Deep Matching Prior: Test-Time Optimization for Dense Correspondence - Supplementary Materials -

In this document, we provide more details of DMP and more results on the Hpatches [1], ETH3D [12], TSS [15], and PF-PASCAL [4].

1. Network Architecture

As shown in Fig. 1, our network consists of two parts: *feature extraction* networks to extract deep features and *matching* networks. Note that in the paper, a single-level version of the networks are illustrated for brevity, while the full model is formulated in a pyramidal fashion. In this section, we will explain our full model.

Feature extraction network. Here, we explain feature extraction networks in detail. We employ an adaptive resolution startegy introduced by [16] to let the network take any resolution, which we down-sample the original input images to 256×256. We then extract features from both the original and down-sampled resolutions using pre-trained backbone networks and freeze them during the optimization and training. After the feature extraction, we additionally use an adaptation layer to refine the features. Adaptation layers are random initialized and separated for each pyramidal feature map in a residual fashion [5]. As described in the paper, the backbone model could be directly optimized in DMP framework, but we found that using an additional adaptation layer to refine the backbone features and optimizing the layer only boosts the performance drastically.

Specifically, we compute the residuals by adding 3×3, 5×5, 7×7 and 9×9 convolutional layers with padding of 1, 2, 3 and 4, respectively, on top of each pyramidal level. We set stride to 1 to ensure that the spatial resolution is preserved. Given VGG-16 as backbone network, identical to [16], we employ the activation after `Conv5-3` and `Conv4-3` for the resized (256×256) input images, and `Conv4-3` and `Conv3-3` for the original resolution image, which outputs spatial resolution of 16×16, 32×32, $\frac{H}{8} \times \frac{W}{8}$ and $\frac{H}{4} \times \frac{W}{4}$, respectively. The number of feature channels of each adaptaion layer are thus 512, 256, 256, and 128, respectively. On the contrary, if ResNet-101 is used as backbone network, we employ the activation after `Conv3` and `Conv4` for the resized input, whereas for the original resolution we employ `Conv2` and `Conv3`. The number of feature channels of each adaptaion layer are thus 1024, 512, 512, and 256, respectively. It should be noted that for geometric matching task, we use VGG features, while for semantic matching task, we use both ResNet and VGG, which by default, unless mentioned, all our models use VGG-16 features.

Matching network. We then provide additional details of matching networks, which consists of two parts: cost computation and inference modules. For the global correlation, we compute the pairwise inner product between features from coarsest level. For the local correlation, we employ $l = 4$ for the search space in the target. As in [9, 16] we feed global correlation into a inference module, which consists of 5 feed-forward convolutional blocks with a 3×3 filter. The number of output channels of each layers are 128, 128, 96, 64, and 32, respectively. For the remaining levels, we use an inference module designed for the local correlation volume which infers the flow field similar to the one in PWC-Net [14]. The numbers of output channels at each layer are 128, 128, 96, 64, and 32, respectively and the size spatial kernel of is also 3×3. The final output of the inference module is computed by feeding into a linear 2D convolution. The soft-argmax [6] computes an output by averaging all the spatial positions with weighted corresponding probabilities. The temperature for the soft-argmax is set to 0.02.

The flow field inferred at each level is up-sampled using bilinear interpolation. From experiments, we observed that using transposed convolution degraded the performance. We thus employed bilinear interpolation at every pyramidal layer.

