Direct Shot Correspondence Matching

Umer Rafi[1]
rafi@vision.rwth-aachen.de
Juergen Gall[2]
gall@informatik.uni-bonn.de
Bastian Leibe[1]
leibe@vision.rwth-aachen.de

[1] Visual Computing Institute
RWTH Aachen University
Aachen, Germany

[2] Computer Vision Group
Institute of Computer Science III,
University of Bonn, Germany

Abstract

We propose a direct shot method for the task of correspondence matching. Instead of minimizing a loss based on positive and negative pairs, which requires hard-negative mining step for training and nearest neighbor search step for inference, we propose a novel similarity heatmap generator that makes these additional steps obsolete. The similarity heatmap generator efficiently generates peaked similarity heatmaps over the target image for all the query keypoints in a single pass. The matching network can be appended to any standard deep network architecture to make it end-to-end trainable with N-pairs based metric learning and achieves superior performance. We evaluate the proposed method on various correspondence matching datasets and achieve state-of-the-art performance.

1 Introduction

Correspondence search is a fundamental problem in computer vision and has applications for many vision tasks such as stereo reconstruction, optical flow estimation, image retrieval, object tracking, and articulated tracking. Variants of the task can range from finding exact matches, *e.g.*, in stereo matching, to finding semantic correspondences, *e.g.*, matching corresponding body parts of different species of birds.

Earlier work on correspondence search relied on hand-crafted features such as SIFT [17] or SURF [1]. Recently, convolutional neural networks (CNNs) have replaced hand-crafted features. In the context of correspondence search, Siamese networks have been proposed which achieve impressive results [5, 6, 10, 13, 21, 28]. Recent approaches for image-to-image semantic keypoint matching use pre-trained VGGNet [22] or GoogleNet [24] for each Siamese branch to learn deep feature descriptors and then apply matching on the top. The common strategies are to either apply a matching framework [9, 15] over the descriptors [13], or search nearest neighbor in the descriptors space learned by the deep network [5]. However, those strategies have the downside that the underlying networks are not trained for the end goal of precise localization of correspondences.

In order to enable end-to-end learning, we instead propose a novel matching network that efficiently generates a similarity heatmaps for every semantic query point in the source image over the target image in a single forward pass. Our proposed matching network can

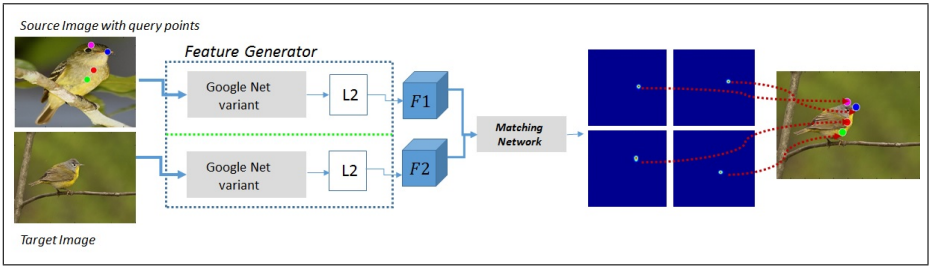

Figure 1: Our end-to-end learning framework consists of two parts: (i) a feature generator and (ii) a matching network. The feature generator takes the source image, the query points, and the target image and generates dense $\mathcal{D}$ dimensional features F_1 and F_2. The matching network takes features F_1 and F_2 as inputs and generates a peaked similarity map for each query point.

be appended to any standard deep network architecture to make it end-to-end trainable for the task of precise correspondence matching. The peaks in the heatmaps then define the locations of the correspondences in the target image. The heatmaps based representation enables N-pairs based metric learning [23] and achieves superior performance than triplets or contrastive divergence based metric learning. In addition to that, N-pairs based metric learning eliminates the need of hard-negative sampling.

Our framework can be used to predict both sparse and dense visual correspondences. We train the network with the multi-class classification loss. In contrast to [8], we use a simpler network architecture that does not include any spatial transformer layers. Compared to [10], our method operates on raw images and does not require region proposals. We evaluate the proposed framework on the PF-Pascal [9], PF-Willow [9], Pascal-Parts [30], the KITTI-Flow 2015 [19] and MPI Sintel [4] datasets respectively. We achieve state-of-the-art performance even when the network is trained from scratch.

