Pyramidal Semantic Correspondence Networks

Sangryul Jeon, *Student Member, IEEE*, Seungryong Kim, *Member, IEEE*, Dongbo Min, *Senior Member, IEEE*, and Kwanghoon Sohn, *Senior Member, IEEE*

Abstract—This paper presents a deep architecture, called pyramidal semantic correspondence networks (PSCNet), that estimates locally-varying affine transformation fields across semantically similar images. To deal with large appearance and shape variations that commonly exist among different instances within the same object category, we leverage a pyramidal model where the affine transformation fields are progressively estimated in a coarse-to-fine manner so that the smoothness constraint is naturally imposed. Different from the previous methods which directly estimate global or local deformations, our method first starts to estimate the transformation from an entire image and then progressively increases the degree of freedom of the transformation by dividing coarse cell into finer ones. To this end, we propose two spatial pyramid models by dividing an image in a form of quad-tree rectangles or into multiple semantic elements of an object. Additionally, to overcome the limitation of insufficient training data, a novel weakly-supervised training scheme is introduced that generates progressively evolving supervisions through the spatial pyramid models by leveraging a correspondence consistency across image pairs. Extensive experimental results on various benchmarks including TSS, Proposal Flow-WILLOW, Proposal Flow-PASCAL, Caltech-101, and SPair-71k demonstrate that the proposed method outperforms the lastest methods for dense semantic correspondence.

Index Terms—Dense semantic correspondence, spatial pyramid model, coarse-to-fine inference

1 Introduction

Establishing dense correspondences across semantically similar images is essential for numerous computer vision and computational photography applications, such as scene parsing, semantic segmentation, and image editing [1], [2], [3], [4], [5]. Unlike classical dense correspondence tasks such as stereo matching [6], [7], [8] or optical flow estimation [9], [10], [11] that have been dramatically advanced, semantic correspondence task still remains unsolved due to severe intra-class appearance and shape variations across semantically similar images.

To address these challenges, several approaches [12], [13], [14], [15] attempted to capture reliable matching evidences by leveraging deep convolutional neural network (CNN) based descriptors with a high invariance to appearance variations. While they examined various geometric models for the spatial regularization of transformation fields, such as translational motion [14], [16] or affine transformation [17], their smoothness constraints are imposed in a handcrafted manner and thus they are often trapped in local minima unless an appropriate initial solution is given.

To alleviate this, recent state-of-the-art techniques [18], [19], [20], [21], [22], [23] begun directly regressing transformation fields through an end-to-end deep network architecture. As a pioneering work, spatial transformer networks (STNs) [24] offer a way to deal with geometric variations within CNNs. Inspired by this, several methods [18], [19], [20] proposed CNN architectures that estimate global transformation fields between input images by mimicking traditional matching pipelines [25], i.e., feature extraction, cost volume construction, and global transformation parameter regression. Modeling an image deformation with global transformation fields provides the robustness against to semantic variations by roughly aligning an overall structure of an object, but simultaneously it has shown limited performance in capturing fine-grained object details, as shown in [23], [26]. More recently, some methods proposed to infer locally-varying transformation fields using neighbourhood consensus [22], recurrent framework [21], or kernel soft argmax [23], outperforming previous methods based on a global transfomation model [18], [19], [20]. However, they often have difficulties in handling relatively large geometric transformations since their matching candidates are strictly constrained within a local region [21], [23], or only local neighborhood patterns are utilized for identifying reliable matches [22] without the explicit consideration of a global deformation.

In this paper, we present a novel CNN architecture, called pyramidal semantic correspondence networks (PSCNet), that estimates locally-varying affine transformation fields across semantically similar images in a coarse-to-fine fashion. Unlike the previous transformation regression networks [18], [19], [20], [21], [22], [23] that suffer from the

- *Sangryul Jeon and Kwanghoon Sohn are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea. E-mail: {cheonjsr, khsohn}@yonsei.ac.kr.*
- *Seungryong Kim is with the Department of Computer Science and Engineering, Korea University, Seoul 02841, South Korea. E-mail: seungryong_kim@korea.ac.kr.*
- *Dongbo Min is with the Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, South Korea. E-mail: dbmin@ewha.ac.kr.*