2. Convergence Analysis Details

In the paper, we showed the comparison of AEE over iteration between models as shown in Fig. 2 (Fig. 4 in the paper). Here, we describe the details for this experiment. For a fair comparison, we iterated 2k times for all the test-time optimization methods, which include GLU-Net‡, DMP, A-DMP and DMP†. We evaluated each method on Hpatches [1] benchmark, which consists of 295 target images, and averaged the AEE at every 10-th iteration. Note

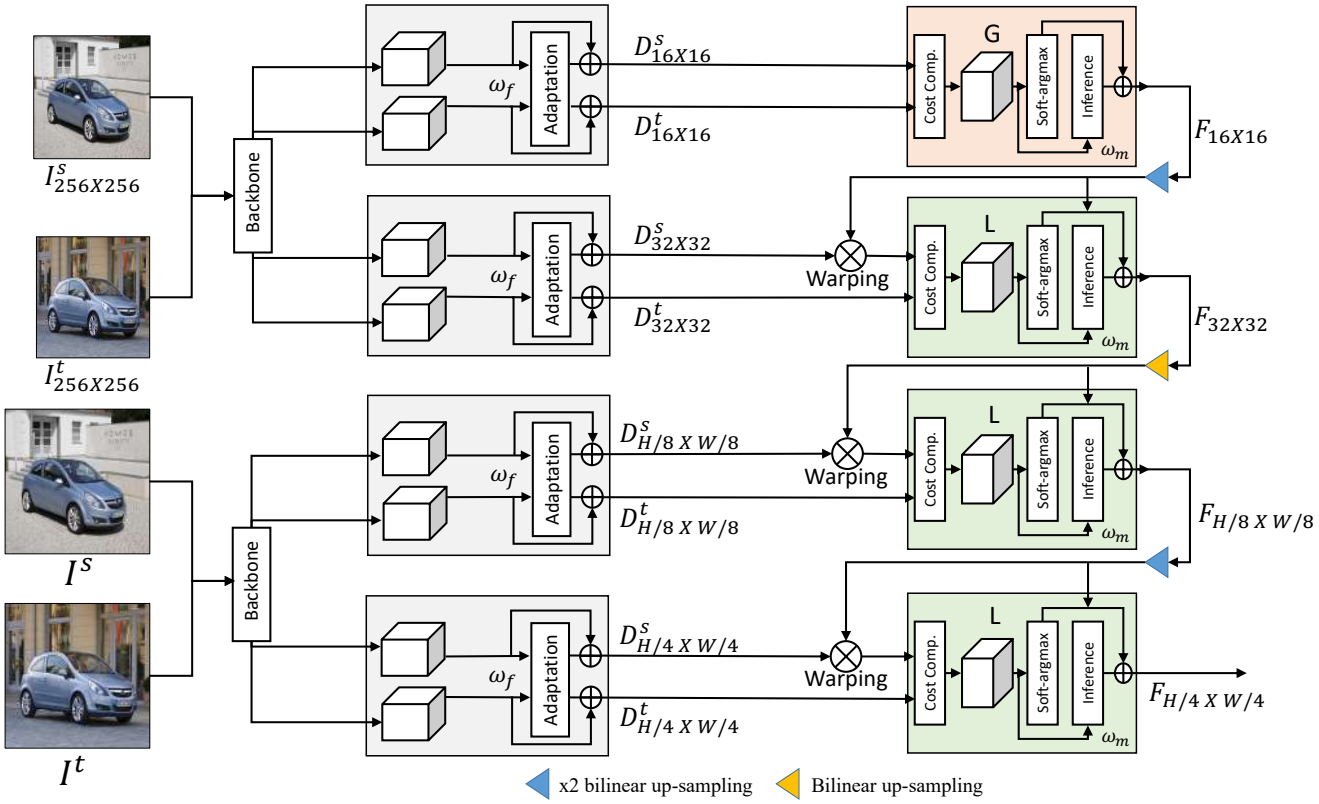


Figure 1. **Overview of DMP architecture.** Overview of our proposed iterative architecture, which consists of feature extraction network and matching network. Source and target images are first fed into feature backbone network to obtain deep features. Each pyramidal features are then fed into adaptation layers and the refined features are obtained. Subsequently, the refined features are fed into a matching network and the estimated flow is up-sampled to warp the next level feature. The final output consists of refined features from target image and the flow field of size $\frac{H}{4} \times \frac{W}{4}$.