To summarize, we make the following contributions : (i) We propose an end-to-end learning method for the task of correspondence search with a novel matching network. (ii) The proposed matching network can be appended to any standard deep network architecture to make it end-to-end trainable for the task of precise correspondence matching.. (iii) The heatmap representation enables N-pairs based metric learning and eliminates the need of hard negative sampling required in triplet or contrastive divergence based metric learning.

2 Related Work

Correspondence search is one of the fundamental problems in computer vision and has generated a lot of literature. Early work on correspondence search focused on using hand-crafted features such as SIFT [17], SURF [1], or DAISY [25].

In recent years Siamese based CNNs have been used extensively for similarity related tasks. They were first used by Bromley *et al*. [3] for signature verification. In [28] a Siamese network is used to measure patch similarity. Other Siamese networks have been used in [21] to learn face embeddings for super-human face identification performance. Zbontar *et al*. [29] used them for stereo matching. Long *et al*. [16] analyzed a CNN pre-trained on ImageNet for the task of semantic correspondence search. In [26] a CNN is trained with

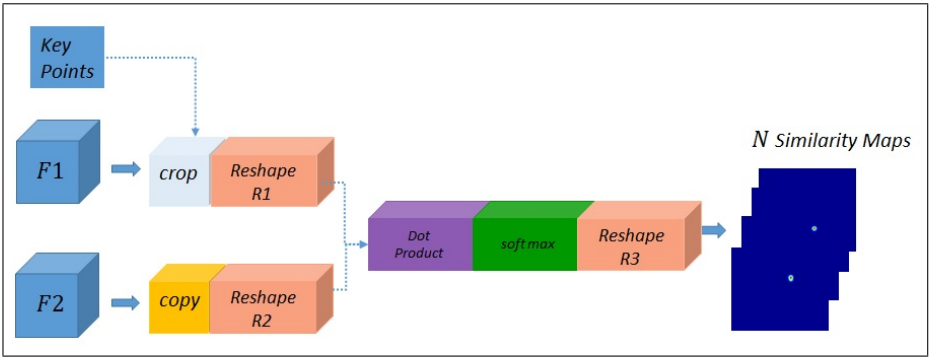

Figure 2: Matching Network. It takes F_1, F_2 and N query points as inputs and outputs N similarity maps for N query-points in image $\mathcal{I}_1$.

triplet loss for fine-grained image ranking. [18] trained a Siamese network with a dot product layer and multi-class classification loss for efficient disparity estimation.

Despite impressive performance, all of the above-mentioned approaches estimate patch-patch or patch-image similarity and require multiple forward passes during training and inference for matching similarity of multiple key-points.

Recently approaches that perform image-image semantic keypoints matching have been proposed . Choy *et al*. [5] introduced a "Universal Correspondence Network" with spatial transformer layers [11]. Their network is trained efficiently using metric learning and requires a single forward pass at inference time for matching multiple key points. However, due to metric learning it requires an extra hard negative mining step that introduces the extra distance measure and k-nearest neighbor hyper-parameters. In contrast, we present an end-end learning framework that does not require the extra hard negative mining step. Kim *et al*. [13] proposed a fully convolutional self-simalrity descriptor for dense semantic keypoints matching. However, their approach still requires a matching framework e.g [9] on the top to establish correspondences. In contrast, our system is self contained and predicts the correspondences directly. [10] recently proposed a system that matches region proposals across a pair of images using appearances and geometry. Our system uses appearances only and does not require region proposals.

Our approach takes inspiration from [23], where N-pairs loss has been proposed and achieves superior performance then contrastive divergence and triplets loss. However, their approach is formulated for patch based matching, while our approach uses N-pairs loss in the context of key points matching.

3 Correspondence Search

In correspondence search between a source image I_1 with query points p_n and a target image I_2, the goal is to precisely localize the matching point q_n for every query point in the target image I_2. Our framework for correspondence search is shown in Figure 1. It consists of a feature generator and a matching network.

The feature generator takes the source image I_1 and the target image I_2 as inputs and generates features F_1 and F_2. The matching network takes the features F_1 and F_2 as inputs and generates N similarity heatmaps for N query points.

This section is organised as follows: In Section 3.1 we describe the feature generator. Section 3.2 explains the matching network. Training and Inference are described in Sections 3.3 and 3.4, respectively.