Manuscript received 7 September 2019; revised 8 June 2021; accepted 16 October 2021. Date of publication 29 October 2021; date of current version 3 November 2022.
This work was supported in part by the R&D program for Advanced Integrated-intelligence for Identification (AIID) through the National Research Foundation under Grant NRF-2018M3E3A1057289 and Yonsei University Research Fund under Grant 2021-22-0001. This work of Seungryong Kim was supported in part by MSIT (Ministry of Science and ICT) under the ICT Creative Consilience program under Grant IITP-2021-2020-0-01819.
(Corresponding author: Kwanghoon Sohn.)
Recommended for acceptance by R. Farrell.
Digital Object Identifier no. 10.1109/TPAMI.2021.3123679

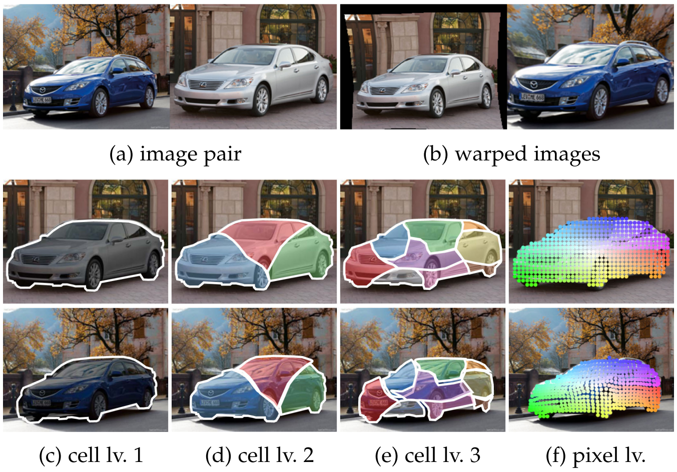

(a) image pair
(b) warped images
(c) cell lv. 1
(d) cell lv. 2
(e) cell lv. 3
(f) pixel lv.

Fig. 1. Visualization of our PSCNet-SE pyramid model based on the semantic elements of an object: (a) source and target images, (b) warped images with the final correspondences of (f). The semantic elements of the source image are warped with the estimated affine field at (c) level 1, (d) level 2, and (e) level 3. Thanks to our coarse-to-fine scheme, we achieve both the robustness to semantic variations and fine-grained localization precision at the same time.

trade-off between robustness to semantic variations and fine-grained localization precision, our method achieves both at the same time by formulating a coarse-to-fine framework within deep learning pipelines. To this end, we propose two spatial pyramid models by dividing an image into quad-tree uniform rectangles (Fig. 5), or into multiple semantic elements of an object (Fig. 1). Both models first estimate a global affine transformation over an entire image, and then progressively increase the degree of freedom of the transformation, naturally imposing the pyramidal smoothness constraint on the pixel-level affine transformation field. The different pyramid levels are linked with a warping operation module of STNs [24] which allows our networks to work in an end-to-end manner. Moreover, to address the lack of training data, a novel weakly-supervised training scheme is introduced that generates progressively evolving supervisions by checking the correspondence consistency at each level. Extensive experimental results on various benchmarks, including TSS [27], Proposal Flow-WILLOW [12], Proposal Flow-PASCAL [28], Caltech-101 [29], and recent SPair-71k [30] demonstrate that the proposed method outperforms the latest methods for dense semantic correspondence.

A preliminary version of this paper has appeared as a full paper in the 2018 European Conference on Computer Vision (ECCV) [26]. Compared to our previous work, we newly add (1) an extended pyramid model based on the semantic elements of the object; (2) an in-depth analysis of our approach; and (3) an extensive comparative study with latest state-of-the-arts using various datasets.

