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

Abstract. Interest point descriptors have fueled progress on almost every problem in computer vision. Recent advances in deep neural networks have enabled task-specific learned descriptors that outperform hand-crafted descriptors on many problems. We demonstrate that commonly used metric learning approaches do not optimally leverage the feature hierarchies learned in a Convolutional Neural Network (CNN), especially when applied to the task of geometric feature matching. While a metric loss applied to the deepest layer of a CNN, is often expected to yield ideal features irrespective of the task, in fact the growing receptive field as well as striding effects cause shallower features to be better at high precision matching tasks. We leverage this insight together with explicit supervision at multiple levels of the feature hierarchy for better regularization, to learn more effective descriptors in the context of geometric matching tasks. Further, we propose to use activation maps at different layers of a CNN, as an effective and principled replacement for the multi-resolution image pyramids often used for matching tasks. We propose concrete CNN architectures employing these ideas, and evaluate them on multiple datasets for 2D and 3D geometric matching as well as optical flow, demonstrating state-of-the-art results and generalization across datasets.

Keywords: Hierarchical metric learning · Hierarchical matching · geometric correspondences · dense correspondences

1 Introduction

The advent of repeatable high curvature point detectors [24, 37, 40] heralded a revolution in computer vision that shifted the emphasis of the field from holistic object models and direct matching of image patches [67], to highly discriminative hand-crafted descriptors. These descriptors made a mark on a wide array of problems in computer vision, with pipelines created to solve tasks such as optical flow [9], object detection [18], 3D reconstruction [51] and action recognition [55].

The current decade is witnessing as wide-ranging a revolution, brought about by the widespread use of deep neural networks. Yet there exist computer vision

*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/.

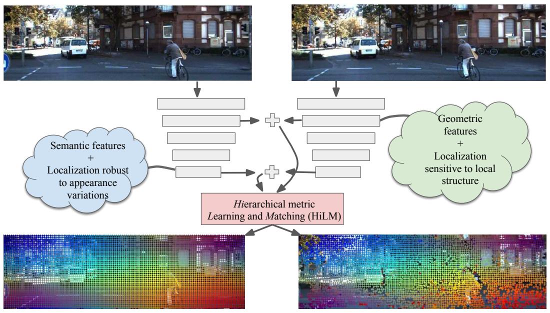

Semantic features
+
Localization robust to appearance variations
Geometric features
+
Localization sensitive to local structure
Hierarchical metric Learning and Matching (HiLM)

Fig. 1: Our hierarchical metric learning retains the best properties of various levels of abstraction in CNN feature representations. For geometric matching, we combine the robustness of deep layers that imbibe greater invariance, with the localization sensitivity of shallow layers. This allows learning better features, as well as a better correspondence search strategy that progressively exploits features from higher recall (robustness) to higher precision (spatial discrimination).

pipelines that, thanks to extensive engineering efforts, have proven impervious to end-to-end learned solutions. Despite some recent efforts [28, 54, 8], deep learning solutions do not yet outperform or achieve similar generality as state-of-the-art methods on problems such as structure from motion (SfM) [56] and object pose estimation [44]. Indeed, we see a consensus emerging that some of the systems employing interest point detectors and descriptors are here to stay, but it might instead be advantageous to leverage deep learning for their individual components.

Recently, a few convolutional neural network (CNN) architectures [61, 16, 65, 58] have been proposed with the aim of learning strong geometric feature descriptors for matching images, and have yielded mixed results [49, 6]. We posit that the ability of CNNs to learn representation hierarchies, which has made them so valuable for many visual recognition tasks, becomes a hurdle when it comes to low-level geometric feature learning, unless specific design choices are made in training and inference to exploit that hierarchy. This paper presents such strategies for the problem of *dense geometric correspondence*.

Most recent works employ various metric learning losses and extract feature descriptors from the deepest layers [61, 16, 65, 58], with the expectation that the loss would yield good features right before the location of the loss layer. On the contrary, several studies [64, 68] suggest that deeper layers respond to high-level abstract concepts and are by design invariant to local transformations in the input image. However, shallower layers are found to be more sensitive to local structure, which is not exploited by most deep-learning based approaches for geometric correspondence that use only deeper layers. To address this, we propose a novel *hierarchical metric learning* approach that combines the best characteristics of various levels of feature hierarchies, to simultaneously achieve robustness and localization sensitivity. Our framework is widely applicable, which we demonstrate through improved matching for interest points in both 2D and 3D data modalities, on KITTI Flow [42] and 3DMatch [65] datasets, respectively.

Further, we leverage recent studies that highlight the importance of carefully marshaling the training process: (i) by deeply supervising [31, 33] intermediate feature layers to learn task-relevant features, and (ii) on-the-fly hard negative mining [16] that forces each iteration of training to achieve more. Finally, we exploit the intermediate activation maps generated within the CNN itself as a proxy for image pyramids traditionally used to enable coarse-to-fine matching [17]. Thus, at test time, we employ a *hierarchical matching* framework, using deeper features to perform coarse matching that benefits from greater context and higher-level visual concepts, followed by a fine grained matching step that involves searching for shallower features. Figure 1 illustrates our proposed approach.

In summary, our contributions include:

- We demonstrate that while in theory metric learning should produce good features irrespective of the layer the loss is applied to, in fact shallower features are superior for high-precision geometric matching tasks, whereas deeper features help obtain greater recall.
- We leverage deep supervision [31, 33] for feature descriptor learning, while employing hard negative mining at multiple layers.
- We propose a CNN-driven scheme for coarse-to-fine hierarchical matching, as an effective and principled replacement for conventional pyramid approaches.
- We experimentally validate our ideas by comparing against state-of-the-art geometric matching approaches and feature fusion baselines, as well as perform an ablative analysis of our proposed solution. We evaluate for the tasks of 2D and 3D interest point matching and refinement, as well as optical flow, demonstrating state-of-the-art results and generalization ability.

We review literature in Section 2 and introduce our framework in Section 3. We discuss experimental results in Section 4, concluding the paper in Section 5.

