

AutoScaler: Scale-Attention Networks for Visual Correspondence

Shenlong Wang

University of Toronto

slwang@cs.toronto.edu

Linjie Luo

Snap Inc.

linjie.luo@snap.com

Ning Zhang

Snap Inc.

ning.zhang@snap.com

Li-Jia Li

Snap Inc.

lijiali@cs.stanford.edu

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 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. This allows our network to have adaptive receptive field sizes over different scales of the input. The entire network is trained end-to-end in a siamese framework for visual correspondence tasks. Our method achieves favorable 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 method can generalize to improve hand-crafted descriptors (e.g Daisy) on general visual correspondence tasks. Finally, our attention network can generate visually interpretable scale attention maps.

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 [54, 43, 36], structure-from-motion [1], visual SLAM [40, 27, 39], stereo matching [59, 34], non-rigid 3D reconstruction [14] as well as video segmentation [19].

Central to the correspondence problem is the design of feature descriptors that needs 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 [33, 7] or many [23] of these scales. However, due to their heuristic nature, the scale analyses of these hand-crafted descriptors are limited to a sparse set of im-

age locations with special structures, such as blobs, corners and high contrast regions [37]. 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 [48, 15, 6].

Recently, convolutional neural network (CNN) triumphed in a variety of challenging computer vision tasks such as image classification [28, 50, 25] and object detection [17, 42]. What makes CNN powerful is its flexible architecture to learn progressively complex visual features from low-level filters to high-level concepts. While higher-level features prove to be discriminative for many applications, they often come with a loss in spatial accuracy in the process of yielding larger receptive fields through successive pooling [28, 47], large strides [28], dilated convolution [57] and multi-scale aggregation [50, 31]. In applications that require spatially accurate correspondences, techniques such as spatial transformer network [13] and multi-scale ensemble model [11] prove effective to improve discriminativeness while keeping the spatial accuracy in the resultant feature maps. However, further study is needed on how to optimally combine features from different scales based on the analysis in the scale space.

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. Specifically, in texture-rich area, the network will weigh more on the fine-scale features to ensure correspondence accuracy while in area with less texture, the network will seek 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.

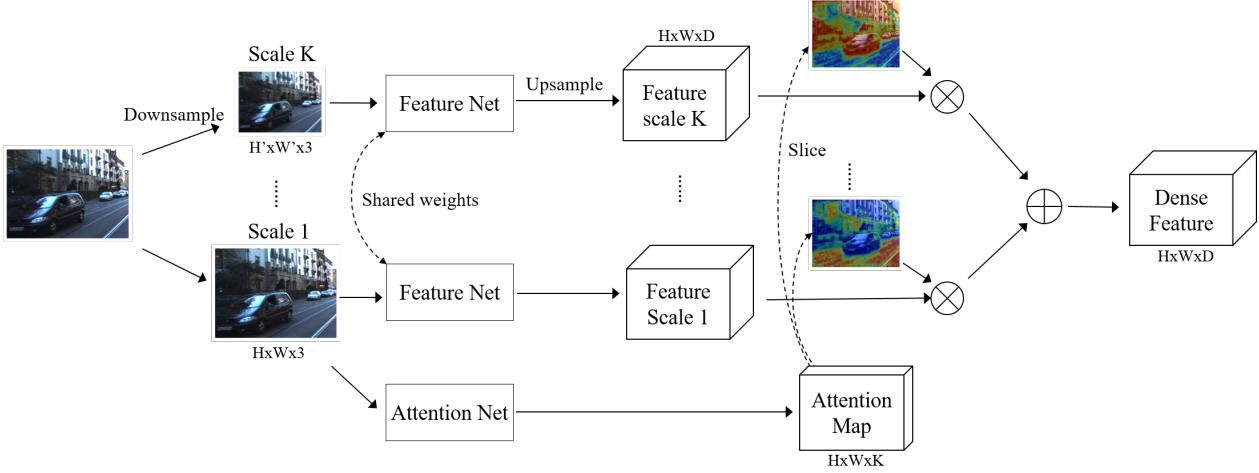


Figure 1: The architecture of AutoScaler. AutoScaler consists of the feature network and the attention network. The feature network extracts feature maps from the input image at multiple scales (note that only two are shown for simplicity) independently with shared weights. The separate attention network computes a pixel-wise attention map from the input image and combine the multi-scale feature maps into one.