2 Related Work

2.1 Regularization With Handcrafted Constraints

Early works for dense semantic correspondence rely on manually designed optimization techniques to estimate spatially regularized transformation fields, employing handcrafted features such as SIFT [31] and DAISY [32]. The SIFT flow [2] pioneered the idea of dense correspondence across different scenes through a hierarchical optimization with a multi-resolution image pyramid. Inspired by this, Kim *et al.* [3] proposed the deformable spatial pyramid (DSP) which performs multi-scale regularization within a hierarchical graph. A more relevant method to ours is the work of Yang *et al.* [33] that constructs object-aware hierarchical graph (OHG) and regulates matching consistency in a coarse-to-fine manner. However, [33] relies on handcrafted algorithm to generate object proposals [34] and to detect semantic parts, yielding limited performance under large appearance and geometric variations.

For the higher invariance to appearance variations, CNN-based descriptors have been recently utilized as a matching evidence. Several methods elevated matching quality by improving the robustness against to the geometric variations. Universal correspondence network (UCN) [35] was proposed to employ STNs [24] at the pixel level for transforming their receptive fields adaptively. Novotny *et al.* [36] proposed AnchorNet that learns geometry-sensitive features for semantic matching with weak image-level labels. They further improve the robustness of AnchorNet [36] by casting learning into a probabilistic formulation [37]. Kim *et al.* [14] proposed fully convolutional self-similarity (FCSS) descriptor that formulates local self-similarity within a fully convolutional network. In [17], they extended FCSS descriptor by explicitly considering affine transformations, and then proposed a discrete-continuous optimization framework to infer dense affine transformation fields efficiently. Ham *et al.* [12] presented the proposal flow (PF) algorithm to estimate correspondences using object proposals. Inspired by PF [12], Ufer *et al.* [16] proposed a method based on convolutional feature pyramids and activation-guided feature selection. Han *et al.* [13] proposed SCNet to learn the similarity function and geometry kernel of PF algorithm, but they compute the final transformation field with non-trainable interpolation step. Note that as all of these techniques rely on the handcrafted regularization, they do not guarantee the robustness to large intra-class deformations that is possible with end-to-end CNN models.

2.2 Regularization With End-to-End CNN Models

Recent state-of-the-art methods for dense semantic correspondence regress the transformation fields directly through an end-to-end CNN model. Rocco *et al.* [18], [20] proposed a CNN architecture mimicking the traditional matching pipeline that estimates a global geometric model such as an affine and TPS transformation. Seo *et al.* [19] extended this architecture with an offset-aware correlation kernel to put more attention to reliable similarity scores. However, all methods focus on estimating the global transformation field and thus exhibits limited performance when dealing with fine-grained geometric deformations.

To address this issue, several methods [21], [38] proposed to estimate locally-varying transformation field instead of global geometry parameters. Kim *et al.* [21] proposed recurrent transformation networks (RTNs) that iteratively estimate spatial transformations between the input images and use these transformations to generate aligned convolutional features. Rocco *et al.* [22] proposed neighbourhood consensus networks (NCNet) that identifies sets of spatially consistent matches by analyzing neighbourhood

