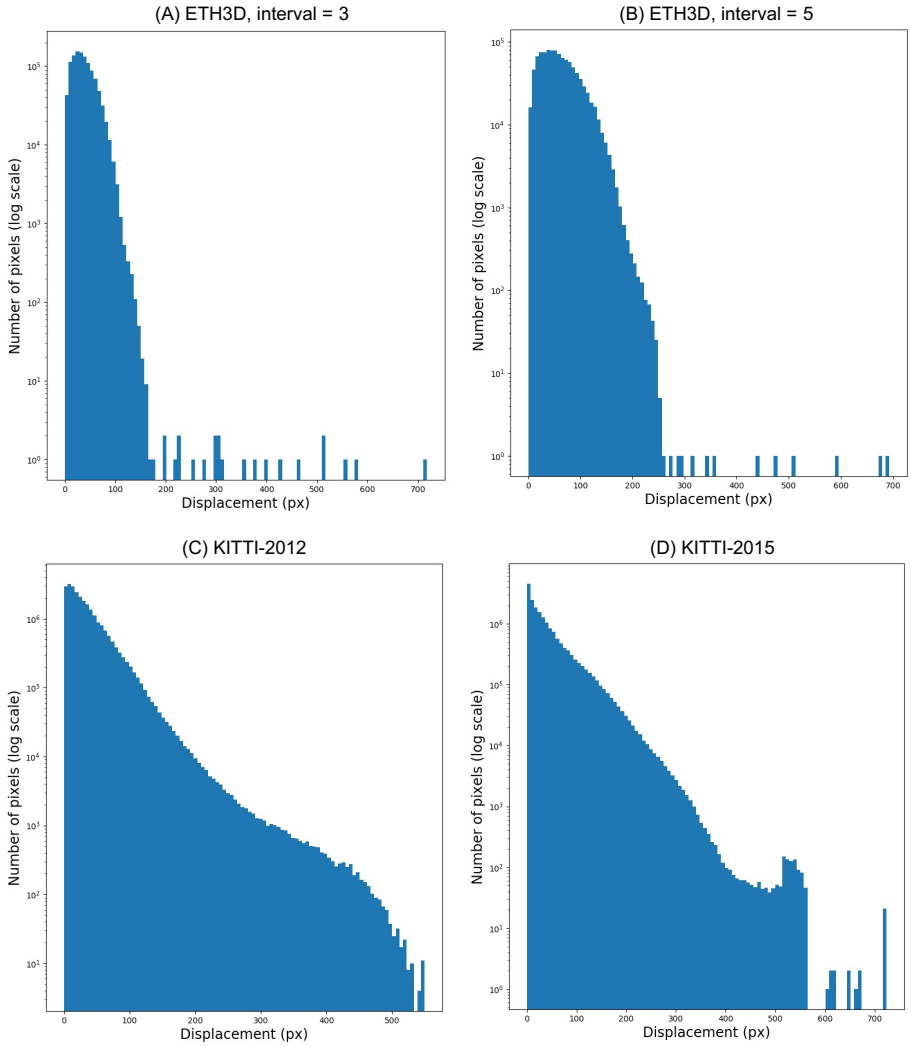


Figure 9. Ground-truth flow distribution (in log scale) for the ETH3D dataset sampled at small intervals and for the KITTI datasets.

	KITTI-2012			KITTI-2015		
	Small	Medium	Large	Small	Medium	Large
PWC-Net	0.63	1.58	10.36	0.94	2.89	28.65
LiteFlowNet	**0.46**	**1.24**	10.83	**0.68**	**2.32**	29.93
DGC-Net	1.53	3.10	21.90	3.44	6.48	36.07
GLU-Net	0.83	1.63	**7.68**	2.25	4.87	**23.01**

Table 5. AEPE for different ground truth pixel-displacement categories on the KITTI datasets. Small is defined as $\|\mathbf{w}_{GT}\|_2 < 10$, Medium as $10 \leq \|\mathbf{w}_{GT}\|_2 < 40$ and Large as $40 \leq \|\mathbf{w}_{GT}\|_2$. The EPE is averaged over all pixels of the dataset.

substantially better on the large-displacement pixels of KITTI compared to PWC-Net and LiteFlowNet, as evidenced in Table 5. This explains the *on-average* advantage of our approach (better AEPE), despite being slightly weaker for small displacements.