3.1 Feature Generator

We refer both the Siamese branches as feature generator as shown in Figure 1. The feature generator consists of two copies of the batch-normalized Google-Net [24] up to layer 17 with shared parameters. Channel wise L2-normalization is applied to the output of each network. The feature generator takes source and target image, I_1 and I_2, as input, passes them through a series of convolution, max-pooling and inception layers, and generates features F_1, $F_2 \in R^{C\times W/P\times H/P}$, where W and H are the widths and heights of the input images and C is the dimensionality of the features, and P is the overall pooling.

3.2 Matching Network

We propose a matching network that efficiently generates a peaked similarity heat map for every query point over the target image. The heatmap representation enables N-pairs based metric learning [23] and eliminates the need of hard-negative sampling required with contrastive divergence or triplet based metric learning. In addition to that, with triplets or contrastive loss the network is not trained for the end goal of precise localization of the correct correspondence as during training pixels within a radius r of the ground-truth correspondence are taken as positives. Although this provides more positive pairs, it may negatively effect the precise localization for small semantic objects like hands of a person or beak of a bird.

During training, the network is trained to minimize the difference between the ground truth and the predicted similarity map. This eliminates the need for hard negative mining, as the network learns as part of its training procedure to generate dissimilar features for *all* the negatives of any query point.

The matching network is shown in Figure 2 and consists of a series of simple layers with no trainable parameters. It takes features F_1, F_2 and N query points as input, passes them through a series of simple layers, and generates N peaked similarity heatmaps for each query-point p_n in image I_1, as shown in Figure 2. In the following, we explain the operations performed by each layer in detail.

Crop layer. The crop layer takes features F_1, query points p_n and their lower resolution scalings $p'_n = \frac{p_n}{P}$ as inputs and generates a set of cropped features, $\{f_n\} \in R^{\mathcal{D}\times 1\times 1}$, by cropping F_1 around each p'_n.

Reshape R1. The Reshape R1 layer receives the set of cropped features $\{f_n\}$. It reshapes each $f_n \in R^{\mathcal{D}\times 1\times 1}$ into a row vector $f'_n \in R^{1\times \mathcal{D}}$ and generates a set of reshaped cropped features $\{f'_n\}$.

Copy. The copy layer generates the set $\{F_2^n\}$ consisting of N shallow copies of the $3D$ tensor F_2.

Reshape R2. The Reshape R2 layer receives the set of $3D$ tensors $\{F_2^n\}$ as inputs and reshapes each F_2 into a $2D$ matrix $\hat{F}_2 \in R^{\mathcal{D}\times WH}$.

Dot Product. The Dot product layer receives the set of reshaped cropped features $\{f'_n\}$ and N reshaped $2D$ matrices $\{\hat{F}_2^n\}$ as inputs and generates a set of $1D$ similarity vectors

$\{\hat{S}^n \in R^{1\times WH}\}$. Each $\hat{S}^n$ is computed as

$$\hat{S}^n = f'_n \odot \hat{F}_2 \ , \tag{1}$$

where $\odot$ is the dot product operation. The dot product layer efficiently computes the similarity of every cropped feature f_n with every feature in F_2.

Soft-Max. The soft-max layer normalizes the similarity score for every element $s_k \in \hat{S}^n$, where $k = \{1,...,WH\}$, as follows:

$$\hat{s}_k = \frac{\exp(s_k)}{\sum_{i=k}\exp(s_i)} \ . \tag{2}$$

Reshape R3. The Reshape R3 layer reshapes each $1D$ similarity vector in $\{\hat{S}^n\}$ into a $2D$ similarity map $\hat{S}^n_{x,y} \in R^{W\times H}$ and generates a set of $2D$ similarity maps $\{\hat{S}^n_{x,y}\}$.

All the above operations in combination with the feature generation are carried out in a single forward pass during training and inference.