2 Related Work

With the use of deep neural networks, many new ideas have emerged both pertaining to learned feature descriptors and directly learning networks for low-level vision tasks in an end-to-end fashion, which we review next.

Hand-Crafted Descriptors. SIFT [40], SURF [7], BRISK [32] were designed to complement high curvature point detectors, with [40] even proposing its own algorithm for such a detector. In fact, despite the interest in learned methods, they are still the state-of-the-art for precision [49, 6], even if they are less effective in achieving high recall rates.

Learned Descriptors. While early work [59, 39, 36] leveraged intermediate activation maps of a CNN trained with an arbitrary loss for keypoint matching, most recent methods rely on an explicit metric loss [63, 22, 61, 16, 65, 60, 66] to learn descriptors. The hidden assumption behind using contrastive or triplet loss at the final layer of a CNN is that this explicit loss will cause the relevant features to emerge at the top of the feature hierarchy. But it has also been observed that early layers of the CNN are the ones that learn local geometric features [64]. Thus, many of these works show superior performance to handcrafted descriptors on semantic matching tasks but often lag behind on geometric matching.

Matching in 2D. LIFT [61] is a moderately deep architecture for end-to-end interest point detection and matching, which uses features at a single level of hierarchy and does not perform dense matching. Universal Correspondence Network (UCN) [16] combines a fully convolutional network in a Siamese setup, with a spatial transformer module [26] and contrastive loss [15] for dense correspondence, to achieve state-of-the-art on semantic matching tasks but not on geometric matching. Like them, we use GPU to speed up k-nearest neighbour for on-the-fly hard negative mining, albeit across multiple feature learning layers. Recently, AutoScaler [58] explicitly applies a learned feature extractor on multiple scales of the input image. While this takes care of the issue that a deep layer may have an unnecessarily large receptive field when learning on the basis of contrastive loss, we argue that it is more elegant for the CNN to "look at the image" at multiple scales, rather than separately process multiple scales.

Matching in 3D. Descriptors for matching in 3D voxel grid representations are learned by 3DMatch [65], employing a Siamese 3D CNN setup on a 30x30x30 cm^3 voxel grid with a contrastive loss. It performs self-supervised learning by utilizing RGB-D scene reconstructions to obtain ground truth correspondence labels for training, outperforming a state-of-the-art hand-crafted descriptor [48]. Thus, 3DMatch provides an additional testbed to validate our ideas, where we report positive results from incorporating our hierarchical metric learning and matching into the approach.

Learned Optical Flow. Recent works achieve state-of-the-art results on optical flow by training CNNs in an end-to-end fashion [20, 25], followed by Conditional Random Field (CRF) inference [45] to capture detailed boundaries. We also demonstrate the efficacy of our matching on optical flow benchmarks. However, we do not use heavily engineered or end-to-end learning for minimizing flow metrics, rather we show that our matches along with an off-the-shelf interpolant [45] already yield strong results.

Deep Supervision. Recent works [31, 33, 34] suggest that providing explicit supervision to intermediate layers of a CNN can yield higher performance on unseen data, by regularizing the training process. However, to the best of our knowledge, the idea has neither been tested on the task of keypoint matching nor had the learned intermediate features been evaluated. We do both in our work.

Image Pyramids and Hierarchical Fusion. Downsampling pyramids have been a steady fixture of computer vision for exploiting information across multiple scales [41]. Recently, many techniques have been developed for fusing features from different layers within a CNN and producing output at high resolution, *e.g.* semantic segmentation [23, 46, 43, 12], depth estimation [21], and optical flow [20, 25]. Inspired by [17] for image alignment, we argue that the growing receptive field in deep CNN layers [64] provides a natural way to parse an image at multiple scales. Thus, in our hierarchical matching scheme, we employ features extracted from a deeper layer with greater receptive field and higher-level semantic notions [68] for coarsely locating the corresponding point, followed by shallower features for precise localization. We show gains in correspondence estimation by using our approach over prior feature fusion methods, *e.g.* [23, 43].

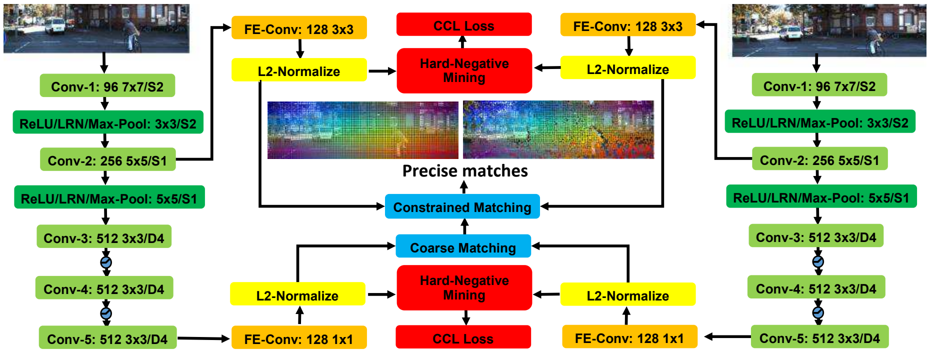


Fig. 2: One instantiation of our proposed ideas. Note that the hard negative mining and CCL losses (red blocks) are relevant for training, and matching (blue blocks) for testing. Convolutional blocks (green) in the left and right Siamese branches share weights. *'S'* and *'D'* denote striding and dilation offsets.

3 Method

In the following, we first identify the general principles behind our framework, then propose concrete neural network architectures that realize them. In this section, we limit our discussion to models for 2D images. We detail and validate our ideas on the 3DMatch [65] architecture in Section 4.3.

3.1 Hierarchical Metric Learning

We follow the standard CNN-based metric learning setup proposed as the Siamese architecture [15]. This involves two Fully Convolutional Networks (FCN) [38] with tied weights, parsing two images of the same scene. We extract features out of the intermediate convolutional layer activation maps at the locations corresponding to the training points, and after normalization obtain their Euclidean distance. At training time, separate contrastive losses are applied to multiple levels in the feature hierarchy to encourage the network to learn embedding functions that minimizes the distance between the descriptors of matching points, while maximizing the distance between unmatched points.