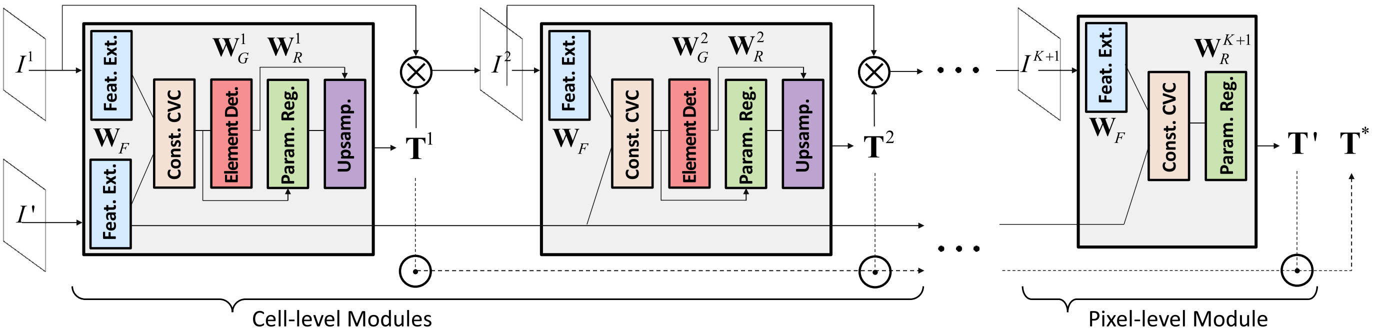


Fig. 2. Network configurations of the PSCNet-UR and PSCNet-SE, which are defined on the pyramidal model and consist of several cell-level modules and a single pixel-level module. Each module is designed to mimic the standard matching process within a deep architecture, including feature extraction, cost volume construction, and transformation field regression. Note that the PSCNet-UR and PSCNet-SE models share the same network architectures except element detection networks. For the PSCNet-UR model, the element detection networks are replaced with the grid generator of STNs [24].

consensus patterns. DCCNet [39] and ANCNet [40] extended NCNet [22] framework with context-aware semantic representation and non-isotropic 4D convolution kernel, respectively. SFNet [23] was proposed to leverage binary foreground masks and the kernel soft argmax function with a loss that combines mask and flow consistency constraints. However, constraining matching candidates within a local region [21], [23] and analyzing only neighborhood patterns [22] often lead to unreliable results when handling relatively large geometric variations. Recently released SPair-71k benchmark [41] allows massive ground-truth keypoint annotations to be utilized for current state-of-the-art techniques. Its baseline, HPF [30], employed multiple intermediate feature maps extracted from backbone networks to achieve both semantic invariance and localization ability. DHPF [42] and SCOT [43] further extended HPF [30] with learnable module for feature selection and with optimal transport formulation for refined similarity scores, respectively.

2.3 Unsupervised Object Part Detection

Methods for unsupervised landmark or part detection generally rely on the equivariance property such that the object landmarks should be consistently detected with respect to given image deformations. As a pioneering work, Thewlis *et al.* [44] proposed to randomly synthesize the image transformations for learning to discover the object landmarks that are equivariant with respect to those transformations. SCOPS [45] employed equivariance and semantic consistency constraints to obtain part-level information normalizing the probability maps of semantic parts along spatial dimension. Shilong *et al.* [46] removed the dependency of SCOPS [45] on object saliency maps by disentangling the appearance and shape representations. However, all the previous equivariance-based approaches biased on a specific object category and unable to handle general object classes that frequently encountered in semantic correspondence problem.

3 Problem Formulation and Overview

Given a pair of images I and I', the objective of dense correspondence estimation is to establish a correspondence i' for each pixel $i = [i_\mathbf{x}, i_\mathbf{y}]$. In this work, we infer a field of affine transformations, each represented by a 2×3 matrix

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{T}_{i,\mathbf{x}} \\ \mathbf{T}_{i,\mathbf{y}} \end{bmatrix}, \tag{1}$$

that maps pixel i to $i' = \mathbf{T}_i \mathbf{i}$, where $\mathbf{i}$ is a pixel i represented in the homogeneous coordinates such that $\mathbf{i} = [i, 1]^T$.

Though recent state-of-the-arts [21], [22], [23] for semantic correspondence estimation yield satisfactory results, they still suffer from addressing relatively large geometric variations. It is mainly because they adopt a local search strategy on the fine scale of input images to estimate fine-detailed transformation fields [47]. Our key observation is that the transformation fields estimated at coarse scales tend to be robust to geometric variations while the results at the finest scale better preserve fine-grained details of objects. To meet both requirements, inspired by the pyramidal graph model [3], [48], [49] that has been commonly used in classical correspondence approaches, we propose a novel deep architecture that directly regresses dense affine transformation fields in a coarse-to-fine manner.