The proposed network can generalize to improve the performance of handcrafted descriptors (e.g. Daisy [51]). The full network is trained end-to-end in a siamese framework (Fig. 2) 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. Moreover, our algorithm is able to generate visually interpretable scale attention maps.

2. Related Work

Our work is closely related to learning based approaches for image correspondence. Early methods consider image correspondence problem as a variational inference and focus on learn parameters for MRFs [44]. Later on, the growing availability of synthetic and real world datasets for image correspondence problems makes learning feature representation for similarity matching possible. Representative works include the usage of boosting [52], random forest [54], convex optimization [46], etc. Recently multiple CNNs based approaches are designed to measure similarity between patches across images [21, 32, 34, 59, 11, 58, 26, 12]. In particular, our method is related to [34] in terms of loss functions.

Scale selection has been extensively studied in previous work [33, 35, 30] to select the most salient scale for matching in the scale space. However, the analyses are limited to a few heuristic rules that apply to a sparse set of key points. It is non-trivial to extend the scale selection for dense matching problems such as optical flow. Instead, many previous works explore to propagate the scale labels from sparse key points to the whole image [23, 41, 24]. But it imposes

strong smoothness prior which leads to degraded matching accuracy.

Many approaches have been proposed to enlarge receptive field size to incorporate more contextual information, such as dilated convolution [57], multi-scale aggregation [50, 31], pooling [28] and large strides [28]. Despite greater discriminativeness, we argue that larger receptive field is not always better for correspondence tasks because it often ‘blurs’ the feature map and reduces pixel-level spatial accuracy as supported by previous experiments [59, 58, 34]. Thus, we propose to optimize the trade-off between larger contextual scale and spatial accuracy by using an scale attention scheme.

Attention mechanism in neural network has been studied in [38, 56, 18, 2] with impressive results for different computer vision tasks. The proposed scale-attention model is most related to scale-attention based semantic segmentation method [9] with two main differences. First, our goal is to find the best trade-off between discriminative contextual scale and spatial accuracy, while [9] aims at handling meaningful semantic objects with different sizes. Second, our proposed scale-attention mechanism puts attentions over multi-scale features for matching, whereas [9] utilizes the attention as late-fusion weights over the predicted output from multiple scales.

3. 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.

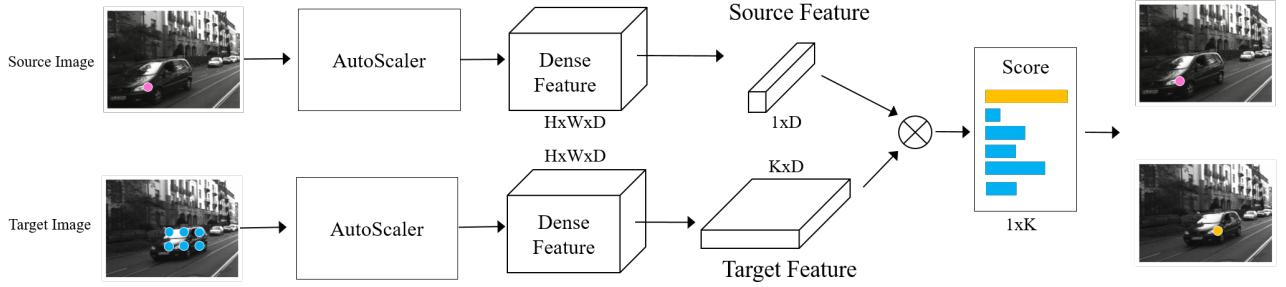


Figure 2: The siamese architecture for visual correspondence. Both source image and target image are fed to our AutoScaler network to extract feature maps. One source feature and a number of target features are selected based on the task at hand. Finally, an inner-product layer is employed to find the correspondence with the best score.