Correspondence Contrastive Loss (CCL). We borrow the correspondence contrastive loss formulation introduced in [16], and adapted from [15]. Here, $\phi_l^I(x)$ represents the feature extracted from the l-th feature level of the reference image I at a pixel location x; similarly, $\phi_l^{I'}(x')$ represents the feature extracted from the l-th feature level of the target image I' at a pixel location x'. Let $\mathcal{D}$ represent a dataset of triplets (x, x', y), where x is a location in the reference image I, x' is a location in the target image I', and $y \in \{0, 1\}$ is 1 if and only if (x, x') are a match. Let m be a margin parameter and c be a window size. We define:

$$\hat{\phi}_l^I(x) := \frac{\phi_l^I(x)}{\|\phi_l^I(x)\|_2}, \qquad d_l(x, x') := \|\hat{\phi}_l^I(x) - \hat{\phi}_l^{I'}(x')\|_2. \tag{1}$$

Then, our training loss, $\mathcal{L}$, sums CCL losses over multiple levels l:

$$\mathcal{L} := \sum_{l=1}^{L} \sum_{(x,x',y)\in\mathcal{D}} y \ . \ d_l^2(x,x') + (1-y) \ . \ (\max(0, m - d_l(x,x')))^2. \qquad (2)$$

Deep Supervision. Our rationale in applying CCL losses at multiple levels of the feature hierarchy is twofold. Recent studies [31, 33] indicate that deep supervision contributes to improved regularization, by encouraging the network early on to learn task-relevant features. Secondly, both deep and shallow layers can be supervised for matching simultaneously within one network.

Hard Negative Mining. Since our training data includes only positive correspondences, we actively search for hard negative matches "on-the-fly" to speed up training and to leverage the latest instance of network weights. We adopt the approach of UCN [16], but in contrast to it, our hard negative mining happens independently for each of the feature levels being supervised.

Network Architectures. We visualize one specific instantiation of the above ideas in Figure 2, adapting the VGG-M [11] architecture for the task. We retain the first 5 convolutional layers, initializing them with weights pre-trained for ImageNet classification [47]. We use ideas from semantic segmentation literature [62, 12] to increase the resolution of the intermediate activation maps by (a) eliminating down-sampling in the second convolutional and pooling layers (setting their stride value to 1, down from 2) (b) increasing the pooling window size for the second layer from 3x3 to 5x5 and (c) dilating [62] the subsequent convolutional layers (*conv3*, *conv4* and *conv5*) to retain their pretrained receptive fields.

At training, the network is provided with a pair of images and a set of point correspondences. The network is replicated in a Siamese scheme [15] during training (with shared weights) where each sub-network processes one image from the pair; and thus after each feed-forward pass, we have 4 feature maps: 2 shallow ones and 2 deep ones, respectively from the second and fifth convolutional layers (*conv2*, *conv5*). We apply supervision after these same layers (*conv2*, *conv5*).

We also experiment with a GoogLeNet [52] baseline as employed in UCN [16]. Specifically, we augment the network with a 1x1 convolutional layer and L2 normalization following the fourth convolutional block (*inception_4*a/output) for learning deep features, as in UCN. In addition, for learning shallow features, we augment the network with a 3x3 convolutional layer right after the second convolutional layer (*conv2*/3x3), followed by L2 normalization, but before the corresponding non-linear ReLU squashing function. We extract the shallow and deep feature maps based on the normalized outputs after the second convolutional layer *conv2*/3x3 and the *inception_4*a/output layers respectively. We provide the detailed architecture of our GoogLeNet variant as supplementary material.

Network Training. We implement our system in Caffe [27] and use ADAM [29] to train our network for $50K$ iterations using a base learning rate of 10^{-3} on a P6000 GPU. Pre-trained layers are fine-tuned with a learning rate multiplier of 0.1 whereas the weights of the newly-added feature-extraction layers are randomly initialized using Xavier's method. We use a weight decay parameter of 10^{-4} and L2 weight regularization. During training, each batch consists of three randomly chosen image pairs and we randomly choose 1K positive correspondences from each pair. It takes the VGG-M variant of our system around 43 hours to train whereas it takes 30 hours to train our GoogLeNet-based variant.

3.2 Hierarchical Matching

We adapt and train our networks as described in the previous section, optimizing network weights for matching using features extracted from different layers. Yet, we find that features from different depths offer complementary capabilities as predicted by earlier works [64, 68] and confirmed by our empirical evaluation in Section 4. Specifically, features extracted from shallower layers obtain superior matching accuracies for smaller distance thresholds (precision), whereas those from deeper layers provide better accuracies for larger distance thresholds (recall). Such coarse-to-fine matching has been well-known in computer vision [41], however recent work highlights how employing CNN feature hierarchies for the task (at least in the context of image alignment [17]) is more robust.

To establish correspondences, we compare the deep and shallow features of the input images I and I' as follows. Assuming the shallow feature coordinates p_s and the deep feature coordinates p_d in the reference image I are related by $p_d = p_s * 1/f$ with a scaling factor f, we first use the deep feature descriptor $\phi_d^I(p_d)$ in the reference image I to find the point p'_d in the target image I' with $\phi_d^{I'}(p'_d)$ closest to $\phi_d^I(p_d)$ with nearest neighbor search.[1] Next, we refine the location of p'_d by searching within a circle of a radius of 32 pixels around $p'_s = p'_d * f$ (assuming input images have the same size, thus, $f' = f$) to find the point $\hat{p}'_s$ whose shallow feature descriptor $\phi_s^{I'}(\hat{p}'_s)$ is closest to $\phi_s^I(p_s)$, forming a correspondence $(p_s, \hat{q}'_s)$.

Our proposed hierarchical matching is implemented on CUDA and run on a P6000 GPU, requiring an average of 8.41 seconds to densely extract features and compute correspondences for a pair of input images of size 1242 × 376.

