AutoScaler: Scale-Attention Networks for Visual Correspondence

Shenlong Wang[1]
slwang@cs.toronto.edu
Linjie Luo[2]
linjie.luo@snap.com
Ning Zhang[2]
ning.zhang@snapchat.com
Jia Li[3]
lijiali@cs.stanford.edu

[1] University of Toronto
Toronto, Canada
[2] Snap, Inc.
Venice, USA
[3] Google
Mountain View, USA

Abstract

Finding visual correspondence between local features is key to many computer vision problems. While defining features with larger contextual scales usually implies greater discriminativeness, it could also lead to less spatial accuracy of the features. We propose AutoScaler, a scale-attention network to explicitly optimize this trade-off in visual correspondence tasks. Our architecture consists of a weight-sharing feature network to compute multi-scale feature maps and an attention network to combine them optimally in the scale space. This allows our network to have adaptive sizes of equivalent receptive field over different scales of the input. The entire network can be trained end-to-end in a Siamese framework for visual correspondence tasks. Using the latest off-the-shelf architecture for the feature network, our method achieves competitive results compared to state-of-the-art methods on challenging optical flow and semantic matching benchmarks, including Sintel, KITTI and CUB-2011. We also show that our attention network alone can be applied to existing hand-crafted feature descriptors (e.g Daisy) and improve their performance on visual correspondence tasks. Finally, we illustrate how the scale-attention maps generated from the attention network are visually interpretable.

1 Introduction

Finding correspondences between local features in multiple related images is a fundamental problem in computer vision. It is crucial for a plethora of applications, including optical flow [32, 37, 46], structure-from-motion [1], visual SLAM [26, 35, 36], stereo matching [31, 53], non-rigid 3D reconstruction [14] as well as video segmentation [18].

Central to the correspondence problem is the design of feature descriptors that need to be resilient to lighting change and different object poses and scales. To select the characteristic scales, many hand-crafted descriptors analyze feature saliency in a scale space formed by applying heuristic image processing operators on different scales of the images. The resultant descriptors are extracted from either one [8, 30] or many [22] of these scales. However, due to their heuristic nature, the scale analyses of these hand-crafted descriptors are limited

© 2017. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

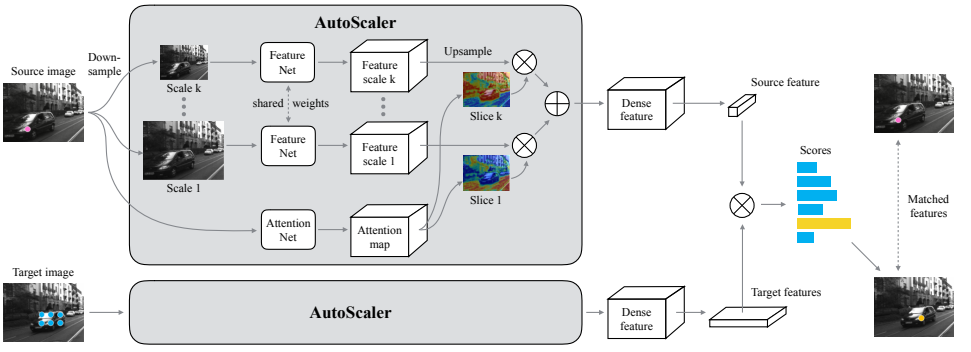


Figure 1: The architecture of AutoScaler. AutoScaler consists of the feature network and the attention network. The weight-sharing feature network extracts feature maps from the input image at multiple scales (two are shown for simplicity). The attention network computes attention maps for each scale and optimally combines the multi-scale feature maps. The entire network can be trained for the task end-to-end in a Siamese framework.

to a sparse set of image locations with special structures, such as blobs, corners and high contrast regions [33]. To compute dense correspondences using these descriptors, one needs to impose smoothness prior to regularize the correspondence map from the sparse matches, which often experiences loss in accuracy [7, 15, 41].