As illustrated in Fig. 2, starting from estimating a global affine transformation field, we progressively increase the degree of freedom of the transformation by dividing coarse cell into finer ones through K cell-level modules, and finally into every pixels with a single pixel-level module. In our pyramidal model, the input image I^k is obtained by warping image I^{k-1} with transformation field $\mathbf{T}^{k-1}$. Thus, each module needs to estimate the residual transformation field. By composing all the estimated affine fields from $K+1$ modules, the final transformation field $\mathbf{T}^*$ can be computed as the multiplications of augmented matrix in homogeneous coordinates such that

$$\mathbf{M}(\mathbf{T}_i^*) = \textstyle\prod_{n \in \{1,\ldots,K\}} \mathbf{M}(\mathbf{T}_i^n) \cdot \mathbf{M}(\mathbf{T}_i'), \tag{2}$$

where $\mathbf{T}'$ denotes an affine transformation field estimated at the pixel-level module, $\prod$ is a matrix product operator, and $\mathbf{M}(\mathbf{T})$ represents $\mathbf{T}$ in homogeneous coordinates as a form of matrix $[\mathbf{T}; [0, 0, 1]]$.

4 Pyramidal Semantic Correspondence Networks

Our PSCNet consists of two different modules; K cell-level module and a single pixel-level module. Both modules are

TABLE 1
Descriptions of the Used Notations in Our Framework

Notations for PSCNet-UR and PSCNet-SE models		
Symbol	Description	Dimension
$\mathbf{F}$	Extracted feature from image I	$H \times W \times D$
$\mathbf{C}$	Cost volume	$H \times W \times r^2$
$\hat{\mathbf{T}}$	Cell-wise affine transformation field	$N \times 6$
$\mathbf{T}$	Upsampled affine transformation field	$H \times W \times 6$
Additional notations for PSCNet-SE model		
Symbol	Description	Dimension
$\psi(n)$	Probability map of n^{th} semantic part	$H \times W \times 1$
$\phi(n)$	Spatial coordinate of barycenter of $\psi(n)$	2×1
τ	Pseudo ground-truth annotation map	$H \times W \times 2$

The superscript k is dropped for simplicity.

designed to infer a dense affine transformation field by mimicking the standard matching process, i.e., feature extraction, cost volume construction, and transformation regression. As shown in Fig. 2, when two images I and I' are given, convolutional features are first extracted through the feature extraction networks by concatenating multi-level intermediate activations. Then, the similarity scores between image feature maps are computed at the cost volume construction layer where the search candidates are constrained in a coarse-to-fine fashion along our pyramidal model. Finally, a dense affine transformation field is estimated by passing the constructed cost volume sequentially through the element detection networks that estimate spatial probabilities of each semantic elements, the regression networks that output the parameters of affine transformation for those elements, and the upsampling layer that interpolates a sparsely inferred affine transformation field to a dense one. For readability, we summarized the descriptions of the used notations in our framework in Table 1.

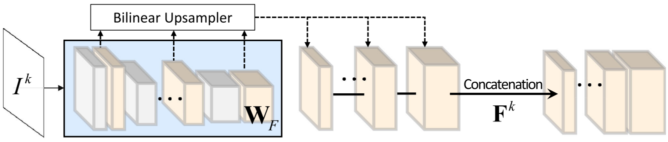
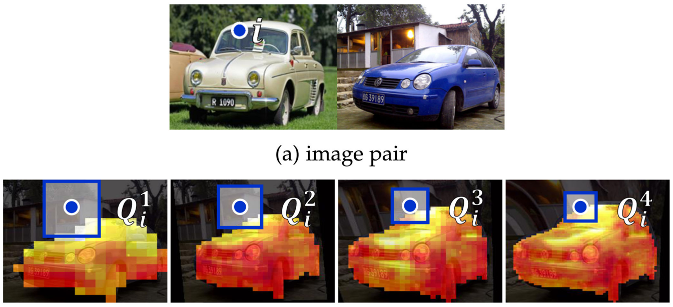


Fig. 3. Visualization of multi-scale feature extraction. To resolve local ambiguities, we leverage the inherent hierarchy of CNNs by pooling multi-level intermediate convolutional activations with upsampling.



(a) image pair

(b) cell lv. 1 (c) cell lv. 2 (d) cell lv. 3 (e) pixel lv.

Fig. 4. Visualization of the constrained search window Q_i^k: (a) source image and a reference pixel (blue colored). The matching costs are visualized as the heat maps for the reference pixel at (b) level 1, (c) level 2, (d) level 3, and (e) pixel-level.