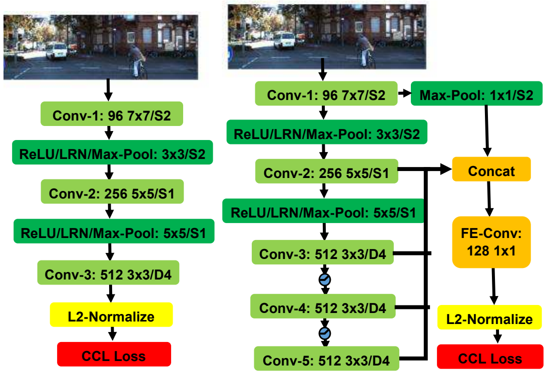
4 Experiments

In this section, we first benchmark our proposed method for 2D correspondence estimation against standard metric learning and matching approaches, feature fusion, as well as state-of-the-art learned and hand-crafted methods for extracting correspondences. Next, we show how our method for correspondence estimation can be applied for optical flow and compare it against recent optical flow methods. Finally, we incorporate our ideas in a state-of-the-art 3D fully convolutional network [65] and show improved performance. In the following, we denote our method as *HiLM*, which is short for *Hi*erarchical metric *L*earning and *M*atching.

4.1 2D Correspondence Experiments

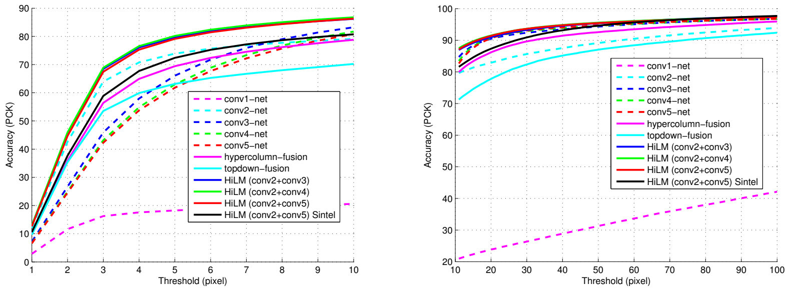
We empirically evaluate our ideas against different approaches for dense correspondence estimation. We first consider metric learning and matching approaches

[1] If p_d is fractional, we use bilinear interpolation to compute $\phi_d^I(p_d)$.



(a) conv3-net (b) hypercolumn-fusion

Fig. 3: One Siamese branch of two for baseline architectures in our evaluation. The *conv3-net* (a) is obtained by truncating all layers after VGG-M *conv3* in Figure 2 and adding a convolutional layer, L2 normalization and CCL loss. Other *convi-net* baselines are obtained similarly. The 1x1 max pooling layer after *conv1* in the *hypercolumn-fusion* baseline (b) is added to down sample the *conv1* feature map for valid concatenation with other feature maps. *'S'* and *'D'* denote striding and dilation offsets.



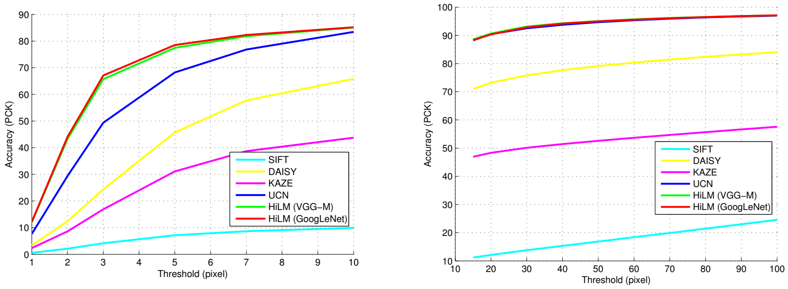
(a) Accuracy over small thresholds **(b)** Accuracy over large thresholds

Fig. 4: Accuracy of different CNN-based methods for 2D correspondence estimation on KITTI Flow 2015.

based on feature sets extracted from a single convolutional layer [2], where we separately train five networks, based on the VGG-M baseline in Figure 2. Each one of the five networks has a different depth and we refer to the i-th network by *convi-net* to indicate that the network is truncated at the i-th convolutional layer (*convi*), for $i \in 1, 2, ..., 5$. We train a *convi-net* network by adding a convolutional layer, L2 normalization, and CCL loss after the output of the last layer (*convi*). Figure 3 (a) shows one branch of the *conv3-net* baseline as an example.

In addition, we also compare our method against two alternatives for fusing features from different layers inspired by ideas from semantic segmentation [23, 43]. One is *hypercolumn-fusion* – Figure 3 (b), where feature sets from all layers (first through fifth) are concatenated for every interest point and a set of 1x1

[2] LIFT [61] is not designed for dense matching and hence not included in our experiments. Note that LIFT also uses features from only a single convolutional layer.



(a) Accuracy over small thresholds

(b) Accuracy over large thresholds

Fig. 5: Accuracy of CNN-based and hand-crafted methods for 2D correspondence estimation on KITTI Flow 2015.

convolution kernels are trained to fuse features before L2 normalization and CCL loss. Instead of upsampling deeper feature maps as in [23], we extract deep features at higher resolution by setting the stride of multiple convolutional/pooling layers to 1 while dilating the subsequent convolutions appropriately as shown in Figure 3. Another approach we consider is *topdown-fusion*, where refinement modules similar to [43] are used to refine the top-level *conv5* features gradually down the network by combining with lower-level features till *conv2* (please see supplementary material for details).

We evaluate on KITTI Flow 2015 [42] where all networks are trained on 80% of the image pairs and the remaining 20% are used for evaluation. For a fair comparison, we use the same train-test split for all methods and train each with 1K correspondences per image pair and for 50K iterations. During testing, we use the correspondences $\{(x_i, x'_i)\}$ in each image pair (obtained using all non-occluded ground truth flows) for evaluation. Specifically, each method predicts a point $\hat{x}'_i$ in the target image that matches the point x_i from the reference image $\forall i$.

