Supplementary Material: Hierarchical Metric Learning and Matching for 2D and 3D Geometric Correspondences*†

Mohammed E. Fathy[1], Quoc-Huy Tran[2], M. Zeeshan Zia[3], Paul Vernaza[2], and Manmohan Chandraker[2,4]

[1]Google Cloud AI
[2]NEC Laboratories America, Inc.
[3]Microsoft Hololens
[4]University of California, San Diego

This document is organized as follows. In Section 1, we provide architectural details of our GoogLeNet variant and the *topdown-fusion* baseline, as well as summarize architecture differences between the original VGG-M [3] and GoogLeNet [7] networks and our VGG-M and GoogLeNet variants respectively. Next, Section 2 discusses generalization results when training on MPI Sintel [2] and HPatches [1] and testing on KITTI Flow 2015 [5]. Sections 3 and 4 present results of ablation study with two levels and experiment with more than two levels respectively. Finally, we show additional results on 3D correspondence estimation with 90x90x90 cm^3 search volumes in Section 5.

1 Network Architectures

We present the network architectures of our GoogLeNet variant and the *topdown-fusion* baseline in Figures S1 and S2 respectively. Our GoogLeNet variant is described in Section 3.1 of the main paper, whereas the *topdown-fusion* baseline is inspired by ideas from [6] for fusing features from different layers in a top-down scheme and used in our comparisons in Section 4.1 of the main paper.

In addition, we summarize the architectural changes that we introduced to the original VGG-M [3] and GoogLeNet [7] networks to obtain the VGG-M and GoogLeNet variants of our approach in Tables S1 and S2 respectively.

*Part of this work was done during M. E. Fathy's internship at NEC Labs America. Code and models will be made available at http://www.nec-labs.com/∼mas/HiLM/.

†The authors thank C. B. Choy and A. Zeng for their help with the code of UCN and 3DMatch respectively.

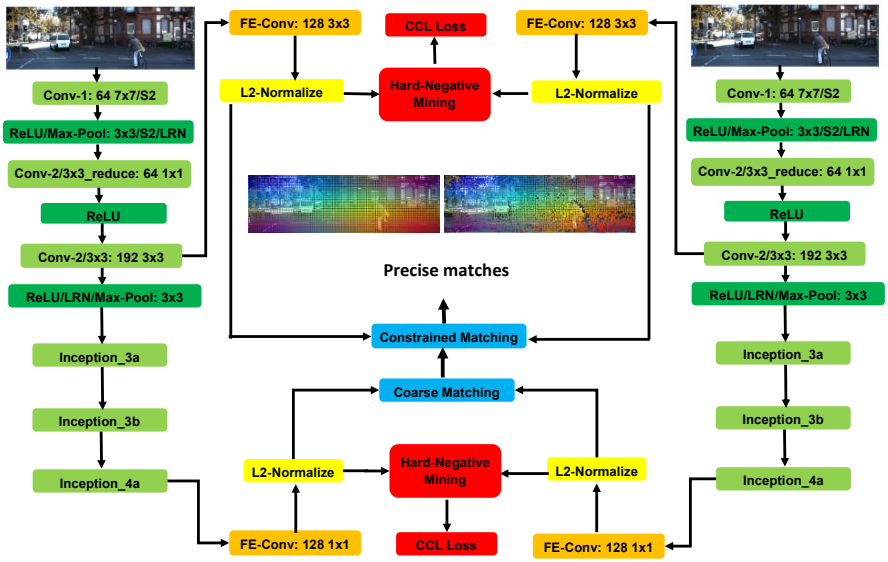


Fig. S1: One instantiation of our proposed ideas using the GoogLeNet [7] baseline. Note that the hard negative mining and CCL losses (red blocks) are relevant for training, and matching (blue blocks) for testing. Convolutional blocks (green) in the left and right Siamese branches share weights. *'S'* and *'D'* denote striding and dilation offsets.