4.1 Multi-Scale Feature Extraction

While conventional CNN-based descriptors have shown excellent capabilities in handling intra-class appearance variations [50], [51], they have difficulites in yielding both semantic robustness and matching precision ability at the same time due to the fixed scale of their receptive fields. To overcome this limitation, we exploit the different levels of features among early to late layers of CNNs, as illustrated in Fig. 3. At each level k, we pool some of multi-level intermediate feature maps by concatenating them along channels with upsampling. Given an image I, this will produce a dense set of descriptors $\mathbf{F}^k \in \mathbb{R}^{H \times W \times D^k}$ denoting H and W as the number of features along image height and width (i.e., the spatial resolution of the features), and D as the dimension of the features

$$\mathbf{F}^k = \bigcup\nolimits_{n \in M^k} \mathcal{F}(I^k; \mathbf{W}_F^n), \tag{3}$$

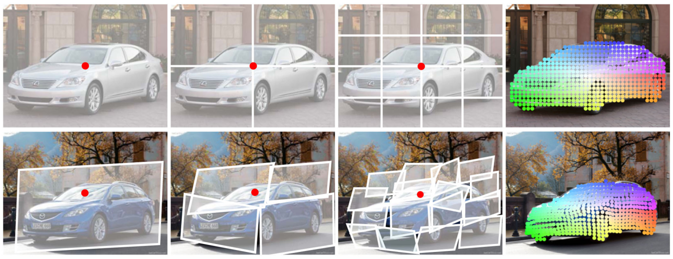
where $\bigcup$ denotes the concatenation operator, $\mathbf{W}_F^n$ is the feature extraction network parameter until nth convolutional layer and M^k is the sampled indices of convolutional layers. Similarily, the feature map of target image I', denoted as $\mathbf{F}'^{,k}$, can be extracted in a siamese network configuration. Note that the convolutional activations of I' are computed only once and then $\mathbf{F}'^{,k}$ is computed by concatenating some of them according to the indices M^k.

While existing transformation regression networks [18], [19], [20], [23] utilize only fixed and untransformed versions of the features, the proposed method iteratively extracts convolutional features of the warped image I^k in the pyramidal model, enabling geometric-invariant feature representation in a progressive fashion [21].

4.2 Constrained Cost Volume

To estimate a transformation field between image pair I^k and I', the matching cost according to search spaces should be computed using extracted features $\mathbf{F}^k$ and $\mathbf{F}'^{,k}$. Following the recently proposed transformation regression networks [18], [19], [20], [22], [23], we first construct the cost volume computed with respect to a set of translational motions within the search space, and then determine a locally-varying affine transformation field by passing it through subsequent convolutional layers.

Compared to [18], [19], [20], [22] that construct a full cost volume considering all possible samples within an image, we construct a partial cost volume by constraining the



(c) cell lv. 1 (d) cell lv. 2 (e) cell lv. 3 (f) pixel lv.

Fig. 5. Visualization of spatial pyramid model of PSCNet-UR based on the uniformly divided rectangles: estimated affine field at (a) level 1, (b) level 2, (c) level 3, and (d) pixel-level.

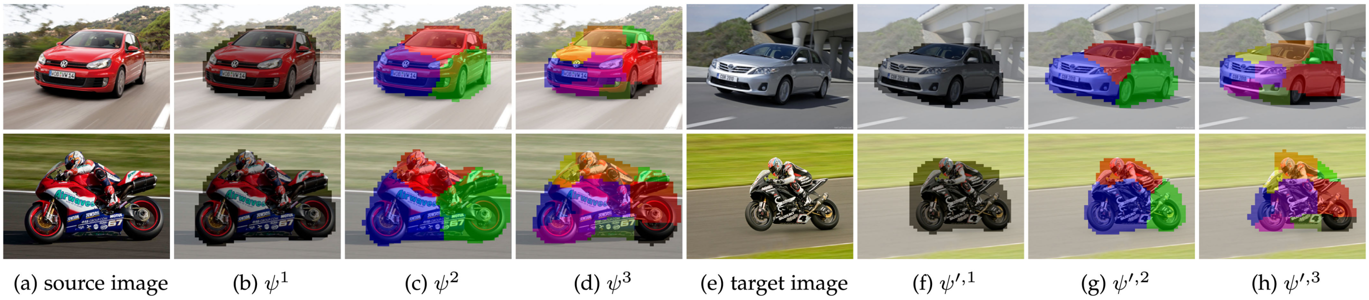

(a) source image (b) ψ^1 (c) ψ^2 (d) ψ^3 (e) target image (f) $\psi'^{,1}$ (g) $\psi'^{,2}$ (h) $\psi'^{,3}$