Evaluation Metric. Following prior works [39, 16, 58], we use Percentage of Correct Keypoints (PCK) as our evaluation metric. Given a pixel threshold θ, the PCK measures the percentage of predicted points $\hat{x}'_i$ that are within θ pixels from the ground truth corresponding point x'_i (and so are considered as correct matches up to θ pixels).

Single-Layer and Feature Fusion Descriptors. We plot PCK curves obtained for all methods under consideration in Figure 4 where we split the graph into sub-graphs based on the pixel threshold range. These plots reveal that, for smaller thresholds, shallower features (*e.g. conv2-net* with 73.89% @ 5 pixels) provide higher PCK than deeper ones (*e.g. conv5-net* with 61.78% @ 5 pixels), with the exception of *conv1-net* which performs worst. Contrarily, deeper features have better performance for higher thresholds (*e.g. conv5-net* with 87.57% versus *conv2-net* with 81.36% @ 15 pixels). This suggests that, for best performance, one would need to utilize the shallower as well as deeper features produced by the network rather than just the output of the last layer.

The plot also indicates that while baseline approaches for fusing features improve the PCK for smaller thresholds (*e.g.* *hypercolumn-fusion* with 69.41% versus *conv5-net* with 61.78% @ 5 pixels), they do not perform on par with the simple *conv2*-based features (*e.g.* *conv2-net* with 73.89% @ 5 pixels).

Different variants of our full approach achieve the highest PCK for smaller thresholds (*e.g.* HiLM (*conv2+conv4*) with 80.17% @ 5 pixels), without losing accuracy for higher thresholds. In fact, our method is able to outperform the *conv2* features (*e.g.* *conv2-net* with 73.89% @ 5 pixels) although it uses them for refining the rough correspondences estimated by the deeper layers. This is explained by the relative invariance of deeper features to local structure, which helps to avoid matching patches that have similar local appearance but rather belong to different objects.

Generalization. We also perform experiments on cross-domain generalization ability. Specifically, we train HiLM (*conv2+conv5*) on MPI Sintel [10] and evaluate it on KITTI Flow 2015 as the previous experiment, plotting the result in Figure 4 (black curve). As expected the Sintel model is subpar compared to the same model trained on KITTI (72.37% vs. 79.11% @ 5 pixels), however it outperforms both *hypercolumn-fusion* (69.41%) and *topdown-fusion* (63.14%) trained on KITTI, across all PCK thresholds. Similar generalization results are obtained when cross-training with HPatches [6] (please see supplementary material for details).

Hand-Crafted Descriptors. We also compare the performance of (a) our HiLM (*conv2+conv5*, VGG-M), (b) a variant of our method based on GoogLeNet/ UCN (described in Section 3), (c) the original UCN [16], and (d) the following hand-crafted descriptors: SIFT [40], KAZE [2], DAISY [53]. We use the same KITTI Flow 2015 evaluation set utilized in the previous experiment. To evaluate hand-crafted approaches, we use them to compute the descriptors at test pixels in the reference image (for which ground truth correspondences are available) and match the resulting descriptors against the descriptors computed on the target image over a grid of 4 pixel spacing in both directions.

Figure 5 compares the resulting PCKs and shows that our HiLM (VGG-M) outperforms UCN [16] for smaller thresholds (*e.g.* HiLM (VGG-M) with 43.26% versus UCN with 29.38% @ 2 pixels). That difference in performance is not the result of baseline shift since our GoogLeNet variant (same baseline network as UCN) has similar or slightly better performance compared to our VGG-M variant. The graph also indicates the relatively higher invariance of CNN-based descriptors to local structure that allows them to obtain a higher percentage of roughly-localized correspondences (*e.g.* UCN with 83.42%, HiLM (VGG-M) with 85.08%, and HiLM (GoogLeNet) with 85.18%, all at 10 pixel threshold).

4.2 Optical Flow Experiments

In this section, we demonstrate the application of our geometric correrspondences for obtaining optical flows. We emphasize that the objective here is not to outperform methods that have been extensively engineered [4, 50, 25] for optical flows, including minimizing flow metric (end-point error) directly, *e.g.* FlowNet2 [25].

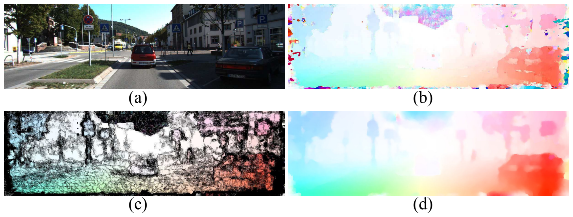


Fig. 6: Optical flow pipeline. (a) Input image. (b) Initial HiLM matches. (c) Filtered matches after consistency checks and motion constraints. (d) After interpolation using EpicFlow [45].

Method	*Fl-bg*	*Fl-fg*	*Fl-all*
FlowNet2 [25]	*10.75*%	**8.75**%	**10.41**%
SDF [4]	**8.61**%	26.69%	*11.62*%
SOF [50]	14.63%	27.73%	16.81%
CNN-HPM [5]	18.33%	24.96%	19.44%
HiLM (Ours)	23.73%	*21.79*%	23.41%
SPM-BP [35]	24.06%	24.97%	24.21%
FullFlow [13]	23.09%	30.11%	24.26%
AutoScaler [58]	21.85%	31.62%	25.64%
EpicFlow [45]	25.81%	33.56%	27.10%
DeepFlow2 [59]	27.96%	35.28%	29.18%
PatchCollider [57]	30.60%	33.09%	31.01%

Table 1: Quantitative results on KITTI Flow 2015. Following KITTI convention: *'Fl-bl'*, *'Fl-fg'*, and *'Fl-all'* represent the outlier percentage on background pixels, foreground pixels and all pixels respectively. The methods are ranked by their *'Fl-all'* errors. **Bold** numbers represent best results, while *underlined* numbers are second best ones. Note that FlowNet2 [25] optimizes flow metric directly, while SDF [4] and SOF [50] require semantic knowledge.

Yet, we consider it useful to garner insights from flow benchmarks since the tasks (*i.e.* geometric correspondence and optical flow) are conceptually similar.