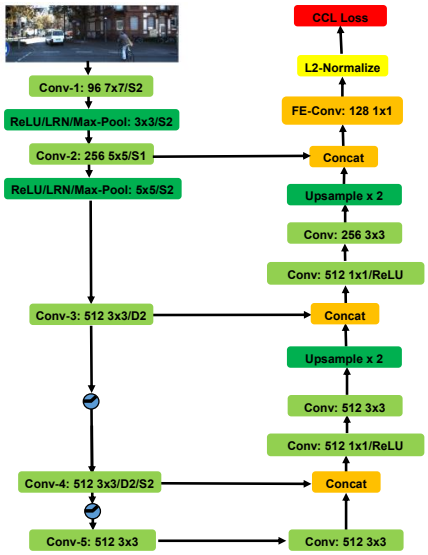


Fig. S2: One Siamese branch of the *topdown-fusion* baseline, inspired by ideas from [6], in our evaluation. *'S'* and *'D'* denote striding and dilation offsets.

Table S1: Architectural differences between the original VGG-M [3] network and the VGG-M variant of our approach (highlighted in **bold**). Note that layers after *'Conv5'* in the original network are not included in ours. *'S'* and *'D'* denote striding and dilation offsets.

Layer	VGG-M (Original)	HiLM (VGG-M) (Ours)
Conv1	Pad 0, 96 7x7/S2, ReLU, LRN	**Pad 3**, 96 7x7/S2, ReLU, LRN
Conv1-MaxPool	Pad 1, 3x3/S2	Pad 1, 3x3/S2
Conv2-Lin	Pad 1, 256 5x5/S2	**Pad 2**, 256 5x5/**S1**
Conv2-NonLin	ReLU, LRN	ReLU, LRN
Conv2-MaxPool	Pad 0, 3x3/S2	**Pad 2**, **5x5/S1**
Conv3	Pad 1, 512 3x3/S1, ReLU	**Pad 4**, 512 3x3/**D4**/S1, ReLU
Conv4	Pad 1, 512 3x3/S1, ReLU	**Pad 4**, 512 3x3/**D4**/S1, ReLU
Conv5	Pad 1, 512 3x3/S1, ReLU	**Pad 5**, 512 3x3/**D4**/S1
FE-Conv2 (input: Conv2-Lin)	N/A	**Pad 1, 128 3x3/S1**
FE-Conv5 (input: Conv5)	N/A	**Pad 0, 128 1x1/S1**

Table S2: Architectural differences between the original GoogLeNet [3] network and the GoogLeNet variant of our approach (highlighted in **bold**). Note that layers after *'Inception-4a'* in the original network are not included in ours. *'S'* denotes stride.

Layer	GoogLeNet (Original)	HiLM (GoogLeNet) (Ours)
Conv1	Pad 3, 64 7x7/S2, ReLU	Pad 3, 96 7x7/S2, ReLU, LRN
Conv1-MaxPool	Pad 0, 3x3/S2, LRN	Pad 1, 3x3/S2
Conv2/3x3_r	Pad 0, 64 1x1/S1, ReLU	Pad 0, 64 1x1/S1, ReLU
Conv2/3x3-Lin	Pad 1, 192 3x3/S1	Pad 1, 192 3x3/S1
Conv2/3x3	ReLU, LRN	ReLU, LRN
Conv2-MaxPool	Pad 0, 3x3/S2	**Pad 1**, 3x3/**S1**
Inception-3a	Same	Same
Inception-3b	Same	Same
MaxPool-3b	Pad 0, 3x3/S2	**Pad 1**, 3x3/**S1**
Inception-4a	Same	Same (final ReLU removed)
FE-Conv2 (input: Conv2/3x3-Lin)	N/A	**Pad 1, 128 3x3/S1**
FE-4a (input: Inception-4a)	N/A	**Pad 0, 128 1x1/S1**

2 Generalization Results

As mentioned in Section 4.1 of the main paper, we perform experiments to evaluate the generalization ability of our approach by training and testing on different scenes. In the following, we first provide the details of network training on MPI Sintel [2] and HPatches [1], and then present the results of our networks when testing on KITTI Flow 2015 [5]. Note that the evaluation is conducted on the same test set from KITTI Flow 2015 that was used in Section 4.1 of the main paper, and there is no fine tuning on KITTI Flow 2015.