Recently, with the growing availability of synthetic and real-world datasets, learning-based approaches have been applied to compute similarity metrics for visual correspondence problems. These approaches include boosting [44], random forest [46], convex optimization [40] and most notably, convolutional neural network (CNN) [20, 25, 29, 52, 53]. Compared to traditional hand-crafted descriptors, CNN is powerful in that it can learn discriminative robust features from large amount of training data. However, in order to be robust against scale change, these features often resort to larger receptive fields through successive pooling [27, 39], large strides [27], dilated convolution [51] and multi-scale aggregation [28, 42]. As a result, the spatial accuracy of the resultant features is compromised. To address this problem, a few techniques are introduced such as spatial transformer network in [13] to normalize object pose change and bi-scale ensemble model in [12] to combine matching scores from two scales. Nevertheless, it remains unclear how to optimally combine features from different scales based on the analysis in the scale space for visual correspondence problems.

One effective tool for this kind of analysis is the attention mechanism which has been widely studied for numerous computer vision tasks [2, 17, 34, 50]. In particular, [10] proposed an attention model that combines the score maps from multiple scales for semantic image segmentation. Despite its success to robustly segment semantic objects with different sizes, the proposed attention model is not designed to optimize for the spatial accuracy of visual correspondences. Also, as a late-fusion step that combines multi-scale score maps, the attention model cannot be used to derive a scale-resilient feature space to provide the similarity measurement for visual correspondence tasks.

In this paper, we propose the *AutoScaler*, a scale-attention network to optimally combine feature maps from different scales for visual correspondence tasks. Our key insight is that the trade-off between the spatial accuracy and the discriminative contextual scales of local features can be explicitly optimized via a scale-attention network to improve visual correspondence accuracy. Intuitively, in texture-rich area, the network weighs more on the

fine-scale features to ensure correspondence accuracy while in area with less texture, the network seeks for the features at larger scales for more discriminative contextual information.

Our AutoScaler network consists of a weight-sharing *feature network* to compute multi-scale feature maps and an *attention network* to combine them optimally in the scale space (Fig. 1). By sharing weights across multi-scale feature network, our system can handle large scale changes. The attention network alone can be applied to any existing handcrafted descriptors (e.g. Daisy [43]) and improve their performance in visual correspondence tasks. The full network can be trained end-to-end in a Siamese framework without explicit supervision on scale-attention. We demonstrate the effectiveness of the proposed method over optical flow, semantic correspondence tasks and find it compared favorably with the state-of-the-art methods. Our method is also able to generate visually interpretable scale attention maps. In sum, our main contributions are:

- A scale-attention network that optimally combines features from multiple scales in terms of contextual discriminativeness and spatial accuracy.
- The proposed scale-attention network can be a general performance-improver for existing hand-crafted descriptors (e.g. Daisy) on visual correspondence tasks.
- Using the off-the-shelf architecture for feature network, our simple approach can achieve competitive results in challenging visual correspondence benchmarks, especially in terms of fine-scale correspondence accuracy.
- The resultant scale-attention maps are visually interpretable.

2 Method

In this section, we will elaborate our formulation for the visual correspondence tasks of interest as well as implementation details to train the underlying models.

2.1 Formulation

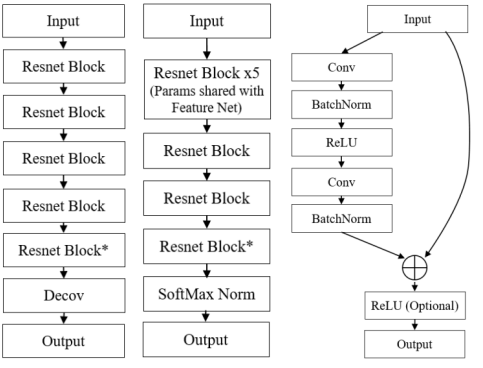
We are interested in finding distinctive local correspondence given a pair of related images I and I'. A typical correspondence problem tackles the problem by computing a similarity measure $s(\mathbf{p}_i, \mathbf{q}_j)$ between a given position $\mathbf{p}_i$ from the source image and its all possible matching candidates $\mathcal{N}_\mathbf{p} = \{\mathbf{q}_{j|j=1,\ldots,N}\}$ in the target image; and choose the most similar sample. The candidates set $\mathcal{N}_\mathbf{p}$ varies depending on tasks. For instance, we search points along the epipolar line for stereo matching, within a 2D neighborhood for optical flow, and within the whole image for semantic matching. Computation of the similarity measure is typically done by measuring the cost associated with local features located at $\mathbf{p}$ and $\mathbf{q}$.