3.3 Training

We denote training examples as $(I_1, I_2, \{p_n\}, \{q_n\})$. We generate a ground truth similarity map $S^n_{x,y} \in R^{W/P\times H/P}$, with $x = \{1,...,W/P\}$ and $y = \{1,...,H/P\}$, for every query point $p_n = (x_n, y_n)$ with ground truth correspondence $q_n = (x_q, y_q)$ as

$$\mathcal{S}^n_{x,y} = \begin{cases} 1, & \text{if } x = x_{q_n}, y = y_{q_n} \\ 0, & \text{otherwise} \end{cases} \ . \tag{3}$$

During training, given training examples $\{(I_1, I_2, \{p_n\}, \{q_n\}, \{S^n_{x,y}\})\}$, we minimize the binary cross-entropy error between every predicted $\hat{S}^n_{x,y}$ and ground-truth similarity map $S^n_{x,y}$ in an image pair as:

$$\min_w -\sum_{x,y} S^n_{x,y}\log(\hat{S}^n_{x,y}) + (1 - S^n_{x,y})(1 - \log(\hat{S}^n_{x,y})) \ , \tag{4}$$

where w are weights of the network. The weights are learned using back-propagation with the Adam optimizer.

3.4 Inference

During inference we our method predicts a matching correspondence q_n for every query point p_n as follows:

$$q_n = \arg\max_{x,y} \hat{S}^n_{x,y} \tag{5}$$

where q_n is the position of the peak in the predicted similarity map. We up-sample the generated similarity-map $\hat{S}^n_{x,y}$ using bilinear-sampling before localizing the peak.

4 Experiments

We evaluate the proposed method on the tasks of semantic keypoints matching and dense matching.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
NAMHOG [8]	72.9	73.6	31.5	52.2	37.9	71.7	71.6	34.7	26.7	48.7	28.3	34.0	50.5	61.9	26.7	51.7	66.9	48.2	47.8	59.0	52.5
PHMHOG [8]	78.3	76.8	48.5	46.7	45.9	72.5	72.1	47.9	49.0	84.0	37.2	46.5	51.3	72.7	38.4	53.6	67.2	50.9	60.0	63.4	60.3
LOMHOG [8]	73.3	74.4	54.4	50.9	49.6	73.8	72.9	63.6	46.1	79.8	42.5	48.0	68.3	66.3	42.1	62.1	65.2	57.1	64.4	58.0	62.5
UCN [8]	64.8	58.7	42.8	59.6	47.0	42.2	61.0	45.6	49.9	52.0	48.5	49.5	53.2	72.7	53.0	41.4	83.3	49.0	73.0	66.0	55.6
SCNet-A [20]	67.6	72.9	69.3	59.7	**74.5**	72.7	73.2	59.5	51.4	78.2	39.4	50.1	67.0	62.1	69.3	**68.5**	78.2	63.3	57.7	59.8	66.3
SCNet-AG [20]	83.9	81.4	70.6	62.5	60.6	81.3	81.2	59.5	53.1	81.2	**62.0**	58.7	65.5	73.3	51.2	58.3	60.0	69.3	61.5	**80.0**	69.7
SCNet-AG+ [20]	85.5	84.4	66.3	70.8	57.4	82.7	82.3	71.6	54.3	95.8	55.2	59.5	68.6	75.0	56.3	60.4	60.0	**73.7**	66.5	76.7	72.2
Ours	**88.9**	**94.5**	81.2	56.2	65.7	**91.0**	**83.2**	**76.1**	**68.6**	**97.9**	56.8	**75.5**	91.1	88.7	72.4	68.1	85.7	55.7	81.0	62.3	**81**

Table 1: PCK comparison with state-of-the-art on the 300 test pairs of the PF-Pascal dataset [9] across 20 object categories at fixed threshold $T = 0.1$. Our method outperform state-of-the-art significantly with a mean PCK of 81 .

Experiment Protocol. We use the Torch7 [6] framework for training and inference. Our experimental settings are as follows: We train the network from scratch without any pre-training with a learning rate of .00092 using a stair case decay of .95 applied every 73 epochs. We use the Adam optimizer [14] with $\beta_1 = 0.9$ and $\varepsilon = 0.1$. We train the network for 160 epochs for each dataset. During each training epoch we augment each pair with random scaling $s \in [0.5, 1.5]$ and rotation $r \in [-20, 20]$ to handle scale and rotation variations across pairs. Horizontal flipping is applied with probability .5.

4.1 Semantic Matching

We report results on PF-PASCAL [9], PF-Willow [9] and Pascal-Parts [30] datasets and following recent work [10] we train our method on the PF-PASCAL dataset only.