Network Architecture. For dense optical flow estimation, we leverage GoogLeNet [52] as our backbone architecture. However, at test time, we modify the trained network to obtain dense per-pixel correspondences. To this end: (i) we set the stride to 1 in the first convolutional and pooling layers (*conv1* and *pool1*), (ii) we set the kernel size of the first pooling layer (*pool1*) to 5 instead of 3, (iii) we set the dilation offset of the second convolutional layer (*conv2*) to 4, and (iv) we set the stride of the second pooling layer (*pool2*) to 4. These changes allow us to obtain our shallow feature maps at the same resolution as the input images (W x H) and the deep feature maps at $W/4$ x $H/4$, and to obtain dense per-pixel correspondences faster and with significantly fewer requirements on the GPU memory as compared to an approach that would process the feature maps at full resolution through all layers of the network.

Procedure. We first extract and match feature descriptors for every pixel in the input images using our proposed method. These initial matches are usually contaminated by outliers or incorrect matches. Therefore, we follow the protocol of AutoScaler[58] for outlier removal. In particular, we enforce local motion constraints using a window of $[-240, 240]$x$[-240, 240]$ and perform forward-backward consistency checks with a threshold of 0 pixel. These filtered matches are then fed to EpicFlow [45] interpolation for producing the final optical flow output. Figure 6 illustrates an example of this procedure.

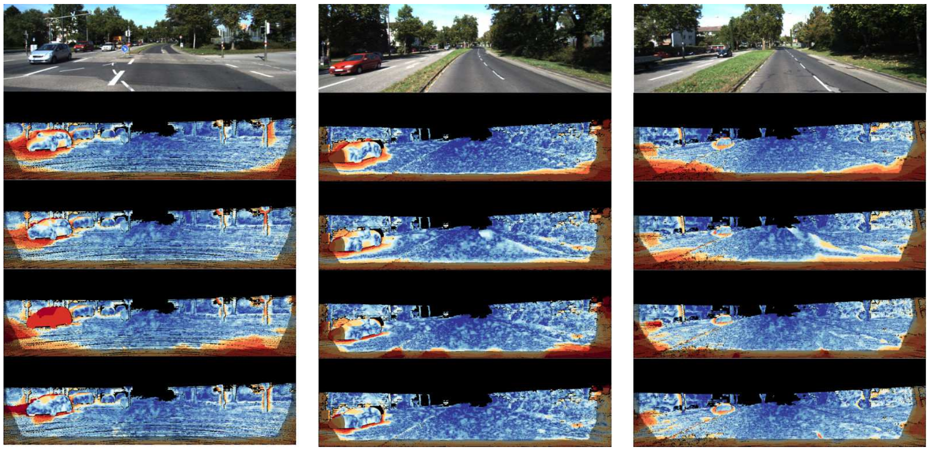


Fig. 7: Qualitative results on KITTI Flow 2015. First row: input images. Second row: DeepFlow2 [59]. Third row: EpicFlow [45]. Forth row: SPM-BP [35]. Fifth row: HiLM. Red colors mean high errors while blue colors mean low errors.

Quantitative Evaluation. We tabulate our quantitative evaluation results on KITTI Flow 2015 in Table 1. As mentioned earlier, our objective is not necessarily to obtain the best optical flow performance, rather we wish to emphasize that we are able to provide high-quality interest point matches. In fact, many recent works [4, 50] focus on embedding rich domain priors at the level of explicit object classes into their models, which allows them to make good guesses when data is missing (*e.g.* due to occlusions, truncations, homogenous surfaces). Yet, we are able to outperform several methods in our comparisons except [25] for foreground pixels (*i.e.* by *Fl-fg*, HiLM with 21.79% versus other methods with 24.96–35.28%, excluding [25] with 8.75%). As expected, we do not get as good matches in regions of the image where relatively less structure is present (*e.g.* background), and for such regions methods [4, 50] employing strong prior models have significant advantages. However, even on background regions, we are able to either beat or perform on par with most of our competitors (*i.e.* by *Fl-bg*, 23.73% versus 18.33–30.60%), including machinery proposed for optical flows such as [59, 45, 13]. Overall, we outperform 6 state-of-the-art methods evaluated in Table 1 (*i.e.* by *Fl-all*), including the multi-scale correspondence approach of [58].

Qualitative Evaluation. We plot some qualitative results in Figure 7, to contrast DeepFlow2 [59], EpicFlow [45], and SPM-BP [35] against our method. As expected from the earlier discussion, we observe superior results for our method on the image regions belonging to the vehicles, because of strong local structures, whereas for instance in first column (fourth row) SPM-BP [35] entirely fails on the blue car. We observe errors in the estimates of our method largely in regions which are occluded (surroundings of other cars) or truncated (lower portion of the images), where the competing methods also have high errors.

4.3 3D Correspondence Experiments

To demonstrate the generality of our contributions to different data modalities, we now consider an extension of our proposed method in Section 3 to 3D correspondence estimation. In the following, we first present the details of our network architecture and then discuss the results of our quantitative evaluation.

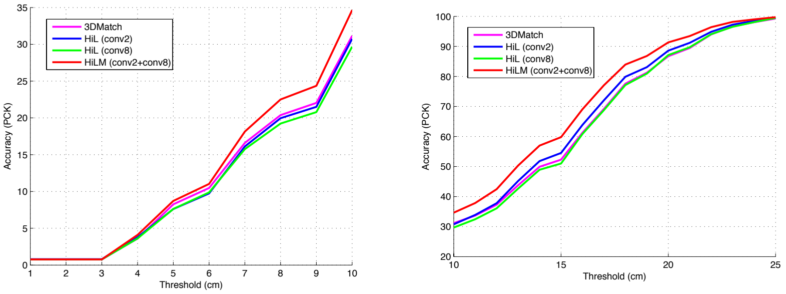
Network Architecture. We use 3DMatch [65] as our baseline architecture. We insert two 3x3x3 convolutional layers (stride of 2 each) and one 5x5x5 pooling layer (stride of 1) after the second convolutional layer of 3DMatch to obtain a 512-dimensional vector, which serves as the shallow feature descriptor. Our deep feature descriptor is computed after the eighth convolutional layer in the same manner as 3DMatch. Our hierarchical metric learning scheme again employs two CCL losses (Section 3.1) for learning shallow and deep feature descriptors simultaneously. We disable hard negative mining in this experiment to enable a fair comparison with 3DMatch. Our network is implemented in Marvin [1] and trained with stochastic gradient descent using a base learning rate of 10^{-3} for 137K iterations on a TITAN XP GPU. We use pre-trained weights provided by 3DMatch to initialize the common layers in our network, which have a learning rate multiplier of 0.1, whereas the weights of the newly added layers are initialized using Xavier's method and have a learning rate multiplier of 1.0. We generate correspondence data for training using the same procedure as 3DMatch.