Our general matching scheme is a siamese architecture, where two branch process the source and target images separately with sharing parameters. In the feature extraction stage, each image is passed into a scale-attention network, called AutoScaler. AutoScaler firstly generates a pyramid of input images across different scales as shown in Fig. 1. Each scale is then passed into a CNN feature net and produces a feature map. The parameters of CNN feature net are shared, which makes same input image across multiple scales generate correlated features. Each scale's output is upsampled into the original size of the input image, in order to ensure that the feature maps across scales have the same size. In the meantime, an attention network is introduced to predict a dense weight map for each point across all

the scales. The final dense feature is then computed through a weighted sum across all the scales. Fig. 1 depicts the whole process of the dense scale-aware feature computation.

In the matching stage, after we get the dense feature maps, for each point that we are interested in from the source image, we extract its corresponding source feature as well as the features from all the candidate points in the target image. Then an inner-product layer is used to generate the similarity between the source feature and the target features. Point with highest similarity is picked as a corresponding point in the target image.

Architecture Both the attention network and feature network have a fully convolutional network architecture with short-cut connections to generate pixel-wise feature/score map. The CNN feature net contains five ResNet [23] blocks, each of which contains a `conv-batchnorm-relu-conv-batchnorm` structure, followed by a short-cut element-wise sum and a final `relu` layer. The last `relu` unit in the feature net is removed for non-sparse feature. The attention map contains 9 ResNet blocks, with top five sharing parameters with the feature net. The final output is passed through a pixel-wise soft-max layer to ensure the attention map is between $[0, 1]$ with interpretability. We do not use any pooling or strided convolution to ensure that feature maps preserve sub-pixel information. The receptive field size is equal to 23×23 for a single scale feature net. In our experiments, the number of filters is 64 (sintel and CUB) or 128 (KITTI). We use 64 filters as example to describe our method in the following.



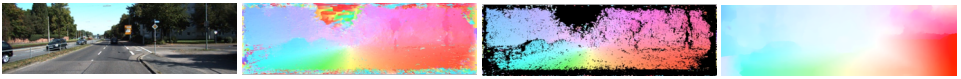
(a) Feature net (b) Attention net (c) One ResNet block

2.2 Training

Training data We use the ground-truth pixel-wise correspondence from the dataset to train our neural network. For each pair of images we pick a subset of corresponding pixel pairs. For each pair in the target images, we randomly sample some pixels over all the candidates within the searching range of ground-truth as negative points. This negative sampling is motivated by the fact that points nearby the ground-truth are most likely to be false positive. In practice we choose 200 negative samples and this results in 201 candidates for each pair with one ground-truth for each source point. We extract features from these points, which results in 64-dimensional source vector and 64×201-dimensional target feature.

Loss Through computing the inner product between the source feature and all the columns in the target feature, we have a 201-dimensional score vector describing the confidence of each possible candidates to be a correspondent point. Intuitively, we expect the GT correspondent to have higher score while others have lower score. Thus we define our objective to be the cross-entropy loss between the GT correspondence and the matching score, normalized by softmax and minimize the loss with respect to the parameters of our neural networks.

Optimization We train our network using stochastic gradient descent with Nesterov momentum. The momentum is set to be 0.9 and the initial learning rate is set to be 0.002. A learning rate policy is set to reduce the learning rate by a factor of 5 for every 50K iterations.