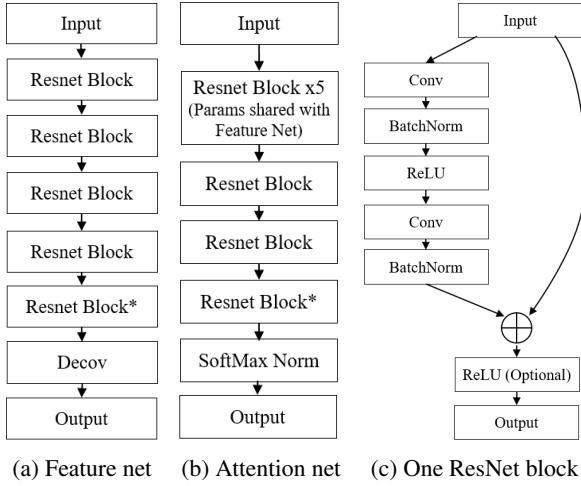


Figure 3: The detailed architecture of the feature network and the attention network. Both networks are built from basic ResNet blocks. *Note that the last ResNet blocks of attention / feature networks do not have ReLU layer.

3.1. 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 $\mathcal{N}_{\mathbf{p}} = \{\mathbf{q}_j | j=1, \dots, N\}$ in the target images; 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 shown in Fig. 2, where each branch processes 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. Fig. 2 depicts the detailed inference process.

Architecture Both the attention network and feature network have a fully convolutional network architecture with shortcut connections to generate pixel-wise feature/score map. The CNN feature net contains five ResNet [25] blocks, each of which contains a conv-batchnorm-relu-conv-batchnorm structure, followed by a shortcut element-wise sum and a final relu layer. The last relu unit in the feature net is removed for non-sparse feature map. 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 weights lies on the range $[0, 1]$ with inter-

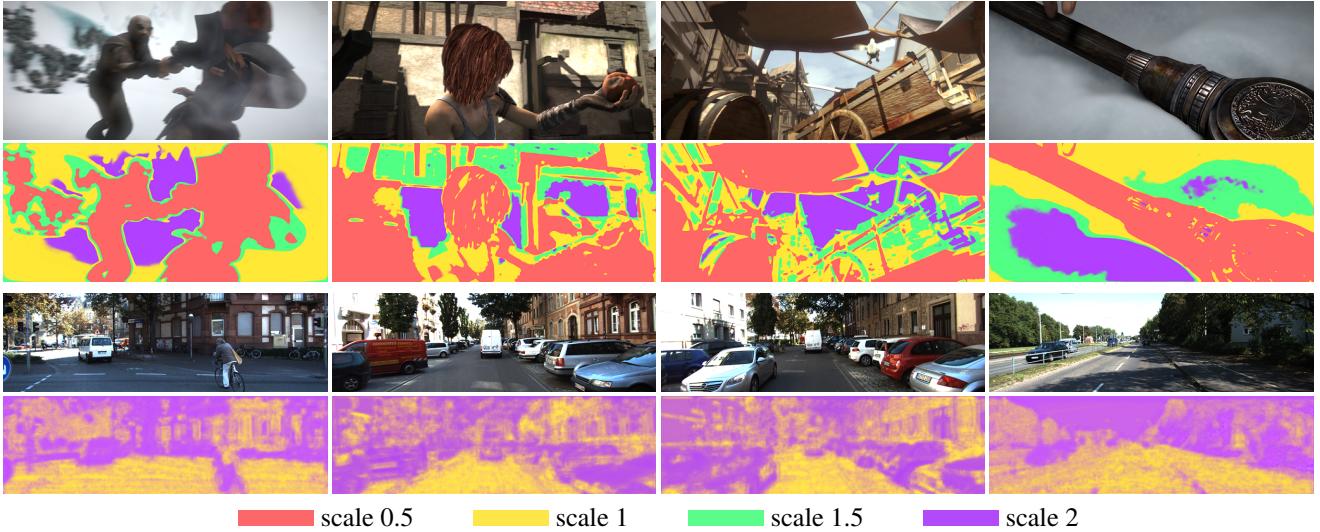


Figure 4: Visualization of scale-attention maps. Four scales are shown for Sintel dataset (top) and two for KITTI (bottom). Scale values indicate down-sample factors of the input image to the feature network (0.5 being the up-sampled finest scale and 2 the down-sampled coarsest). Note that the attention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

preability. Note that we do not use any pooling or strided convolution to ensure that feature/score maps preserve sub-pixel level information. Thus the receptive field size is equal to 23×23 for a single scale feature net. Please refer Fig. 3 for an illustration. In our experiments, the number of filters is 64 (sintel and CUB) or 128 (KITTI). In the following sections, we use 64 filters as example to describe our method.