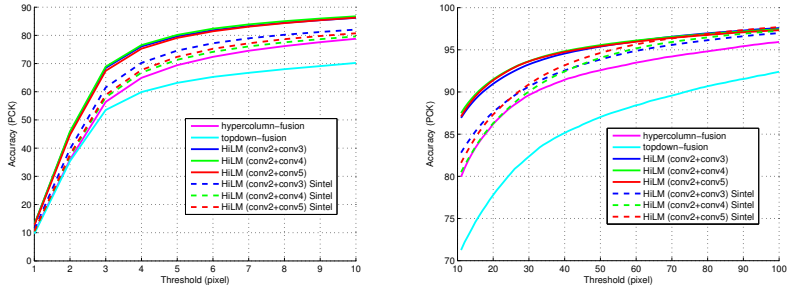
Network Training on MPI Sintel. The MPI Sintel [2] dataset contains synthetic image pairs with ground truth optical flows. We train different variants of our proposed approach on MPI Sintel following the procedure that we use for training on KITTI Flow 2015 described in Section 4.1 of the main paper.

Network Training on HPatches. We also train our networks on the full image set of the HPatches [1] dataset, which consists of 116 sequences with 57 sequences having illumination variations and 59 sequences having geometric (projective) variations. Each sequence has 6 images and 5 ground truth homography transformations between the first image and i-th image for $i = 2, 3, \ldots, 6$. To train our networks, we use all 116 sequences, each with 5 image pairs $(1, i)$ for $i = 2, 3, \ldots, 6$. An input image pair is preprocessed for training by randomly cropping a 512×380 subimage from the reference image and using the ground truth homography to help compute the coordinates of the corresponding 512×380 cropped region in the target image. We use these 512×380 cropped regions as input for our networks and randomly select 1K correspondending points between the cropped images for training. The training is run for 50K iterations and we use a batch of 4 randomly chosen image pairs per iteration.

Generalization Results of HiLM. We show in Figure S3 the generalization results of different variants of our proposed approach when training on MPI Sintel [2] and testing on KITTI Flow 2015[5]. Our Sintel-trained models outperform both feature fusion baselines, namely *hypercolumn-fusion* [4] and *topdown-fusion* [6], that were trained on KITTI, across all PCK thresholds, *e.g.* HiLM (*conv2+conv3*) Sintel, HiLM (*conv2+conv4*) Sintel, HiLM (*conv2+conv5*) Sintel with 74.60%, 71.35%, 72.37% PCK respectively, versus, *hypercolumn-fusion*, *topdown-fusion* with 69.41%, 63.14% respectively, @ 5 pixels. However, as expected, our networks trained on Sintel do not perform as well as the same networks trained on KITTI, *e.g.* HiLM (*conv2+conv3*), HiLM (*conv2+conv4*), HiLM (*conv2+conv5*) with 79.67%, 80.17%, 79.11% respectively, @ 5 pixels.

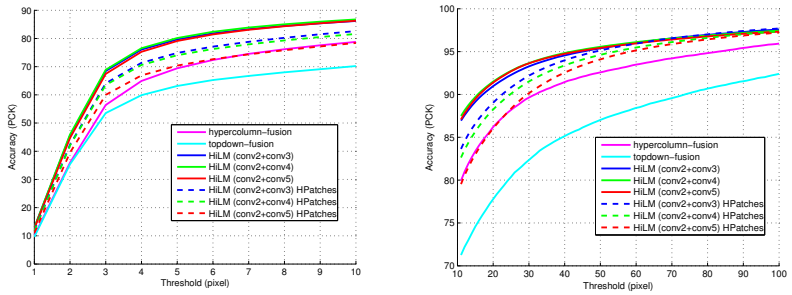
Similar generalization results are also presented in Figure S4, when we train our models on HPatches [1] and test them on KITTI Flow 2015 [5], further demonstrating our cross-domain generalization ability.