Evaluation Metric. Following recent work on semantic correspondence matching [5, 10], we use PCK as the evaluation metric. According to the metric, a predicted correspondence is correct if its euclidean distance in pixels to the ground-truth lies with in a threshold T. Different image resolutions are taken into account by normalizing the euclidean distance with the diagonal of the target image [5, 10]. We use PCK @ T for all experiments unless stated otherwise.

PF-Pascal dataset. The PF-Pascal dataset [9] consists of 1300 image pairs. The annotations are used from the PASCAL-Berkeley dataset [2, 3]. For fair comparison with recent work, we use 700 training pairs, 300 validation pairs and 300 test pairs from [10] respectively. We compare the PCK achieved by our method at fixed threshold $T = 0.1$ to recent work in Table 1 on the 300 PF-Pascal [9] test pairs. The comparison is done across 20 object categories. Our method outperforms the state-of-the-art significantly with a mean PCK of 81.

PF-Willow dataset. We evaluate the transferability of the features learned by our method on the PF-Willow dataset [9]. The PF-Willow dataset consists of 10 objects sub-classes with 10 keypoint annotations for each image. We evaluate the performance using the PCK at thresholds of 0.05, 0.1 and 0.15 respectively and comapare it with recent state-of-the-art. The comparison is shown in Table 2 . The PCK is averaged over the 10 object sub-classes. Our method outperforms SCNet for PCK@.05 and PCK@0.1 and achieves competitive performance for PCK@.15 based on raw point correspondences computed using appearance only.

Pascal Parts Dataset. We also evaluate transferability on the sampled Pascal-Parts dataset [30] using PCK at a fixed threshold of 0.05 and compare against recent state-of-the-art. The PCK is averaged over all the classes. The results are shown in Table 3. Our method outperforms with a PCK of .33

Method	PCK @ 0.05	PCK @ 0.1	PCK @ 0.15
SIFTFlow [13]	0.247	0.380	0.504
DAISY w/SF [27]	0.324	0.456	0.555
FCSS w/SF [13]	0.354	0.532	0.681
FCSS w/PF [13]	0.295	0.584	0.715
LOMHOG [9]	0.284	0.568	0.682
UCN [8]	0.291	0.417	0.513
SCNet-A [10]	0.390	0.725	0.873
SCNet-AG [10]	0.394	0.721	**0.871**
SCNet-AG+ [10]	0.386	0.704	0.853
Ours	**0.520**	**0.780**	0.833

Table 2: PCK comparison with state-of-the-art on the PF-Willow [9] dataset @ thresholds of 0.05, 0.1 and 0.15 respectively. The PCK is averaged over the 10 sub-classes from the dataset.

Method	PCK @ 0.05
NAMHOG [9]	0.13
PHMHOG [9]	0.17
LOMHOG [9]	0.17
FCSS w/SF [13]	0.28
FCSS w/PF [13]	0.29
SCNet-A [10]	0.17
SCNet-AG [10]	0.17
SCNet-AG+ [10]	0.18
Ours	**0.33**

Table 3: PCK comparison with state-of-the-art on the sampled Pascal-Parts [30] dataset at a fixed threshold of 0.05.