(a) Input image (b) Initial noisy estimation (c) Outlier removal (d) Final flow

Figure 2: The dense optical flow pipeline. From the input image (a), an initial noisy flow is estimated (b), followed by outlier removal (c) and interpolation using [37] (d). (Sec. 3.1.1)

2.3 Discussions

Receptive field size The advantage of the proposed model is its content-aware receptive field size. The attention model adjusts the receptive field according to the image content through weighting each scale. Given an image pyramid with smallest scale $\times 4$, our algorithm is able to produce a maximum receptive field with $23 \times 4 = 132$. This approach introduces more context into the matching scheme. It would greatly help resolve ambiguity because of repetitive, smooth textures, or matching along edges. On the other hand, in regions with unique structures, it learns to focus on finer scales with a smaller receptive field, excluding unnecessary context. See Fig. 4 for the visualization of the scale-attention maps.

Extensions to hand-crafted features Our AutoScalar model can be extended to hand-crafted features, such as SIFT and DAISY. To be specific, instead of using a neural network to compute multi-scale features, we can generate multi-scale features through changing the hyper-parameters of SIFT and DAISY. Then an attention net is trained to combine these multi-scale features in a content-aware manner towards a better performance.

3 Experiments

This section presents the result of the proposed scale-attention network on both geometric matching and semantic matching tasks. For geometric matching task, we select the challenging optical flow benchmarks MPI-Sintel [9] and KITTI [16] for evaluations. The semantic matching experiment is conducted over the Caltech-UCSD Birds 2011 dataset [45]. We compare with the state-of-the-art systems, as well as visualize the generated attention maps and discuss their interpretability as shown in Fig. 4.

3.1 Evaluation on Optical Flow

3.1.1 MPI-Sintel Optical Flow Benchmark

We first evaluate our method on the challenging MPI-Sintel optical flow benchmark [9], which consists of more than 1200 pairs of training images and 1500 pairs of testing images. It is a synthetic dataset with extremely large motion from both cameras and objects with various appearance changes due to motion blur, illumination and non-rigid deformation. The benchmark error metric is end-point-error (EPE), which is the average euclidean distance between the flow fields. We refer to EPE-matched and EPE-unmatched as average end-point-error over regions that remain visible in adjacent frames and average end-point-error over regions that are visible only in one of two adjacent frames. And EPE-all is the end-point-error over all the pixels.

During training, we split the 22 training sequences into training (1-16) and validation (17-22). For each pair of images, we randomly sampled 10K local corresponding pairs, and for

Method	EPE-m	EPE-u	EPE-a
DCFlow [49]	2.283	28.228	5.119
FFCNN [6]	2.303	30.313	5.363
MRFlow [53]	2.818	26.235	5.376
DeepDisFlow [19]	2.623	31.042	5.728
FullFlow [11]	2.684	30.793	5.895
PatchCollider [46]	2.938	31.309	6.040
EpicFlow [37]	3.060	32.564	6.285
DeepFlow2 [47]	3.093	38.166	6.928
Ours	2.569	34.656	6.076

Sintel Results

Method	Fl-bg	Fl-fg	Fl-all
FlowNet2 [24]	10.75%	8.75%	10.41%
SDF† [8]	8.61 %	26.69 %	11.62 %
SOF† [38]	14.63 %	27.73 %	16.81 %
CNN-HPM[8]	18.33 %	24.96 %	19.44%
FullFlow [11]	23.09 %	30.11 %	24.26 %
EpicFlow [37]	25.81 %	33.56 %	27.10 %
DeepFlow2 [47]	27.96 %	35.28 %	29.18 %
PatchCollider [46]	30.60 %	33.09 %	31.01 %
Ours	21.85 %	31.62 %	25.64 %

KITTI Results

Table 1: Optical flow results on both Sintel and KITTI benchmarks. Despite the simplicity of our method, we achieve competitive results on Sintel and KITTI. See Sec. 3.1 for detail.