Generalization Results of *conv2-net*. It is worth noting that, for a wide range of pixel thresholds, the performance of HiLM (trained on MPI Sintel [2]) is better than the performance of most single layer and feature fusion baselines even trained on KITTI Flow 2015 [5]. The only exception is the *conv2-net* baseline trained on KITTI Flow 2015. We further evaluate the generalization performance of the *conv2-net* baseline by training versions of it on MPI Sintel [2]



(a) Accuracy over small thresholds **(b)** Accuracy over large thresholds

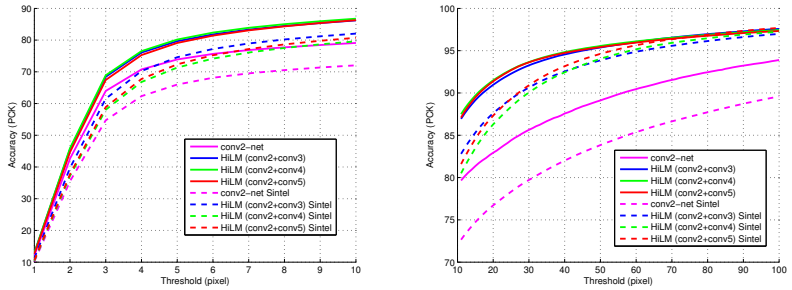
Fig. S3: Generalization results of our approach when training on MPI Sintel [2] and evaluating on KITTI Flow 2015 [5].



(a) Accuracy over small thresholds **(b)** Accuracy over large thresholds

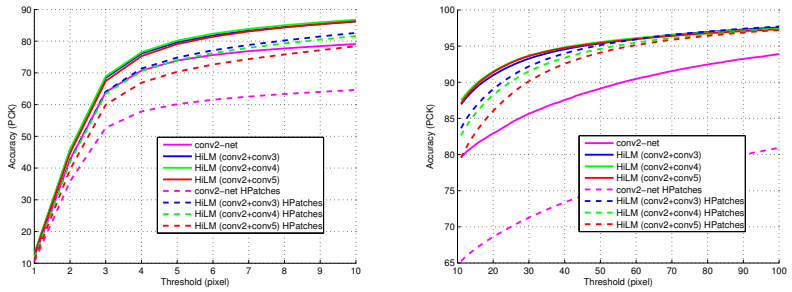
Fig. S4: Generalization results of our approach when training on HPatches [1] and evaluating on KITTI Flow 2015 [5].

and HPatches [1] and testing them on KITTI Flow 2015 [5], plotting the results in Figures S5 and S6. The results show that the generalization performance of *conv2-net* is not on par with HiLM, which indicates the benefit of our hierarchical learning that combines information from multiple levels of the feature hierarchy.



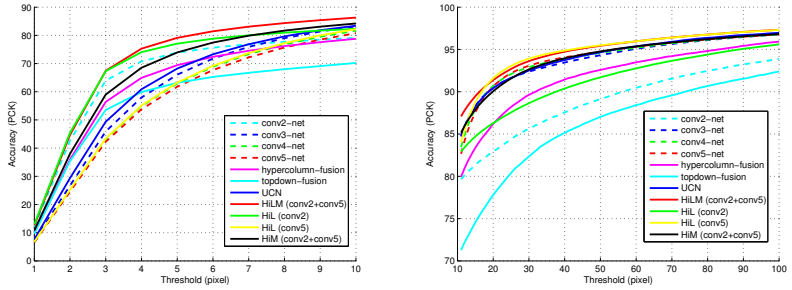
(a) Accuracy over small thresholds **(b)** Accuracy over large thresholds

Fig. S5: Generalization results of the *conv2-net* baseline when training on MPI Sintel [2] and evaluating on KITTI Flow 2015 [5].



(a) Accuracy over small thresholds **(b)** Accuracy over large thresholds

Fig. S6: Generalization results of the *conv2-net* baseline when training on HPatches [1] and evaluating on KITTI Flow 2015 [5].



(a) Accuracy over small thresholds **(b)** Accuracy over large thresholds

Fig. S7: Ablation results on KITTI Flow 2015 [5].