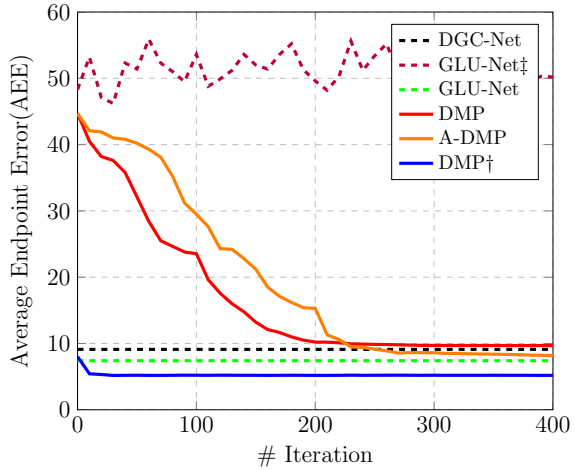


Figure 2. **Convergence analysis of DMP.**

that in Fig. 2, we only show the range of 0-400 for x-axis as the AEE for all the methods except GLU-Net‡ converge. To conduct experiment on GLU-Net‡, we simply replaced the model to GLU-Net within our optimization implementation under the identical experimental setting to DMP test-time optimization. We did not find noticeable differences when we attempted optimizing with different hyperparameter settings e.g., learning rate. We conducted experiment on GLU-Net‡ to show that the several choices we made, including architecture and loss, were critical for the untrained network to guarantee a meaningful convergence.

3. Limitations

In this session, we would like to discuss the limitations of DMP and its variants. One limitation that all the methods, including DMP and its variants, is the time they take to converge. Although with good initialization, the optimization time required to obtain correspondences significantly reduces, our approach fundamentally isn't applicable for real-time applications. Furthermore, even though DMP attained competitive results for standard benchmarks by optimizing from untrained networks, it fails to find accurate correspondences given difficult images, i.e., ETH3D interval 15. To overcome, we pre-trained DMP to provide strong initialization, but this may result in weakening of DMP's advantage,

(a) Original (b) Affine (c) Homography (d) TPS

Figure 3. **Example of the synthetic images** [11].

an ability to avoid generalization issues. Although designed to address difficult cases, A-DMP suffers from doubled optimization time. RANSAC-DMP successfully avoids this challenge, but the use of RANSAC often yields unstable results that may lead to failure to find correspondences.

We proposed, for the first time, to find correspondences between a pair of images by test-time optimization, and we believe that further improvements could be made in this direction.

4. Implementation and Experimental Details

We first pre-process the input images by centering the mean and normalizing the values using the mean and standard deviation of ImageNet [2]. For DMP and A-DMP, we initially set the learning rate to $3e^{-3}$ and divide it by 2 at every 300 iterations. For DMP† and variants that exploit RANSAC [13], we use learning rate of $1e^{-5}$. We use Adam optimizer [7] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We implemented our model using `PyTorch`[10].

To obtain the randomly-augmented target for A-DMP, we use the same kind of geometric transformation to GLU-Net and DGC-Net. Specifically, Rocco et al. [11] generates synthetic data using affine and thin-plate spline transformation which we additionally use homography transformation as in DGC-Net as shown in Fig. 3. To conduct experiments on variants of ours that utilize RANSAC to obtain coarsely aligned pair of images, we followed the protocol of [13] to obtain a pair of coarsely aligned images first and then fed the aligned images into our network.

We additionally showed an ablation study on RANSAC-Flow [13] in the paper, to validate the effect of test-time optimization. We first obtained coarsely aligned input images and then implemented using the full loss function provided in [13] for the test-time optimization and iterated 2000 times with identical hyperparameter setting to RANSAC-Flow trained on Mega-Depth [8]. We did not find drastic diffference when the matchability loss was not included within the total loss. For evaluating test-time optimization of RANSAC-Flow on original resolution of Hpatches, we up-sampled the estimated flow using bilinear interpolation and calculated the AEE and PCK.

5. More Results

In this section, we provide additional qualitative examples on the Hpatches [1], ETH3D [12], TSS [15], and PF-PASCAL [4].