Protocol. 3DMatch evalutes classification accuracy of putative correspondences, using fixed keypoint locations and binary labels. Since our method enables refinement with shallow features and hence shifts hypothesized correspondence location in space, we define a protocol suitable to measure refinement performance. We employ PCK as our evaluation metric, similar to 2D experiments. We generate test data consisting of 10K ground truth correspondences using the procedure of 3DMatch. We use a region of 30x30x30 cm^3 centered on the reference keypoint (in the reference "image") following [65] to compute the reference descriptor. This is matched against putative keypoints in a 60x60x60 cm^3 region (in the target "image"), to refine this coarse prior estimate[3]. Specifically, we divide this region into subvolumes of 30x30x30 cm^3 and employ our hierarchical matching approach to exhaustively search [4] for the subvolume whose descriptor is most similar to the reference descriptor. In particular, once the coarse matching using deeper feature descriptors yields an approximate location in the 60x60x60 cm^3 region, we constrain the refinement by shallow feature descriptors to a search radius of 15 cm around the approximate location returned from the coarse matching.

Quantitative Evaluation. We compare our complete framework, namely, HiLM (*conv2*+*conv8*) against variants which are trained with hierarchical metric loss but rely either on deep or shallow features for matching (HiL (*conv8*) and HiL (*conv2*), respectively), and 3DMatch which use only deep features. Figure 8 shows the PCK curves of all competing methods computed over 10K test

[3] In fact, the ground truth keypoint correspondence lies at the center of this region, but this knowledge is not available to the method in any way.

[4] We use a sampling gap of 3 cm along all three dimensions in searching for subvolumes to reduce computational costs.



(a) Accuracy over small thresholds

(b) Accuracy over large thresholds

Fig. 8: Accuracy of different CNN-based methods for 3D correspondence estimation.

correspondences generated by the procedure of 3DMatch. From the results, our shallow features trained with hierarchical metric learning are able to outperform their deep counterparts for most PCK thresholds (*e.g.* HiL (*conv2*) with 21.50% versus HiL (*conv8*) with 20.78% @ 9 cm). By utilizing both deep and shallow features, our complete framework achieves higher PCK numbers than its variants and outperforms 3DMatch across all PCK thresholds (*e.g.* HiLM (*conv2*+*conv8*) with 24.36% versus 3DMatch with 22.04% @ 9 cm).

5 Conclusion and Future Work

We draw inspiration from recent studies [64, 68] as well as conventional intuitions about CNN architectures to enhance learned representations for dense 2D and 3D geometric matching. Convolutional network architectures naturally learn hierarchies of features, thus, a contrastive loss applied at a deep layer will return features that are less sensitive to local image structure. We propose to remedy this by employing features at multiple levels of the feature hierarchy for interest point description. Further, we leverage recent ideas in deep supervision to explicitly obtain task-relevant features at intermediate layers. Finally, we exploit the receptive field growth for increasing layer depths as a proxy to replace conventional coarse-to-fine image pyramid approaches for matching. We thoroughly evaluate these ideas realized as concrete network architectures, on challenging benchmark datasets. Our evaluation on the task of explicit keypoint matching outperforms hand-crafted descriptors, a state-of-the-art descriptor learning approach [16], as well as various ablative baselines including hypercolumn-fusion and topdown-fusion. Further, an evaluation for optical flow computation outperforms several competing methods even without extensive engineering or leveraging higher-level semantic scene understanding. Finally, augmenting a recent 3D descriptor learning framework [65] with our ideas yields performance improvements, hinting at wider applicability. Our future work will explore applications of our correspondences, such as flexible ground modeling [30, 19, 3] and geometric registration [14, 65].

References

1. Marvin: A minimalist GPU-only N-dimensional ConvNet framework. `http://marvin.is`, accessed: 2015-11-10
2. Alcantarilla, P.F., Bartoli, A., Davison, A.J.: KAZE features. In: ECCV (2012)
3. Ansari, J.A., Sharma, S., Majumdar, A., Murthy, J.K., Krishna, K.M.: The Earth ain't Flat: Monocular Reconstruction of Vehicles on Steep and Graded Roads from a Moving Camera. In: ArXiv (2018)
4. Bai, M., Luo, W., Kundu, K., Urtasun, R.: Exploiting Semantic Information and Deep Matching for Optical Flow. In: ECCV (2016)
5. Bailer, C., Varanasi, K., Stricker, D.: CNN-based Patch Matching for Optical Flow with Thresholded Hinge Embedding Loss. In: CVPR (2017)
6. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: CVPR (2017)
7. Bay, H., Tuytelaars, T., Gool, L.V.: SURF: Speeded Up Robust Features. In: ECCV (2006)
8. Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Frank Michel, S.G., Rother, C.: DSAC - Differentiable RANSAC for Camera Localization. In: CVPR (2017)
9. Brox, T., Malik, J.: Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation. PAMI **33**(3), 500–513 (2011)
10. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: ECCV (2012)
11. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: BMVC (2014)
12. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. PAMI (2017)
13. Chen, Q., Koltun, V.: Full Flow: Optical Flow Estimation by Global Optimization over Regular Grids. In: CVPR (2016)
14. Choi, S., Zhou, Q.Y., Koltun, V.: Robust Reconstruction of Indoor Scenes. In: CVPR (2015)
15. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR (2005)
16. Choy, C.B., Gwak, J., Savarese, S., Chandraker, M.: Universal Correspondence Network. In: NIPS (2016)
17. Czarnowski, J., Leutenegger, S., Davison, A.J.: Semantic Texture for Robust Dense Tracking. In: ICCVW (2017)
18. Dalal, N., Triggs, B.: Histogram of Oriented Gradients for Human Detection. In: CVPR (2005)
19. Dhiman, V., Tran, Q.H., Corso, J.J., Chandraker, M.: A Continuous Occlusion Model for Road Scene Understanding. In: CVPR (2016)
20. Dosovitskiy, A., Fischer, P., Ilg, E., Husser, P., Hazirbas, C., Golkov, V., v.d. Smagt, P., Cremers, D., Brox, T.: FlowNet: Learning Optical Flow with Convolutional Networks. In: ICCV (2015)
21. Eigen, D., Puhrsch, C., Fergus, R.: Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In: NIPS (2014)
22. Gadot, D., Wolf, L.: PatchBatch: A Batch Augmented Loss for Optical Flow. In: CVPR (2016)
23. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR (2015)

24. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the Alvey Vision Conference (AVC) (1988)
25. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In: CVPR (2017)
26. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial Transformer Networks. In: NIPS (2015)
27. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM Multimedia (2014)
28. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In: ICCV (2015)
29. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. ICLR (2014)
30. Lee, B., Daniilidis, K., Lee, D.D.: Online Self-Supervised Monocular Visual Odometry for Ground Vehicles. In: ICRA (2015)
31. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-Supervised Nets. AISTATS (2015)
32. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: Binary Robust Invariant Scalable Keypoints. In: ICCV (2011)
33. Li, C., Zia, M.Z., Tran, Q.H., Yu, X., Hager, G.D., Chandraker, M.: Deep Supervision with Shape Concepts for Occlusion-Aware 3D Object Parsing. In: CVPR (2017)
34. Li, C., Zia, M.Z., Tran, Q.H., Yu, X., Hager, G.D., Chandraker, M.: Deep Supervision with Intermediate Concepts. In: ArXiv (2018)
35. Li, Y., Min, D., Brown, M.S., Do, M.N., Lu, J.: SPM-BP: Sped-up PatchMatch Belief Propagation for Continuous MRFs. In: ICCV (2015)
36. Lin, K., Lu, J., Chen, C.S., Zhou, J.: Learning Compact Binary Descriptors with Unsupervised Deep Neural Networks. In: CVPR (2016)
37. Lindeberg, T.: Feature detection with automatic scale selection. IJCV **30**(2), 79–116 (1998)
38. Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. In: CVPR (2015)
39. Long, J.L., Zhang, N., Darrel, T.: Do Convnets Learn Correspondence? In: NIPS (2014)
40. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2), 91–110 (2004)
41. Lucas, B.D., Kanade, T.: Optical Navigation by the Method of Differences. In: IJCAI (1985)
42. Menze, M., Geiger, A.: Object Scene Flow for Autonomous Vehicles. In: CVPR (2015)
43. Pinheiro, P.O., Lin, T.Y., Collobert, R., Dollár, P.: Learning to refine object segments. In: ECCV (2016)
44. Rad, M., Lepetit, V.: BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth. In: ICCV (2017)
45. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In: CVPR (2015)
46. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: MICCAI (2015)
47. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)

48. Rusu, R.B., Blodow, N., Beetz, M.: Fast Point Feature Histograms (FPFH) for 3D registration. In: ICRA (2009)
49. Schönberger, J.L., Hardmeier, H., Sattler, T., Pollefeys, M.: Comparative Evaluation of Hand-Crafted and Learned Local Features. In: CVPR (2017)
50. Sevilla-Lara, L., Sun, D., Jampani, V., Black, M.J.: Optical Flow with Semantic Segmentation and Localized Layers. In: CVPR (2016)
51. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. TOG **25**(3), 835–846 (2006)
52. 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)
53. Tola, E., Lepetit, V., Fua, P.: DAISY: An efficient dense descriptor applied to wide-baseline stereo. PAMI **32**(5), 815–830 (2010)
54. Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., Fragkiadaki, K.: SfM-Net: Learning of Structure and Motion from Video. In: ArXiv (2017)
55. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action Recognition by Dense Trajectories. In: CVPR (2011)
56. Wang, S., Clark, R., Wen, H., Trigoni, N.: DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. In: ICRA (2017)
57. Wang, S., Fanello, S., Rhemann, C., Izadi, S., Kohli, P.: The Global Patch Collider. In: CVPR (2016)
58. Wang, S., Luo, L., Zhang, N., Li, J.: AutoScaler: Scale-Attention Networks for Visual Correspondence. In: BMVC (2017)
59. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: Large Displacement Optical Flow with Deep Matching. In: ICCV (2013)
60. Yang, T.Y., Hsu, J.H., Lin, Y.Y., Chuang, Y.Y.: DeepCD: Learning Deep Complementary Descriptors for Patch Representations. In: ICCV (2017)
61. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: LIFT: Learned Invariant Feature Transform. In: ECCV (2016)
62. Yu, F., Koltun, V.: Multi-Scale Context Aggregation by Dilated Convolutions. In: ICLR (2016)
63. Zbontar, J., LeCun, Y.: Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. The Journal of Machine Learning Research (JMLR) **17**, 1–32 (2016)
64. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV (2014)
65. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In: CVPR (2017)
66. Zhang, X., Yu, F.X., Kumar, S., Chang, S.F.: Learning Spread-out Local Feature Descriptors. In: ICCV (2017)
67. Zhang, Z., Deriche, R., Faugeras, O., Luong, Q.T.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artificial Intelligence **78**, 87–119 (1995)
68. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object Detectors Emerge in Deep Scene CNNs. In: ICLR (2015)