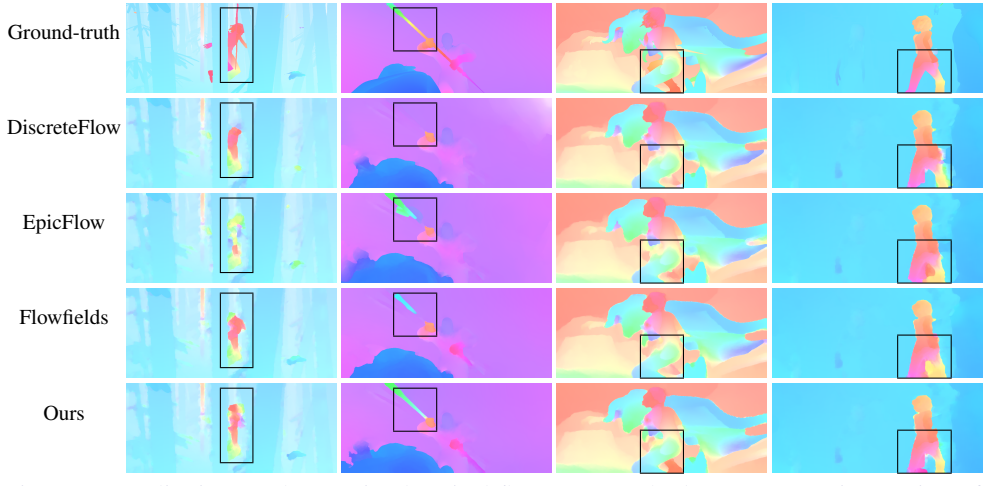


Figure 3: Qualitative results on Sintel optical flow. Our method recovers precise motion of fine structures, like the butterfly, pole and legs as highlighted in boxes. See Sec. 3.1.1.

each pair, we randomly selected 200 negative samples within the motion range $[-210, 200]$.

During testing, we use a simple pipeline to compute the dense flow. First, We compute dense features for both source and target images. For each feature from the source image, we compute the inner-product over all the local features from the target image within the motion range limit $[-240, 240] \times [-240, 240]$ and pick the one of the highest score. This produces an initial estimation of dense flow with outliers. We then remove outliers through forward-backward consistency check: for each pixel $\mathbf{p}$, we check the condition $\|\mathbf{u}_{\text{backward}}(\mathbf{p}+\mathbf{u}_{\text{forward}}(\mathbf{p}))+\mathbf{u}_{\text{forward}}(\mathbf{p})\| \leq t$, where $\mathbf{u}_{\text{backward}}$ is the estimated backward optical flow, $\mathbf{u}_{\text{forward}}$ is the forward optical flow field and t is the threshold we remove outlier motion estimations. In practice $t = 3$ is used. After removing outliers, we interpolate the missing pixels using Epicflow [37]. Fig. 2 illustrates our pipeline of dense optical flow.

Quantitative Results We submit our results to Sintel benchmark and compare it against the top-ranked published systems. We focus on the final benchmark, which is more challenging due to the presence of motion blur and various shading and reflectance changes. Table. 3.1.1 shows the quantitative results against the competing systems. Our method achieves third best on the EPE-matched metric, and comparable on EPE-all metric against all com-

Dataset	Daisy Concat	Daisy	CNN [31]	Single	Concat×2	Sum×2	Ours×2	Ours×4
Sintel	56.79%	78.30%	86.02%	86.95%	87.65%	87.92%	89.12%	**91.84%**
KITTI	73.63%	75.67%	90.10%	90.07%	88.04%	89.60%	**92.06%**	91.78%

Table 2: Top-1 accuracies of variant architectures on Sintel and KITTI (See Sec. 3.1.3).

peting systems. Our relative large EPE-unmatched number is due to the simplicity of our interpolation scheme using Epicflow compared to the time-consuming comprehensive MRF post-processing step to propagate the estimated flow to occluded regions as employed by many competing systems such as Flowfields [4], DiscreteFlow [19, 32] and Fullflow [11]. In principle, this heavy MRF post-processing step can also be used in our system to further boost EPE-unmatched performance if its significant overhead in running-time is acceptable as it is the bottleneck for many methods. Our method takes 0.5 seconds for computing features, 2 minutes for initial matching and 2.5 seconds for Epicflow interpolation on Sintel. All timings are done on a machine with 3.2GHz CPU and M6000 GPU.

Qualitative Results Fig. 3 demonstrates more qualitative results for visual comparison. Thanks to the scale-attention scheme, our method has the best capability in capturing small objects with large motion, as shown in the figure. This is because our method has both large receptive field and sub-pixel accuracy.