3.2. 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 minimize cross-entropy loss with respect to the parameters of

our neural networks. Let us denote the i -th training example a triplet that includes the source feature and all the candidate target features $\mathbf{x}_i = \{\mathbf{p}_i, \mathbf{q}_{j \in \mathcal{N}_i}\}$, the goal is to

$$\min_{\mathbf{w}} \sum_{i, y_i \in \mathcal{N}_i} p_{\text{GT}}(y_i) \log p(y_i; \mathbf{x}_i, \mathbf{w})$$

where $p(y_i; \mathbf{x}_i, \mathbf{w})$ is the softmax probability $p(y_i; \mathbf{x}_i, \mathbf{w}) = \frac{\exp(g_{y_i}(\mathbf{x}_i, \mathbf{w}))}{\sum_j \exp(g_j(\mathbf{x}_i, \mathbf{w}))}$ and the score $g_j(\mathbf{x}_i, \mathbf{w})$ is the inner product between \mathbf{p}_i and \mathbf{q}_j : $g_j(\mathbf{x}_i, \mathbf{w}) = \langle \mathbf{p}_i, \mathbf{q}_j \rangle$; The ground-truth probability $p_{\text{GT}}(y_i) = 1$ if y_i is GT correspondence and otherwise $p_{\text{GT}}(y_i) = 0$. where \mathbf{w} represents all the parameters in the scale-attention network that we want to learn through back-propagation, including both feature net and attention net.

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.

3.3. 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 local matching scheme. It

Dataset	Daisy concat	Daisy attention	CNN-31x31 [34]	Single	Concat x2	AutoScaler x2	AutoScaler x4
Sintel	56.79%	78.30%	86.02%	86.95%	87.65%	89.12%	91.84%
KITTI	73.63%	75.67%	90.10%	90.07%	88.04%	92.06%	91.78%

Table 1: Top-1 accuracy over validation dataset. Our proposed AutoScaler outperforms all the baseline networks.