We first show more qualitative results of convergence process of DMP. Given good initialization, DMP guarantees a meaningful convergence, which also indicates that once the warped image is similar enough to the target image, the convergence process is boosted and DMP can successfully correct the errors in the flow fields during the optimization to find the optimal flow field. As shown in Fig. 4, the convergence is boosted when the warped image is similar to the target image.

For geometric matching task, DMP shows highly competitive results , nearly approximating the ground-truth flow as shown in Fig. 5. Note that our variants estimate extremely accurate flow fields, demonstrating the superiority of our approach. Also we deliberately included the examples that DMP fails to find accurate correspondences while other variants do better. Fig. 6 shows the qualitative comparison on ETH3D [12] dataset. All the results are from the highest intervals, which demand addressing extreme viewpoint changes. Note that our approaches, compared to GLU-Net [16] which obtains satisfactory results, successfully estimate the correspondence field between images with extreme appearance variations.

Semantic matching task requires estimating correspondence fields between images with intra-class variations. Our works, compared to GLU-Net [16], consistently obtain sharp and extremely accurate warped images as shown in Fig. 7 and Fig. 8. We obtain results with fine details preserved and accurately aligned, which demonstrate the superiority of our approaches on semantic matching task.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 1, 3, 6

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3

[3] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 6

[4] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. In *CVPR*, 2016. 1, 3, 9

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[6] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry.

End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017. 1

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 3

[8] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 3

[9] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. Dgc-net: Dense geometric correspondence network. In *WACV*, 2019. 1

[10] 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. 2017. 3

[11] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, 2017. 3

[12] Thomas Schops, Johannes L Schonberger, 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 *CVPR*, 2017. 1, 3, 7

[13] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. Ransac-flow: generic two-stage image alignment. *arXiv:2004.01526*, 2020. 3

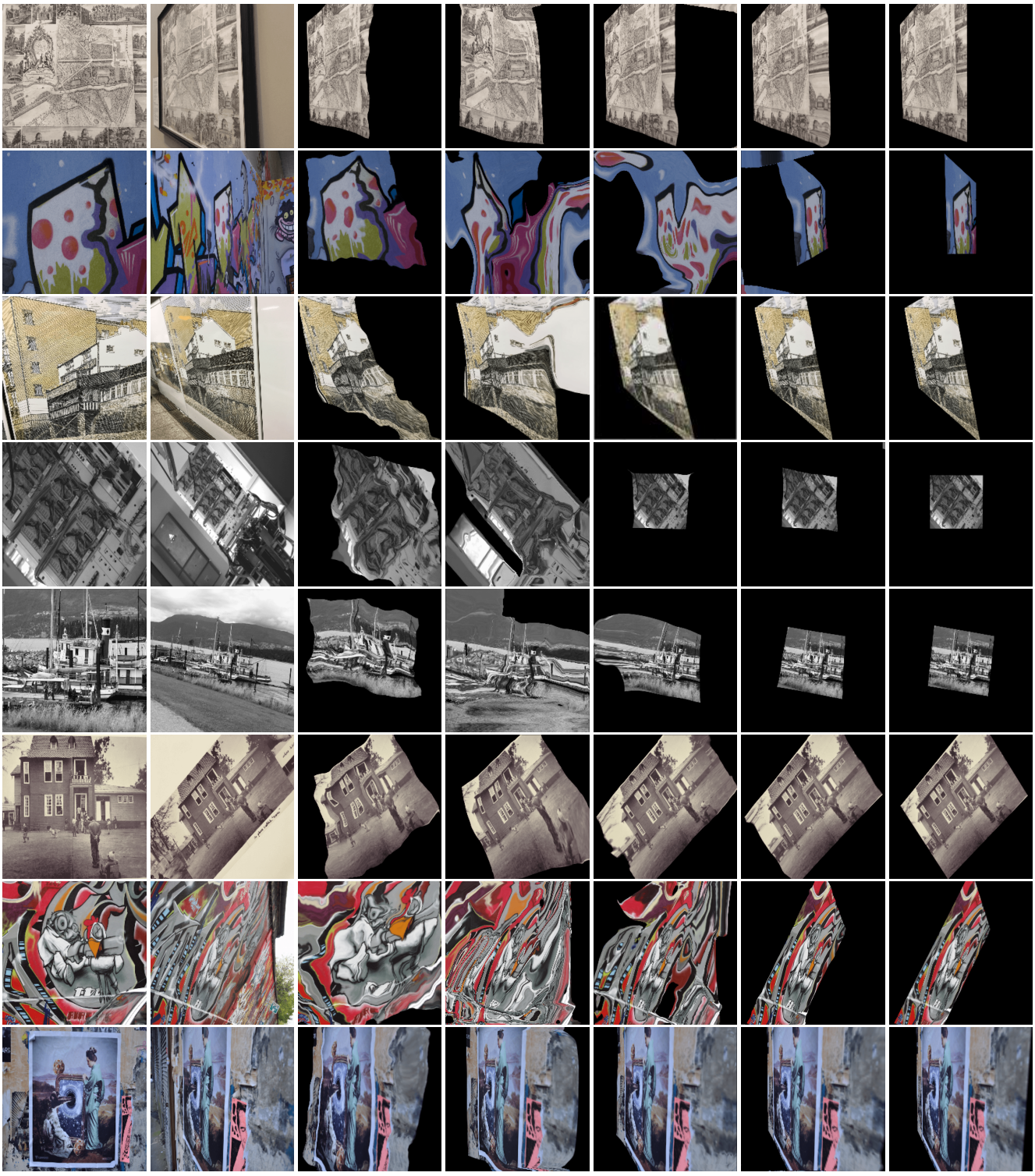
[14] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 1