Fig. 6. Visualization of the semantic elements discovered by PSCNet-SE at each level: (a) source image, (e) target image, the color-coded semantic elements at (b), (f) level 1, (c), (g) level 2, and (d), (h) level 3. The elements with the same color are supposed to match each other.

search range with respect to the transformations estimated at the previous pyramid level. Concretely, the matching costs $\mathbf{C}_{ij}^k$ between extracted features $\mathbf{F}_i^k$, $\mathbf{F}_j'^{,k}$ are computed as a rectified cosine similarity within a search range Q_i^k, such that

$$\mathbf{C}_{ij}^k = \max\left(0, \frac{\mathbf{F}_i'^{,k} \cdot \mathbf{F}_j^k}{||\mathbf{F}_i'^{,k}|| \cdot ||\mathbf{F}_j^k||}\right), \quad \text{where} \quad j \in Q_i^k. \tag{4}$$

Here, as examplified in Fig. 4, we define Q_i^k as a window centered at pixel i with the length of a side r^k, yielding the dimensionality of $\mathbf{C}^k$ as $H \times W \times (r^k)^2$.

In our pyramidal model, the constrained cost volume enables a significant reduction in the matching ambiguities and computational loads. A relatively large window is used at coarser level to estimate a rough yet reliable affine transformation field. The estimated transformation field is further utlized as a guidance to subsequent pyramid levels, and only plausible matching candidates are provided as an input to the following regression networks. As the level goes deeper, the window becomes smaller and the local minima are likely to be avoided when learning the regression networks. The constructed cost volume is also utilized to generate pseudo supervisions through correspondence consistency check. This will be detailed in Section 5.

4.3 Pyramid Construction

To formulate our networks in a coarse-to-fine scheme, we propose two spatial pyramid models by dividing an image in a form of quad-tree rectangles (Section 4.3.1) or into multiple semantic elements of an object (Section 4.3.2).

4.3.1 Based on Uniform Rectangles (PSCNet-UR)

Following conventional pyramid models [48], [52], we start from the entire image and divide it into four non-overlapping rectangular grid cells, yielding $2^{k-1} \times 2^{k-1}$ grid cells at level k as examplified in Fig. 5. The proposed method using this model is called the pyramidal semantic correspondence networks based on uniform rectangles (PSCNet-UR).

To this end, we determine the spatial location of grid cells by equally spacing them over the input image. The distances between the nearest cells along x and y axis can be computed by dividing the width and height of an image with the number of cells along each axis, $2^{k-1} - 1$. In practice, this process can be implemented with the grid generator of STNs [24] instead of requiring learnable network parameters $\mathbf{W}_G^k$.

4.3.2 Based on Semantic Elements (PSCNet-SE)

Though PSCNet-UR may yield satisfactory results for some images, it often exposes several weaknesses especially in the presence of large appearance changes and background clutters. The regular spatial division of an image may create irrelevant patches that do not correspond to visual phrases. Furthermore, some regular grids located on the background clutters may distract from estimating reliable correspondences. To overcome this issue, we propose a new pyramid model that concentrates on semantic parts of an object. The proposed method using this model is called the pyramidal semantic correspondence networks based on semantic elements (PSCNet-SE). The irregularly identified cells in

TABLE 2
Our Network Architecture of Element Detection Networks and Affine Transformation Regression Networks