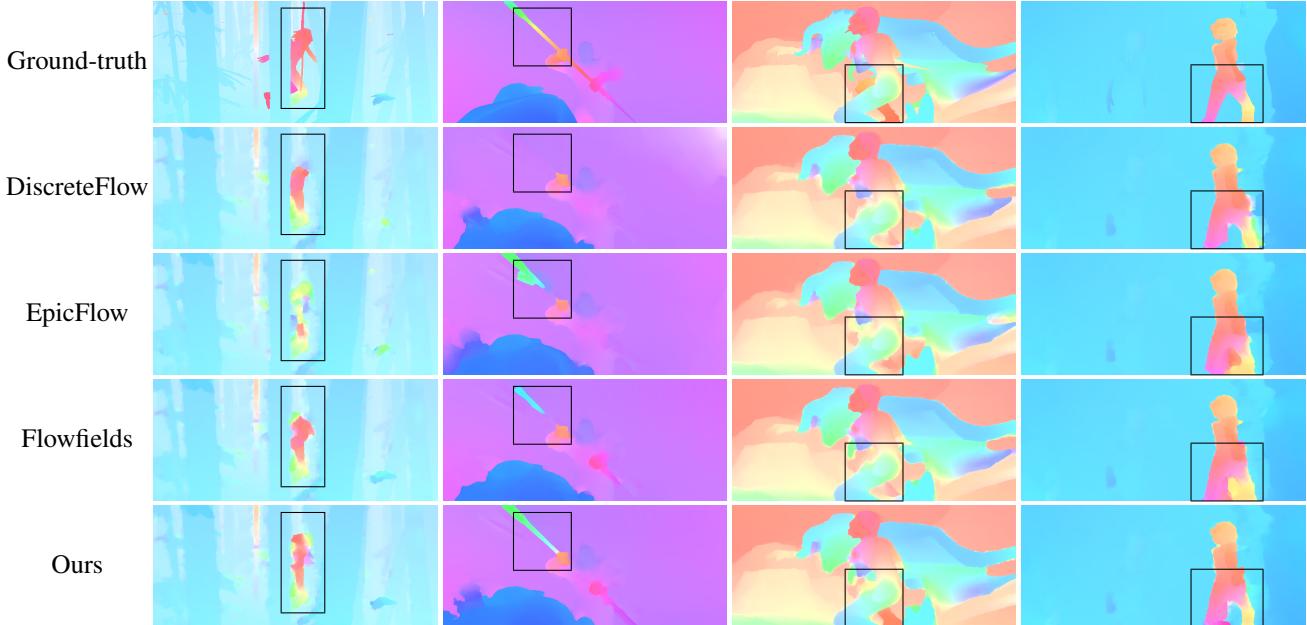


Figure 5: Qualitative results on Sintel optical flow. Our method recovers precise motion of fine structures, like the butterfly, pole and legs as highlighted in boxes.

would greatly help resolve matching ambiguity because of repetitive, smooth textures, or matching along edges. On the other hand, in regions with unique structures, our attention model learns to focus on finer scales with a relatively smaller receptive field, excluding unnecessary context to influence matching.

Extensions to hand-crafted features Our AutoScaler 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.

4. Experiments

This section presents the result of the proposed scale-attention network on both geometric matching and semantic matching tasks. For geometric matching, we perform evaluations on the challenging optical flow benchmarks, MPI-Sintel [8] and KITTI [16]. The semantic matching experiment is conducted over the Caltech-UCSD Birds 2011

Method	EPE-matched	EPE-un	EPE-all
AutoScaler	2.569	34.656	6.076
DeepDisFlow [20]	2.623	31.042	5.728
FlowFields [4]	2.621	31.799	5.810
FullFlow [10]	2.684	30.793	5.895
DiscreteFlow [36]	2.937	31.685	6.077
PatchCollider [54]	2.938	31.309	6.040
EpicFlow [43]	3.060	32.564	6.285
DeepFlow2 [55]	3.093	38.166	6.928
FGI [29]	3.101	35.158	6.607

Table 2: Quantitative experiments on Sintel Dataset.

dataset [53]. We compare with the current state-of-the-art algorithms. Apart from the quantitative experiments, we also visualize and discuss the interpretability of the attention maps that our model generates for different tasks.

4.1. Optical flow on MPI-Sintel

We first evaluate our method on the challenging MPI-Sintel optical flow benchmark [8], which consists of more than 1200 pairs of training images and 1500 pairs of testing images. It is a synthetic dataset with extremely large

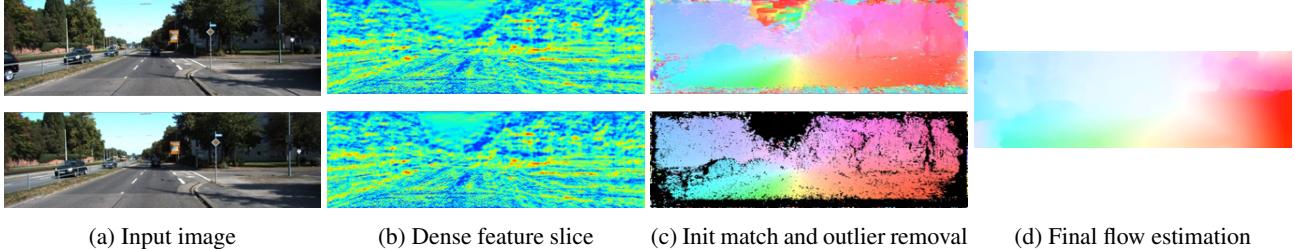


Figure 6: The optical flow estimation pipeline. From left to right: input image pairs, scale-attended feature maps, initial noisy estimation (top) and flow field after outlier removal (bottom), final result after extrapolation [43].

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.

Training data We firstly split the 22 training images into training (1-16) and validation (17-22). For each pair of images, we randomly sampled 10K local correspondent pairs, and for each pair, we randomly selected 200 negative samples within the limit of motion range $[-210, 200]$.