3.6. Detailed ablative analysis

In this section, we provide additional ablation experiments. All networks are trained on *CityScape-DPED-ADE* dataset.

Coarse-to-fine-approach: We first defend the use of a coarse-to-fine approach with a feature pyramid. We report AEPE and PCK metrics for the flow estimates obtained at different levels of the feature pyramid of GLU-Net model in Table 6. On the

	AEPE	PCK-1px [%]	PCK-5px [%]
Level 1 [16×16]	45.49	0.70	13.53
Level 2 [32×32]	30.00	6.27	50.29
Level 3 [$H/8 \times W/8$]	26.43	30.47	74.44
Level 4 [$H/4 \times W/4$]	25.05	39.55	78.54

Table 6. Effect of coarse-to-fine approach for our GLU-Net: Metrics calculated over HP images. The flow estimated at each pyramid level is up-sampled to original image resolution and the metrics are calculated at this resolution.

		GLOCAL-Net	1L = 1 H-Net level	2L = 2 H-Net levels	3L = 3 H-Net levels
HP-240	AEPE	8.77	**7.47**	7.69	8.93
	PCK-1px [%]	48.53	**62.85**	53.83	35.81
	PCK-5px [%]	78.12	**85.32**	83.17	75.97
HP	AEPE	31.64	**24.75**	25.55	32.03
	PCK-1px [%]	10.23	33.92	**35.26**	28.76
	PCK-5px [%]	56.73	**76.99**	75.79	69.78
TSS	PCK [%]	77.29	62.98	**78.97**	69.78

Table 7. Effect of adaptive resolution and its position. All networks are without iterative refinement and without cyclic consistency. 2 H-Net levels (**2L**) is the only alternative for a universal network applicable to geometric matching, semantic correspondence and optical flow.

flow field estimated at each level, we apply bilinear interpolation to the original image resolution and multiply the estimated flow with the corresponding scale factor for the levels of L-Net. The end-point error decreases from the coarsest level to the highest level of the pyramid while the accuracy (PCK) increases. This supports the use of a pyramidal model.

Scale pyramid level of the adaptive resolution: In Table 7, we present the influence of the pyramid level at which the adaptive resolution module is integrated in the four-level pyramid network. Having a single level in L-Net (corresponding to the global correlation layer) and three pyramid levels in H-Net (referred to as **3L**) lead to poor results on all datasets, even compared to GLOCAL-Net. On the other hand, both other alternatives (1 or 2 levels in H-Net) bring about major improvements of robustness (AEPE) and accuracy (PCK) on HPatches dataset, particularly on the high-resolution images HP. However, having only one level in H-Net (**1L**) degrades the performances obtained on the semantic dataset TSS. H-Net and L-Net both comprised of 2 pyramid levels (**2L**) appears as the best option to achieve competitive results on geometric matching, optical flow as well as semantic matching.