3.1.2 KITTI Optical Flow Benchmark

We also report the benchmarking result over KITTI Optical Flow 2015 dataset [16]. This benchmark includes 200 image pairs for training and 200 image pairs for testing. During training, we separate the training dataset into 160 pairs as train and 40 pairs as validation. Following the similar experiment configuration in Sec. 3.1.1, we sample 10k local correspondences from each image pair and for each pair 200 negative samples. During testing, we follow similar pipeline described in Sec. 3.1.1 to generate dense optical flow with our network as shown in Fig. 2.

Quantitative results We submit our results to KITTI optical flow benchmark. The results are shown in Table. 3.1.1. The metrics for KITTI benchmark 'Fl-bg', 'Fl-fg' and 'Fl-all' represent the outlier percentage on background pixels, foreground pixels and all pixels respectively. Note that KITTI is a dataset captured in a special driving scenario, where the motion is mainly due to the ego-motion of the camera plus rigid motion of the cars in the scene. Thus dense flow approaches that exploit the semantics of the scene objects as well as the epipolar constraint would achieve significant improvement [3, 38]. Apart from those methods, our approach achieves competitive results against other methods which utilize generic matching techniques.

3.1.3 Evaluation of Architecture Design

We compare Top-1 matching accuracy of our proposed architecture to many of its variants on both Sintel and KITTI validation sets, as shown in Table. 2. All the variant architectures are trained under the same multi-class Siamese configuration with softmax loss. "Daisy Concat" and "Daisy" refer to the variants where feature network is replaced with Daisy descriptor [43] on two scales and then combined using simple concatenation and our attention network respectively. "CNN" refers to a competing architecture [31] which includes CNN-31×31: a nine-layer fully convolutional network. Similar to our approach, [31] also adopts softmax

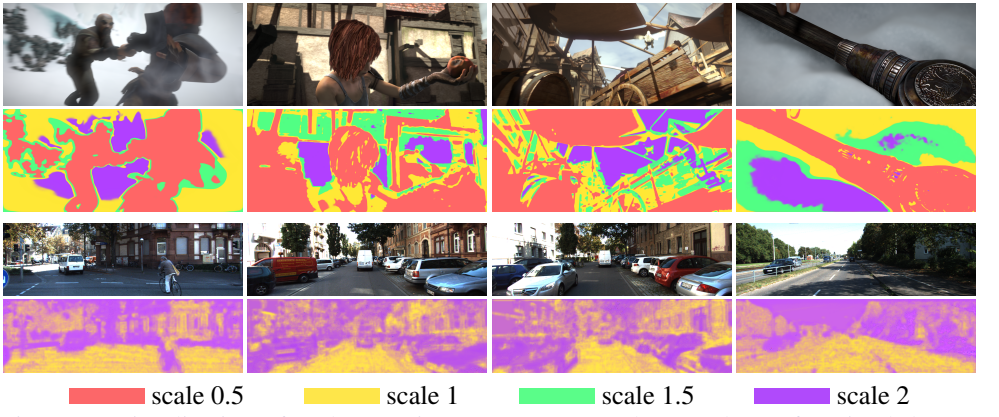


Figure 4: Visualization of scale-attention maps. Four scales are shown for Sintel dataset (top) and two for KITTI (bottom). Scale values indicate down-sample factors of the input image to the feature network (0.5 being the up-sampled finest scale and 2 the down-sampled coarsest). Note that the attention network weighs more on fine scale for texture-rich regions and gradually moves to larger scales in regions with less texture. See Sec. 3.1.4 for details.

loss for training and the architecture does not include pooling or stride convolution. "Single" refers to our proposed architecture with one single scale. "Concat×2" and "Sum×2" refers to our proposed architecture with two scales and then combined using simple concatenation and summation respectively. "Ours×K" is our proposed architecture with K scales. In this experiment we compare the performance between two and four scales.