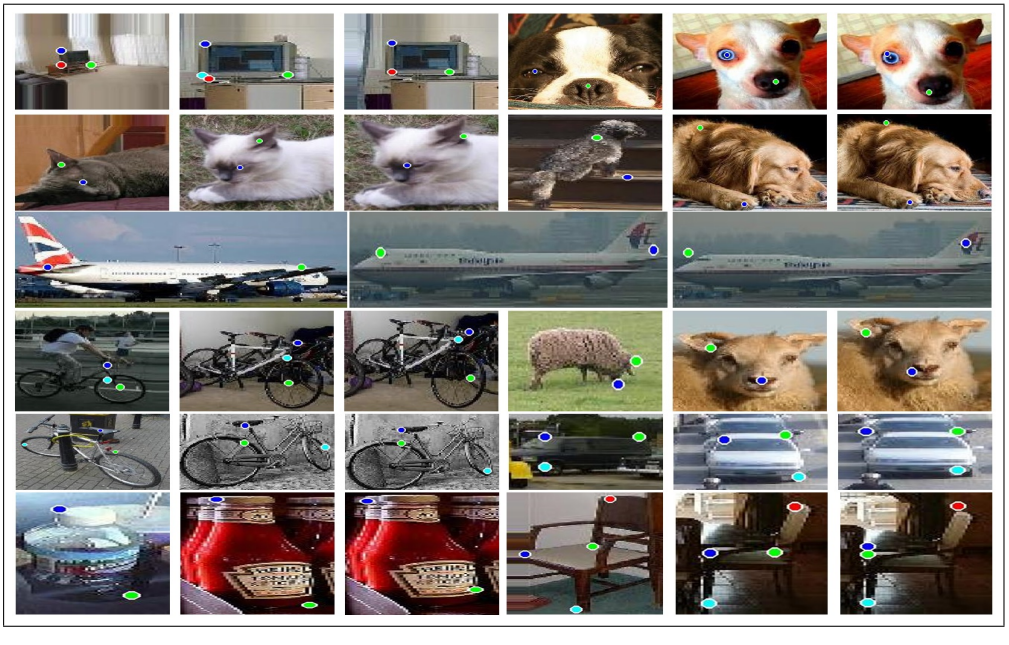

Figure 3: Qualitative results for keypoints transfer on the PF-Pascal ([9]) dataset. (Left) Source image with query points. (Middle) Target image with keypoint transfers through correspondence matching. (Right) Target image with ground truth correspondences.

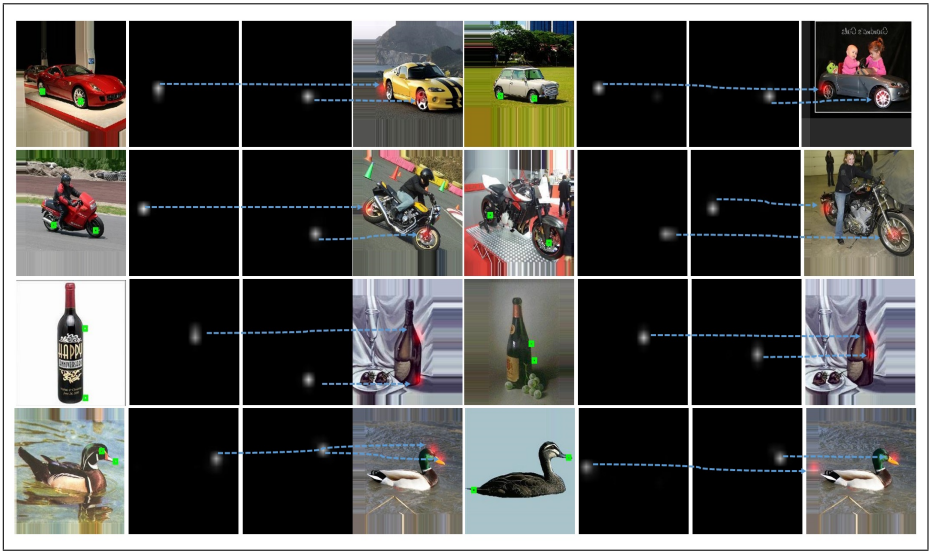

Figure 4: Qualitative results on the PF-Willow [9] dataset with similarity heatmaps. Each row shows two examples. Left image is the source image with query points shown by green rectangles. Predicted similarity heatmaps are shown in the middle images. On the right target image is shown with overlaid heatmaps.

Qualitative Results PF-Pascal. We show qualitative results for keypoints transfer on the PF-Pascal dataset in Figure 3 (a). Left image is the source image with query points. Middle image is the target image with keypoint transfers through correspondence matching. Right image is the target image with ground truth correspondences.

We show qualitative results with similarity heatmaps generated by the matching network on the PF-Willow [9] dataset in Figure 4. Results show that the matching network generates peaky similarity heatmaps that are used to precisely localize the correspondence. Results from rows 1 and 2 show that our method handles rotation, scale variations and parts symmetry like wheels of the car and bikes very well without using any spatial transformer layers or explicit geometry.

4.2 Dense Matching

For dense correspondences, we evaluate on the KITTI-Flow 2015 [19] and MPI Sintel [4] benchmarks. For a fair comparison with [5] we use the train/test split of the training set employed in [5] for both KITTI-Flow and MPI-Sintel. We follow the performance measure of PCK @ $10px$ from deep matching that is used in [5] for evaluation. Here, a predicted correspondence is considered as correctly matched if it lies within 10 pixels of the ground-truth.