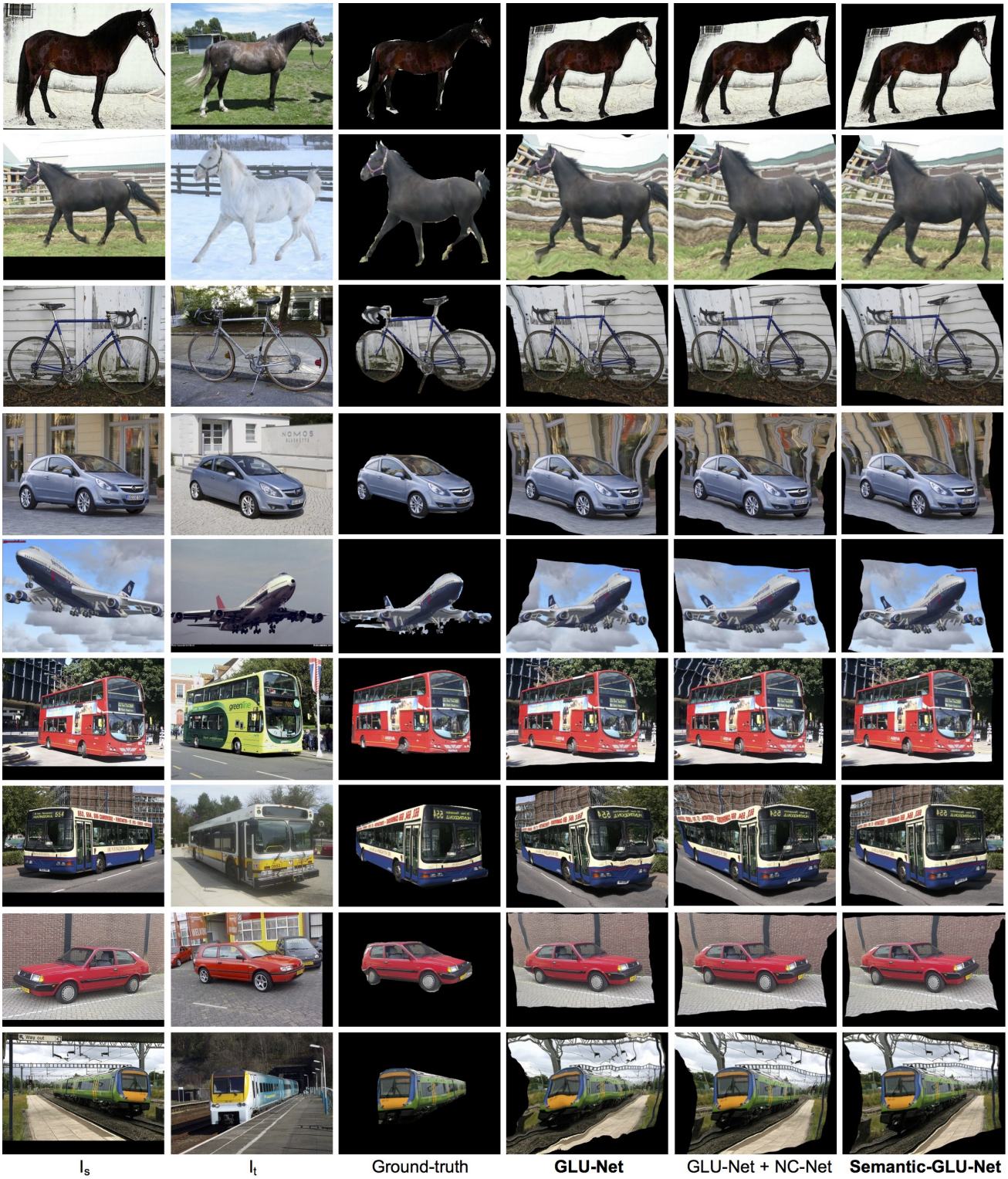


Figure 10. Qualitative examples of our universal network GLU-Net as well as GLU-Net with specific architectural details from the semantic correspondence literature applied to TSS images. The additional architectural modules are the Neighborhood Consensus Network NC-Net [17] and concatenating features within the L-Net [9]. Adopting those two modules leads to Semantic-GLU-Net. The source images are warped according to the flow fields outputted by the different networks. The warped source images should resemble the target images and the ground-truths.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3852–3861, 2017. 11

[2] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 2

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[4] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2758–2766, 2015. 4

[5] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017. 2

[6] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8981–8989, 2018. 6

[7] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3297–3305, 2017. 4

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015. 1

[9] Sangryul Jeon, Seungryong Kim, Dongbo Min, and Kwanghoon Sohn. PARN: pyramidal affine regression networks for dense semantic correspondence. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI*, pages 355–371, 2018. 11, 15

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 4

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012. 4

[12] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, 126(9):942–960, 2018. 5

[13] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. DGC-Net: Dense geometric correspondence network. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 1, 2, 4, 6, 7

[14] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814, 2010. 1

[15] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 4

[16] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 39–48, 2017. 1

[17] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelovic, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 1658–1669, 2018. 1, 11, 15

[18] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2538–2547, 2017. 6

[19] René Schuster, Oliver Wasenmüller, Christian Unger, and Didier Stricker. An empirical evaluation study on the training of SDC features for dense pixel matching. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, page 0, 2019. 5

[20] Rainer Sprengel, Karl Rohr, and H. Siegfried Stiehl. Thin-plate spline approximation for image registration. In *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 1996. 2

[21] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8934–8943, 2018. 2, 4, 6

[22] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of cnns for optical flow estimation. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 5

[23] Tatsunori Taniai, Sudipta N. Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4246–4255, 2016. 11

[24] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *Int. J. Comput. Vis.*, 127(3):302–321, 2019. 4