Architecture design In order to validate the efficacy of our proposed network, we evaluate Top-1 accuracy matching performance over baseline network architecture. We compared different architectures trained under the same multi-class siamese configuration with softmax loss. Table. 1 demonstrates the top-1 accuracy matching performance of different architectures on the validation subset of Sintel and KITTI. The competing algorithms include CNN-31x31, a nine-layer fully convolutional network used in [34]. Similar to our approach, [34] also adopts softmax loss for training and the architecture does not include pooling or stride convolution. “Single” refers to our basic single-scale deep architecture with five ResNet blocks. “Concat x2” is a two-scale deep architecture, with the feature vector as a concatenation of features from the two scales. “AutoScaler $\times K$ ” is the proposed AutoScaler with K scales. In this experiment we compare the performance between two and four scales. As shown in the table, our proposed architecture outperform all the competing algorithms. Especially, we show that with the attention mechanism the matching performance is better than simply concatenating two scales. Moreover, the four-scale AutoScaler outperforms the two-scale version. It is also worth noting that we extract DAISY features from multiple scales, and trained our attention network to fuse the features. We found that it outperforms con-

catenating multi-scale DAISY features with a large margin, which demonstrates the efficacy of the attention model.

Dense Flow In the testing stage, in order to generate the dense flow, we firstly use our network to extract dense features. For each local 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 highest score. This would produce a raw dense flow fields with outliers. We remove outliers through forward-backward consistency check. To be specific, for each pixel p , we check whether the condition $\|\mathbf{u}_{\text{backward}}(p + \mathbf{u}_{\text{forward}}(p)) + \mathbf{u}_{\text{forward}}(p)\| \leq t$ is satisfied, where $\mathbf{u}_{\text{backward}}$ is the estimated backward optical flow, $\mathbf{u}_{\text{forward}}$ is the forward optical flow field. We then discard inconsistent motion estimations with above the threshold t . In practice $t = 3$ is used for dense flow. This gives us an optical flow map with partial observation. And we interpolate/extrapolate the missing pixels with Epicflow algorithm [43]. Fig. 6 illustrates the whole process of dense optical flow pipeline.

Quantitative Results We submit our algorithm’s output to Sintel benchmark and compare it against the top-ranked prior work. We focus on the final benchmark, which is more challenging due to the presence of motion blur and various shading and reflectance changes. Table. 2 shows the quantitative results against the competing algorithms. To be specific, our method achieves the best performance on the EPE-matched measure among all the competing algorithms, and ranks 5th in EPE-all metric. This suggests the proposed network is very competitive in finding existed correspondence. However, relative large errors appear in unmatched regions, which suggests the interpolation might not be tuned to fit for our network’s output. It is worth to note that, unlike some competing algorithms, such as Flow-fields [4], DiscreteFlow [20, 36] and Fullflow [10], the proposed dense optical flow algorithm does not exploit a comprehensive MRF post-processing step to propagate the flow estimation to occluded regions. This would speed-up the optical flow matching process since MRFs processing is the

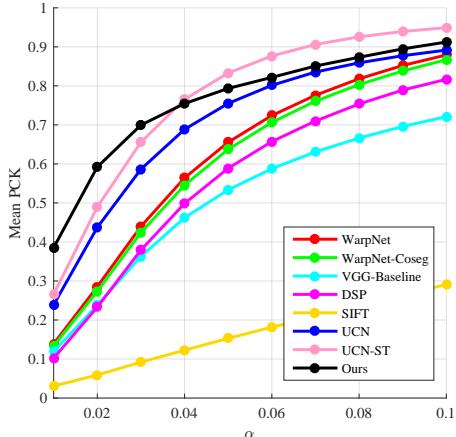


Figure 7: Quantitative result on CUB dataset.

bottleneck for many methods. In practice, our method takes 0.5s for computing features, 2 mins for initial matching and 2.5s for Epicflow interpolation on Sintel.

Qualitative Results Fig. 5 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 resolution.

4.2. Optical flow on KITTI

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.

Training data We separate the training dataset into 160 pairs as train and 40 pairs as validation. Following the similar experiment configuration on Sintel, we sample 10k local correspondences from each image pair and for each pair 200 negative samples.

Dense flow We follow similar pipeline described in Sec. 4.1 to generate dense optical flow with our network, as shown in Fig. 6. To be specific, we firstly compute features with the proposed AutoScalar network, and conduct initial matching. Forward-backward consistency check is used to remove outliers and Epic flow is exploited to generate the final dense optical flow field.