As shown in Table 2, our proposed architecture outperforms all variant architectures. Especially, with the attention mechanism, the matching performance is better than simply concatenating or summing two scales. It is also worth noting that our attention architecture can be applied to improve hand-crafted descriptors like Daisy [43] for the visual correspondence, as we observe noticeable performance improvements with attention mechanism enabled versus simple concatenation of multi-scale features. Moreover, for Sintel dataset, our four-scale variant outperforms the two-scale one while for KITTI their performances are similar. This seems to do with the dataset bias: KITTI has fewer textureless regions that take advantage of features at larger scales.

3.1.4 Visual Interpretation of Scale Attention Maps

To visualize our attention maps, we pick representative frames from Sintel and KITTI datasets and colorize their attention weights at different scales as shown in Fig. 4. Note that our trained attention network weighs more on fine scales for texture-rich regions (e.g. foreground subjects, roads) and gradually moves to larger scales in regions with less texture (e.g. background objects, sky). This highlights our attention network's ability to find optimal trade-off between discriminative contextual scale and spatial accuracy for visual correspondence.

3.2 Evaluation on Semantic Matching

Unlike geometric matching tasks, such as optical flow, the semantic matching aims at finding correspondence that represents coherent semantic meanings, regardless whether these keypoints are similar in appearance, *etc.*. We perform the semantic matching experiments on the

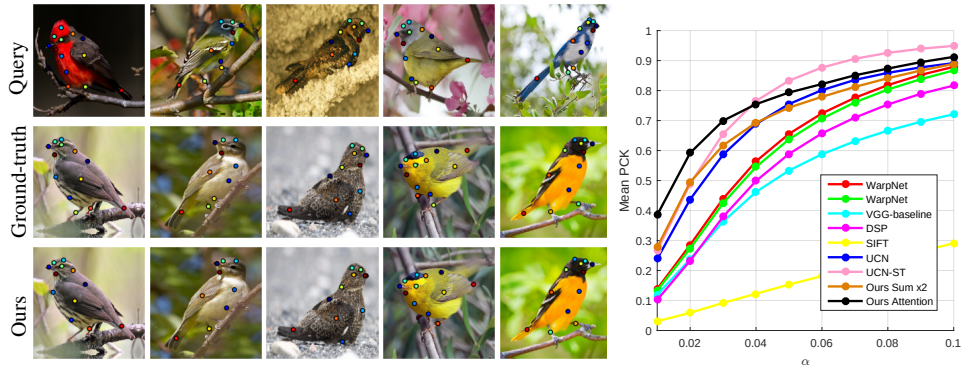


Figure 5: Qualitative and quantitative results on CUB semantic matching. Our method is able to capture semantic meaningful matching across species and poses, with better fine-scale accuracy compared to competing methods. See Sec. 3.2 for more details.

CUB-200-2011 dataset, which contains 11788 images of 200 bird categories, with 15 parts annotated. We follow the experiment configuration of [25], which utilizes the training set to extract training pairs and 5000 pairs images from the validation set as testing pairs. We crop each image with the bounding box of birds as pre-processing step. For each training iteration, we randomly pick two image pairs and use all the corresponding keypoints between them as positive samples, and select random negative samples over the whole target images.

Metric We evaluate the accuracy of matches with the percentage of correct keypoints (PCK@α). A match is considered as correct if it lies with αL pixels of the ground-truth correspondence, where L is the mean diagonal size of the image pairs. We follow the configuration of [25] and discard all the invisible keypoints when computing the metric. We strictly follow UCN and WarpNet's setting without the post-processing in Sec. 3.1.1.

Quantitative result We compared against the recent state-of-the-arts algorithms on CUB matching dataset, namely WarpNet [25], Universal correspondence network [13], and DSP [21], along with two widely used features including VGGnet [39] and SIFT [30]. Fig. 5 depicts the PCK metric along different threshold α. From this figure we can see that our method outperforms all the competing algorithms when α is small, which suggests the highest fine-scale accuracy. When the threshold α becomes large, our method ranks second, following UCN [13]. This suggests that AutoScaler better captures finer details while in the meantime performs competitively in reasoning the semantic meaning of the local parts. Fig. 5 also shows the examples of the qualitative matching results. Our method performs well in most cases across various poses, species and scales. Most failure cases are due to the ambiguity in left and right feet, which could be improved by a global structure prior.