3 Ablation Results

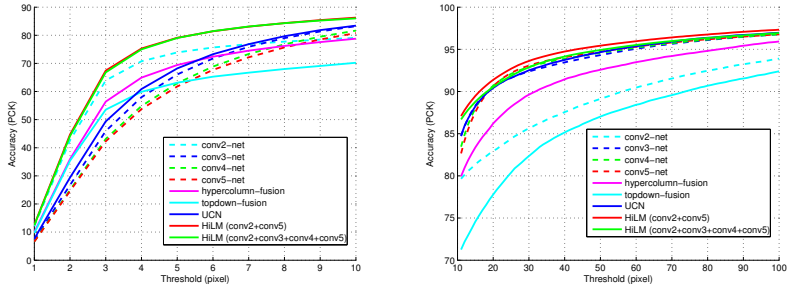
We compare our complete framework HiLM against variants such as HiL, which is trained with hierarchical metric learning but relies either on deep or shallow features for matching (*i.e.*, *conv*5 or *conv*2 features respectively) and HiM, which is not trained with hierarchical metric learning (*i.e.*, it uses contrastive loss based on *conv*5 features only) but employs hierarchical matching. The results in Figure S7 show that our complete framework HiLM outperforms its variants HiL and HiM, *e.g.* HiLM (*conv2+conv5*), HiL (*conv2*), HiM (*conv2+conv5*) with 79.11%, 77.06%, 73.93% PCK respectively, @ 5 pixels. In other words, the best performance is achieved by utilizing both hierarchical metric learning and hierarchical matching.

4 Experiment Results with Four Levels

We combine the best performing deep and shallow feature layers in our two-level approach (*i.e.* HiLM (*conv2+conv5*)). Adding intermediate layers leads to marginal gain as shown in Figure S8, where HiLM with four levels is slightly better than HiLM with two levels, *e.g.* HiLM (*conv2+conv3+conv4+conv5*) with 83.11% PCK versus HiLM (*conv2+conv5*) with 83.08% PCK, @ 7 pixels. Nevertheless, with two layers, we observe significant improvement over previous single-layer and fusion methods, *e.g.* UCN, *hypercolum-fusion*, *topdown-fusion* with 76.83%, 74.54%, 66.70% PCK respectively, @ 7 pixels.

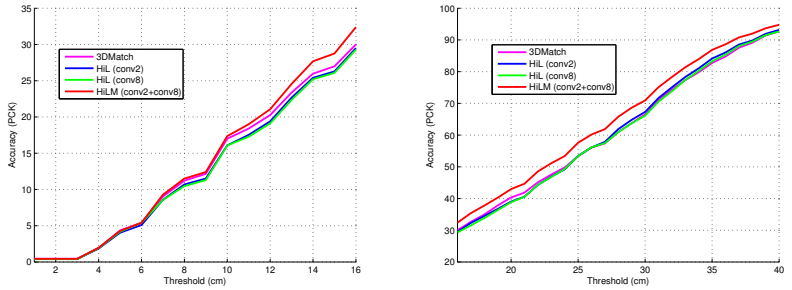
5 3D Correspondence Results

In addition to 3D correspondence estimation results with 60x60x60 cm^3 search volumes presented in Section 4.3 of the main paper, we also conduct similar experiments yet with larger search volumes. In particular, given the descriptor in the reference "image", we search over all candidate keypoints in a 90x90x90 cm^3



(a) Accuracy over small thresholds **(b)** Accuracy over large thresholds

Fig. S8: Experiment results with four levels on KITTI Flow 2015 [5].



(a) Accuracy over small thresholds **(b)** Accuracy over large thresholds

Fig. S9: Accuracy of different CNN-based methods for 3D correspondence search in 90x90x90 cm^3 regions.

region in the target "image" to find the keypoint whose descriptor is most similar to the reference descriptor. Figure S9 presents the results of 90x90x90 cm^3 search regions. Similar observations as with the 60x60x60 cm^3 search volume experiments are obtained. Specifically, our shallow features trained with hierarchical metric loss are usually more effective than their deep counterparts, *e.g.* HiL (*conv2*) with 19.40% versus HiL (*conv8*) with 19.12%, @ 12 cm. In addition, our complete framework outperforms both of its variants, and achieves higher PCK numbers than 3DMatch [8], across all PCK thresholds, *e.g.* HiLM (*conv2+conv8*) with 21.07% versus 3DMatch with 20.27%, @ 12 cm.

References

1. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: CVPR (2017)
2. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: ECCV (2012)
3. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: BMVC (2014)
4. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR (2015)
5. Menze, M., Geiger, A.: Object Scene Flow for Autonomous Vehicles. In: CVPR (2015)
6. Pinheiro, P.O., Lin, T.Y., Collobert, R., Dollár, P.: Learning to refine object segments. In: ECCV (2016)
7. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
8. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In: CVPR (2017)