Architecture comparison We first report the top-1 accuracy over validation set with different architectures, as shown in Table. 1. The proposed AutoScalar model outperforms all competing network architecture. And the attention mechanism also brings limited improvement on DAISY feature. This validates the effectiveness of the proposed architecture. It is worth noting that on KITTI AutoScaler x2

Method	Fl-bg	Fl-fg	Fl-all
SDF [3]	8.61 %	26.69 %	11.62 %
SOF [45]	14.63 %	27.73 %	16.81 %
CNN-HPM[5]	18.33 %	24.96 %	19.44%
DiscreteFlow [36]	21.53 %	26.68 %	22.38 %
AutoScaler	21.85 %	31.62 %	25.64 %
FullFlow [10]	23.09 %	30.11 %	24.26 %
EpicFlow [43]	25.81 %	33.56 %	27.10 %
DeepFlow2 [55]	27.96 %	35.28 %	29.18 %
PatchCollider [54]	30.60 %	33.09 %	31.01 %
SGM+C+NL [49]	40.81 %	35.42 %	39.91 %

Table 3: Quantitative experiments on KITTI Flow 2015 Dataset. 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.

outperforms AutoScaler x4. This contrasts what we find in Sintel. We suspect that it is because the dataset bias: KITTI does not have many large regions without any textures.

Quantitative results We submit our result to KITTI optical flow benchmark. The results are shown in Table. 3. Note that KITTI is a dataset captured in a special autonomous 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, 45]. Apart from those methods, our approach achieves comparable results against other competing algorithms utilizes generic matching techniques. From the table we can see our method is comparable with most competing algorithms. The proposed method is not favorable among all the deep learning based algorithms. One potential reason is the proposed method does not exploit the extrapolation, which brings large error in non-visible regions because of self-occlusion and truncation. We plan to incorporate structured variational prediction into the proposed model to solve this problem in future.

4.3. Semantic Matching

Unlike geometric matching tasks, such as optical flow and stereo, 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 CUB-2011-2011 dataset, which contains 11788 images of 200 bird categories, with 15 parts annotated.

Training data We follow the experiment configuration of [26], which utilizes the training set to extract training pairs and 5000 pairs images from the validation subset as testing pairs. We crop each image with the bounding box of the bird



(a) Query image (b) Ground-truth (c) Ours (d) Query image (e) Ground-truth (f) Ours

Figure 8: Qualitative results on CUB semantic matching. Our method is able to capture semantic meaningful matching across species and poses, with sub-pixel level accuracy. A typical failure case is the left-right feet ambiguity (see the bottom row).

and conduct matching on the cropped image pairs. For each training iteration, we randomly pick two pairs of images and use all the corresponding keypoints between them for training. The negative samples are randomly selected 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 $L = \frac{1}{2}(\sqrt{w_{\text{src}}^2 + h_{\text{src}}^2} + \sqrt{w_{\text{tgt}}^2 + h_{\text{tgt}}^2})$ is the mean diagonal size of the image pairs. Note that not all the 15 keypoints are visible in both images, we follow the configuration of [26] and discard these invisible keypoints when computing the metric.

Quantitative result We compared against the more recent state-of-the-arts algorithms on CUB matching dataset, namely WarpNet [26], Universal correspondence network [13], and DSP [22], along with two widely used features including VGGnet [47] and SIFT [33]. Fig. 7 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 sub-pixel accuracy. When the threshold α becomes large, our method ranks second among all the competing algorithm, following UCN [13]. This suggests that AutoScaler better captures finer accurate details while in the meantime performs competitively in reasoning the semantic meaning of the local part of the birds. Fig. 7 show the examples of the qualitative matching results. As shown in this figure, our method performs well in most cases across various poses, species and scales. The most failure cases are due to the ambiguity in matching left and right feet.

5. Conclusions

We propose the AutoScaler, a scale-attention network that optimally combines dense feature maps from different scales. This scheme allows our neural network to have an adaptive receptive field size. The extensive experiments show that our method is not only extremely effective but is also able to generate visual interpretable scale attentions.