4 Conclusion

We propose AutoScaler, a scale-attention network that optimally combines dense feature maps from multiple scales for contextual discriminativeness and spatial accuracy. We show that our simple approach can achieve competitive results in challenging visual correspondence benchmarks. The scale-attention network can be used as performance-improver for existing handcrafted descriptors and provide visually-interpretable scale-attention maps.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building rome in a day. In *ICCV*, 2009.

[2] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. In *ICLR*, 2015.

[3] Min Bai, Wenjie Luo, Kaustav Kundu, and Raquel Urtasun. Exploiting semantic information and deep matching for optical flow. In *ECCV*, 2016.

[4] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *ICCV*, 2015.

[5] Christian Bailer, Kiran Varanasi, and Didier Stricker. Cnn based patch matching for optical flow with thresholded hinge loss. *arXiv*, 2016.

[6] Christian Bailer, Kiran Varanasi, and Didier Stricker. Cnn-based patch matching for optical flow with thresholded hinge embedding loss. In *CVPR*, 2017.

[7] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. In *ECCV*. 2010.

[8] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[9] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[10] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. *CVPR*, 2016.

[11] Qifeng Chen and Vladlen Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *CVPR*, 2016.

[12] Zhuoyuan Chen, Xun Sun, Liang Wang, Yinan Yu, and Chang Huang. A deep visual correspondence embedding model for stereo matching costs. In *ICCV*, 2015.

[13] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *NIPS*, 2016.

[14] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: real-time performance capture of challenging scenes. *SIGGRAPH*, 2016.

[15] B. Drayer and T. Brox. Combinatorial regularization of descriptor matching for optical flow estimation. In *BMVC*, 2015.

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[17] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *ICML*. 2015.

[18] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010.

[19] Fatma Güney and Andreas Geiger. Deep discrete flow. In *ACCV*, 2016.

[20] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*, 2015.

[21] Tatsuya Harada, Yoshitaka Ushiku, Yuya Yamashita, and Yasuo Kuniyoshi. Discriminative spatial pyramid. In *CVPR*, 2011.

[22] Tal Hassner, Shay Filosof, Viki Mayzels, and Lihi Zelnik-Manor. Sifting through scales. *PAMI*, 2016.

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

[24] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *CVPR*, 2017.

[25] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. *CVPR*, 2016.

[26] Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *IV*, 2010.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[29] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014.

[30] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[31] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *CVPR*, 2016.

[32] Moritz Menze, Christian Heipke, and Andreas Geiger. Discrete optimization for optical flow. In *GCPR*, 2015.

[33] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 2005.

[34] Volodymyr Mnih, Nicolas Heess, Alex Graves, and koray kavukcuoglu. Recurrent models of visual attention. In *NIPS*. 2014.

[35] Raul Mur-Artal, JMM Montiel, and Juan D Tardós. Orb-slam: a versatile and accurate monocular slam system. *IEEE Trans on Robotics*, 2015.

[36] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011.

[37] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015.

[38] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J. Black. Optical flow with semantic segmentation and localized layers. In *CVPR*, 2016.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[40] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1573–1585, 2014.

[41] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.

[42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[43] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. In *PAMI*, 2010.

[44] Tomasz Trzcinski, Mario Christoudias, Vincent Lepetit, and Pascal Fua. Learning image descriptors with the boosting-trick. In *Advances in neural information processing systems*, pages 269–277, 2012.

[45] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[46] Shenlong Wang, Sean Ryan Fanello, Christoph Rhemann, Shahram Izadi, and Pushmeet Kohli. The global patch collider. In *CVPR*, 2016.

[47] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013.

[48] Jonas Wulff, Laura Sevilla-Lara, and Michael J. Black. Optical flow in mostly rigid scenes. In *CVPR*, 2017.

[49] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *CVPR*, 2017.

[50] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutidinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*. 2015.

[51] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016.

[52] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015.

[53] Jure Žbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR*, 2015.