We use image resolutions of 1242×375 and 1024×436 for KITTI-Flow and MPI-Sintel during training and inference, respectively. We randomly sample 1000 query-match points for each image pair during each training epoch.

The results for dense correspondences are shown in Table 4. Our approach significantly

Method	Kitti-Flow	MPI-Sintel
SIFT-NN[17]	48.9	68.4
HOG-NN[7]	53.7	71.2
SIFT-flow[17]	67.3	89.0
DaisyFF[27]	79.6	87.3
DSP[12]	58.0	85.3
DM[20]	85.6	89.2
UCN-HN[5]	86.5	91.5
UCN-HN-S[5]	83.4	90.7
Ours	**91.8**	**92.0**

Table 4: Dense Correspondence matching performance for PCK@10 on the Kitti-Flow and MPI-Sintel test sets. Our raw-correspondences outperform all previous state-of-the-art.

outperforms all previous state-of-the-art methods using just raw correspondences from our framework without any post-processing steps. In particular, our approach also achieves better performance than DAISY [27], DSP [12], and DM [20], which apply global optimization as a post-processing step to achieve more precise correspondences. We also do not use spatial transformer layers [11] and therefore do not experience overfitting with less training examples for the KITTI-Flow [19] dataset, as reported in [5].

5 Conclusion

In this work, we have proposed a novel matching network that efficiently generates peaked similarity heatmaps for every query point over the target image. The matching network can be appended to any base network e,g VGG or GoogleNet to make it end-to-end trainable for precise location of keypoints. The heatmap representation enables N-pairs based metric learning and achieves superior performance on the PF-Pascal [9], PF-Willow [9], Pascal-Parts [30], the KITTI-Flow 2015 [19] and MPI Sintel [4] datasets.

6 Acknowledgments

This project was funded, in parts, by ERC Consolidator Grant project DeeViSe (ERC-2017-CoG-773161) and by Amazon Academic Research Award grant "End-to-end Deep Learning for Human Pose Estimation in Video". Juergen Gall was supported by the ERC Starting Grant ARCA.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *CVIU*, 2008.

[2] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.

[3] J. Bromley, I. Guyon, Y. Lecun, E. SÃd'ckinger, and R. Shah. Signature verification using a Siamese time delay neural network. In *NIPS*, 1994.

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[5] CB. Choy, J. Gwak, S. Savarese, and M. Chandraker. Universal Correspondence Network. In *NIPS*, 2016.

[6] R. Collobert, K. Kavukcuoglu, and C. Farabe. Torch7: A matlab-like environment for machine learning. 2011.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] M. Everingham, L. Van Gool, C. K. I., Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. Technical report, 2011.

[9] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. In *CVPR*, 2016.

[10] K. Han, R.S.R., B. Ham, KK. Wong, M. Cho, C. Schmid, and J. Ponce. Scnet: Learning semantic correspondence. In *ICCV*, 2017.

[11] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. In *NIPS*, 2015.

[12] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspon- dences. In *CVPR*, 2013.

[13] S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn. Fully convolutional self-similarity for dense semantic corre- spondence. In *CVPR*, 2017.

[14] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. 2014.

[15] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 2011.

[16] J. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *NIPS*, 2014.

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[18] W. Luo, A. G. Schwing, and R. Urtasun. Efficient Deep Learning for Stereo Matching. In *CVPR*, 2016.

[19] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[20] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. DeepMatching: Hierarchical Deformable Dense Matching. *IJCV*, 2016.

[21] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *CVPR*, 2015.

[22] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[23] K. Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *NIPS*, 2016.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *CVPR*, 2015.

[25] E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *PAMI*, 2010.

[26] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014.

[27] H. Yang, W. Y. Lin, and J. Lu. DAISY filter flow: A generalized approach to discrete dense correspondences. In *CVPR*, 2014.

[28] S. Zagoruyko and N. Komodakis. Learning to Compare Image Patches via Convolutional Neural Networks. In *CVPR*, 2015.

[29] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a CNN. In *CVPR*, 2015.

[30] T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros. Flowweb. In *CVPR*, 2015.