References

- [1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2009. 1
- [2] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. In *ICLR*, 2015. 2
- [3] M. Bai, W. Luo, K. Kundu, and R. Urtasun. Exploiting semantic information and deep matching for optical flow. In *ECCV*, 2016. 7
- [4] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *ICCV*, 2015. 5, 6
- [5] C. Bailer, K. Varanasi, and D. Stricker. Cnn based patch matching for optical flow with thresholded hinge loss. *arXiv*, 2016. 7
- [6] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. In *ECCV*. 2010. 1
- [7] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 1
- [8] 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
- [9] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. *CVPR*, 2016. 2
- [10] Q. Chen and V. Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *CVPR*, 2016. 5, 6, 7
- [11] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang. A deep visual correspondence embedding model for stereo matching costs. In *ICCV*, 2015. 1, 2
- [12] C. B. Choy, J. Gawk, S. Savarese, and M. Chandraker. Universal correspondence network. *NIPS*, 2016. 2
- [13] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker. Universal correspondence network. In *NIPS*, 2016. 1, 8
- [14] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, et al. Fusion4d: real-time performance capture of challenging scenes. *SIGGRAPH*, 2016. 1
- [15] B. Drayer and T. Brox. Combinatorial regularization of descriptor matching for optical flow estimation. In *BMVC*, 2015. 1
- [16] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 5, 7
- [17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1
- [18] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *ICML*. 2015. 2
- [19] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2141–2148. IEEE, 2010. 1
- [20] F. Güney and A. Geiger. Deep discrete flow. In *ACCV*, 2016. 5, 6
- [21] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015. 2
- [22] T. Harada, Y. Ushiku, Y. Yamashita, and Y. Kuniyoshi. Discriminative spatial pyramid. In *CVPR*, 2011. 8
- [23] T. Hassner, S. Filosof, V. Mayzels, and L. Zelnik-Manor. Sifting through scales. *PAMI*, 2016. 1, 2
- [24] T. Hassner, V. Mayzels, and L. Zelnik-Manor. On sifts and their scales. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012. 2
- [25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016. 1, 3
- [26] A. Kanazawa, D. W. Jacobs, and M. Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. *CVPR*, 2016. 2, 8
- [27] B. Kitt, A. Geiger, and H. Lategahn. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *IV*, 2010. 1
- [28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 2
- [29] Y. Li, D. Min, M. N. Do, and J. Lu. Fast guided global interpolation for depth and motion. In *ECCV*, 2016. 5
- [30] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 1998. 2
- [31] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2
- [32] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014. 2
- [33] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 1, 2, 8
- [34] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *CVPR*, 2016. 1, 2, 5, 6
- [35] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *IVC*, 2004. 2
- [36] M. Menze, C. Heipke, and A. Geiger. Discrete optimization for optical flow. In *GCPR*, 2015. 1, 5, 6, 7
- [37] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, Nov. 2005. 1
- [38] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu. Recurrent models of visual attention. In *NIPS*. 2014. 2
- [39] R. Mur-Artal, J. Montiel, and J. D. Tardós. Orb-slam: a versatile and accurate monocular slam system. *IEEE Trans on Robotics*, 2015. 1
- [40] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtm: Dense tracking and mapping in real-time. In *ICCV*, 2011. 1
- [41] W. Qiu, X. Wang, X. Bai, A. Yuille, and Z. Tu. Scale-space sift flow. In *WACV*, 2014. 2
- [42] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1

- [43] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. [1](#), [5](#), [6](#), [7](#)
- [44] S. Roth and M. J. Black. On the spatial statistics of optical flow. In *IJCV*, 2007. [2](#)
- [45] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical flow with semantic segmentation and localized layers. In *CVPR*, 2016. [7](#)
- [46] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1573–1585, 2014. [2](#)
- [47] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [1](#), [8](#)
- [48] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. [1](#)
- [49] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. [7](#)
- [50] 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. [1](#), [2](#)
- [51] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. In *PAMI*, 2010. [2](#)
- [52] T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua. Learning image descriptors with the boosting-trick. In *Advances in neural information processing systems*, pages 269–277, 2012. [2](#)
- [53] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. [5](#)
- [54] S. Wang, S. Ryan Fanello, C. Rhemann, S. Izadi, and P. Kohli. The global patch collider. In *CVPR*, 2016. [1](#), [2](#), [5](#), [7](#)
- [55] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013. [5](#), [7](#)
- [56] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*. 2015. [2](#)
- [57] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016. [1](#), [2](#)
- [58] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015. [2](#)
- [59] J. Žbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR*, 2015. [1](#), [2](#)