[15] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR*, 2016. 1, 3, 8

[16] Prune Truong, Martin Danelljan, and Radu Timofte. Glu-net: Global-local universal network for dense flow and correspondences. In *CVPR*, 2020. 1, 3, 6, 7, 8

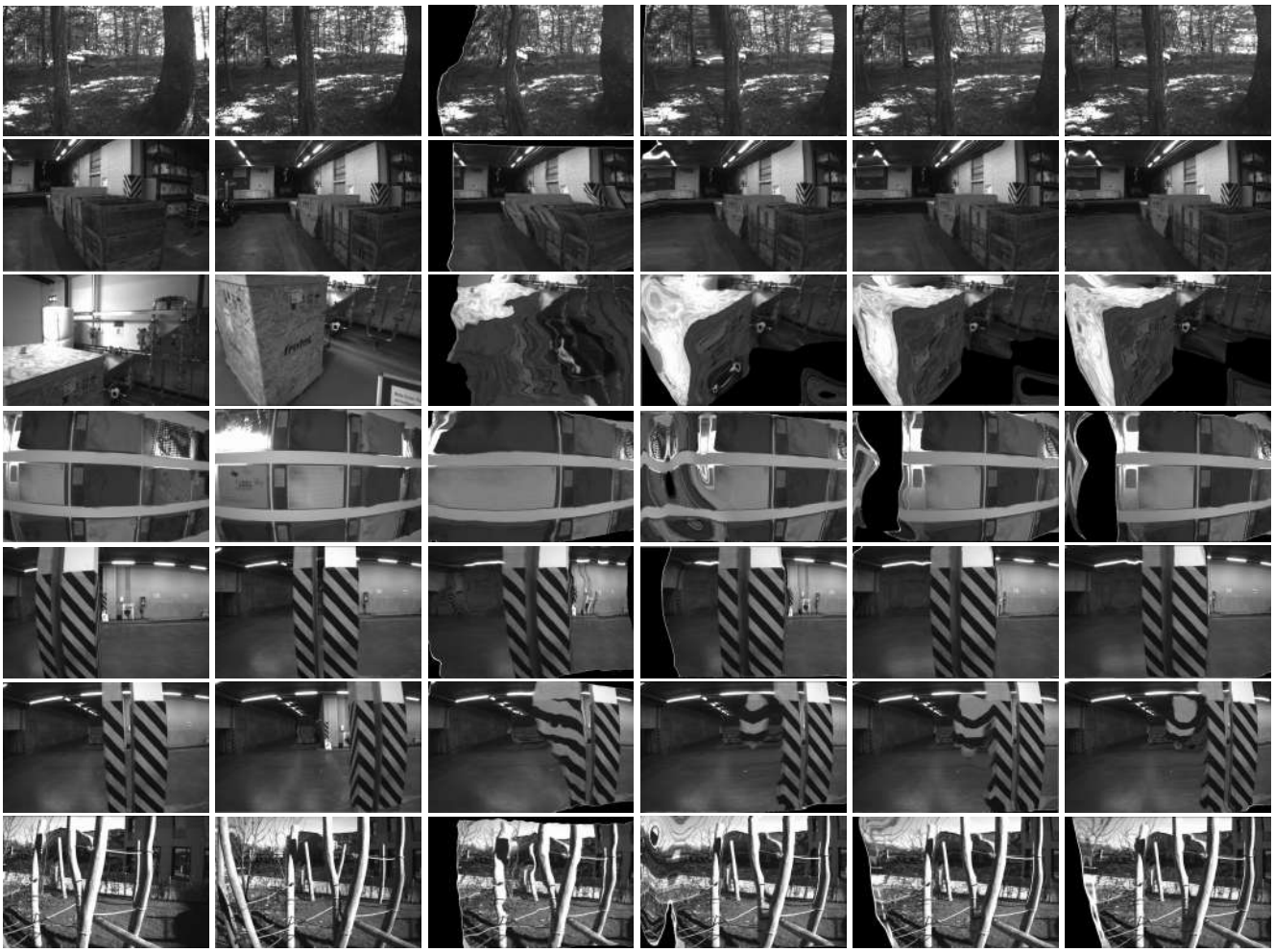
(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4. **Convergence of DMP.** (a) source image, (b) target image, (c), (d), (e), (f), (g), and (h) iterative evolution of warped images by DMP. The error signal received at each iteration helps to correct the flow field, which the predicted transformation fields become progressively more accurate through iterative estimation.



(a) (b) (c) (d) (e) (f) (g)

Figure 5. **Qualitative results on the Hpatches benchmark [1]**. (a) source and (b) target images, warped source images using correspondences of (c) GLU-Net [16], (d) DMP, (e) DMP†, (f) RANSAC-DMP, and (g) Ground-truth. Here, we provide only the samples with extremely large geometric variations to compare the outputs produced by each variants and GLU-Net. Note that DMP, starting from untrained network, achieves competitive results against GLU-Net trained on a large-scale dataset. Thanks to RANSAC [3], DMP starts the optimization with good initialization, which results RANSAC-DMP producing highly accurate flow fields.

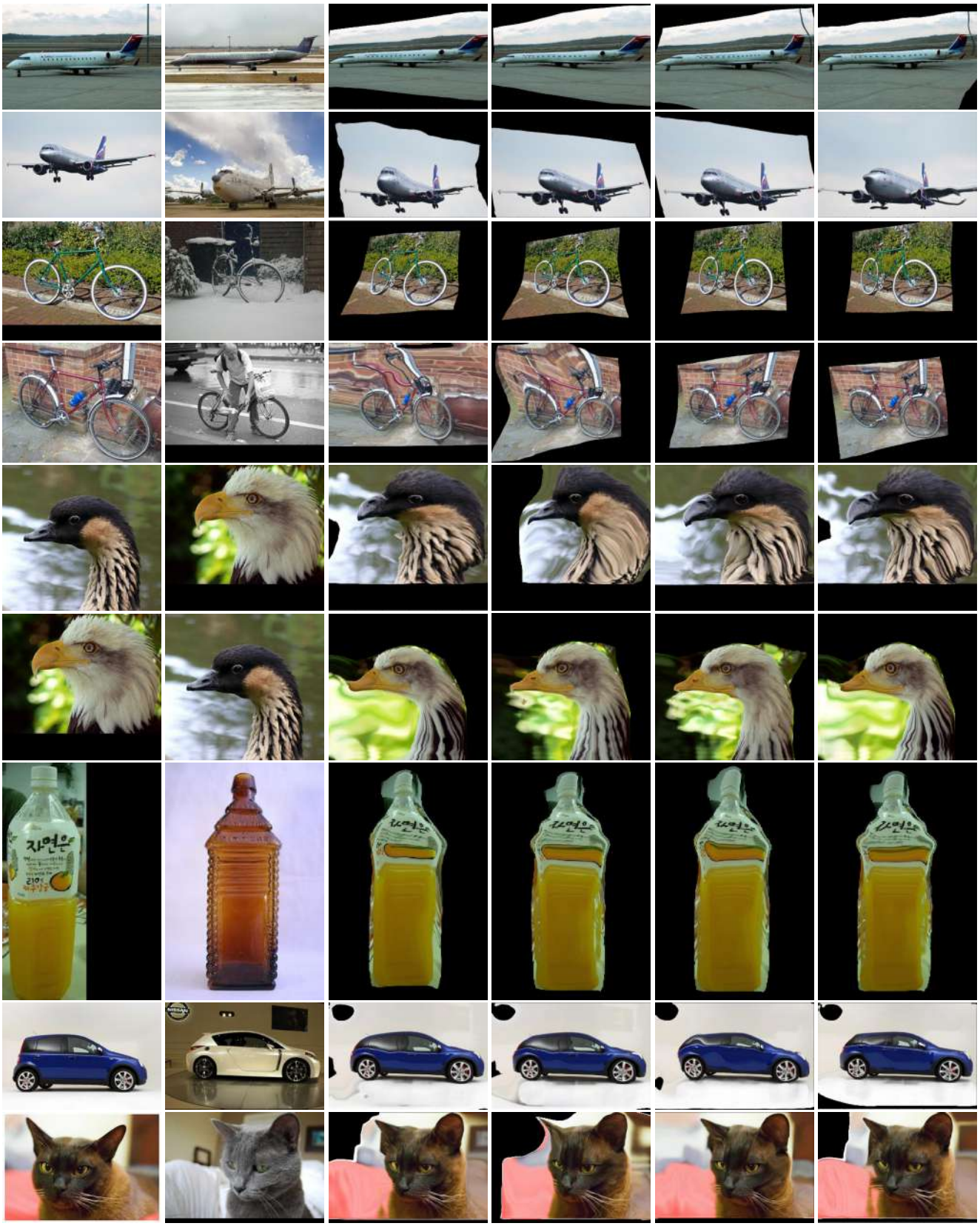


(a) (b) (c) (d) (e) (f)

Figure 6. Qualitative results on the ETH3D benchmark [12]: (a) source and (b) target images, warped source images using correspondences of (c) GLU-Net [16], (d) DMP, (e) A-DMP, and (f) DMP† . Note that our loss function allows error correction, allowing more optimal estimation of flow fields.

(a) (b) (c) (d) (e) (f)

Figure 7. **Qualitative results on the TSS [15] benchmarks**. (a) source image, (b) target image, (c) ground-truth, (d) GLU-Net [16], (e) DMP, and (f) DMP†-ResN. It is clearly visible that warped source images produced by our models resemble the target images. Note that more accurate flow fields are estimated when ResNet is used for the feature backbone network.



(a) (b) (c) (d) (e) (f)

Figure 8. **Qualitative results on the PF-PASCAL [4] benchmarks**. (a) source image, (b) target image, (c) DMP, (d) A-DMP, (e) DMP†, and (f) DMP†-ResN.