Level	Layer	Kernel	Ch I/O	Input
Element Detection Networks				
k	conv1$_g$	3×3	$(r^k)^2/256$	$\mathbf{C}^k$
	conv2$_g$	3×3	$256/128$	conv1$_g$
	conv3$_g$	3×3	$128/64$	conv2$_g$
	conv4$_g$	3×3	$64/32$	conv3$_g$
	conv5$_g$	3×3	$32/(N^k+1)$	conv4$_g$
Affine Transformation Regression Networks				
Level	Layer	Kernel	Ch I/O	Input
Cell-level1	conv1$_{c1}$	7×7	$(r^1)^2/128$	$\mathbf{C}^1$
	conv2$_{c1}$	5×5	$128/32$	conv1$_{c1}$
	conv3$_{c1}$	5×5	$32/6$	conv2$_{c1}$
Cell-level2	conv1$_{c2}$	7×7	$(r^2)^2/512$	$\mathbf{C}^2$
	conv2$_{c2}$	7×7	$512/256$	conv1$_{c2}$
	conv3$_{c2}$	7×7	$256/128$	conv2$_{c2}$
	conv4$_{c2}$	7×7	$128/32$	conv3$_{c2}$
	conv5$_{c2}$	5×5	$32/6$	conv4$_{c2}$
Cell-level3	conv1$_{c3}$	7×7	$(r^3)^2/512$	$\mathbf{C}^3$
	conv2$_{c3}$	7×7	$512/256$	conv1$_{c3}$
	conv3$_{c3}$	7×7	$256/128$	conv2$_{c3}$
	conv4$_{c3}$	5×5	$128/32$	conv3$_{c3}$
	conv5$_{c3}$	5×5	$32/6$	conv4$_{c3}$
Pixel-level	conv1$_p$	3×3	$(r^4)^2/1024$	$\mathbf{C}^4$
	conv2$_p$	3×3	$1024/256$	conv1$_p$
	conv3$_p$	3×3	$256/128$	conv2$_p$
	deconv1$_p$	3×3	$128/64$	conv3$_p$
	deconv2$_p$	3×3	$64/32$	deconv1$_p$
	deconv3$_p$	3×3	$32/6$	deconv2$_p$

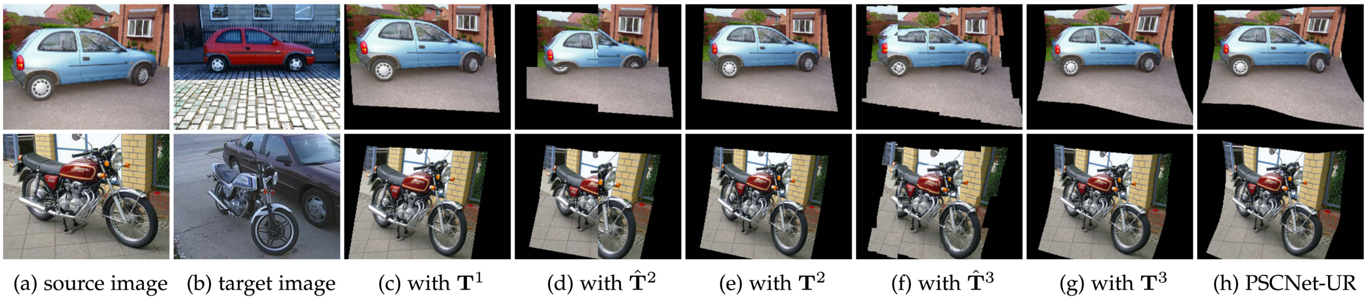


Fig. 7. Qualitative results of the PSCNet-UR at each level: (a) source image, (b) target image, warping result with estimated affine transformation fields (c) $\mathbf{T}^1$, (d) $\hat{\mathbf{T}}^2$, (e) $\mathbf{T}^2$, (f) $\hat{\mathbf{T}}^3$, (g) $\mathbf{T}^3$, and (h) $\mathbf{T}^*$ (PSCNet-UR).

PSCNet-SE model allow us to deal with larger geometric variations with improved flexibility than PSCNet-UR. Meanwhile, the adverse impact of increased flexibility can be effectively suppressed by the nature of our coarse-to-fine scheme.

As examplified in Figs. 1 and 6, a representation of PSCNet-SE model starts from a coarse cell that contains a whole object to finer cells that cover the pre-defined number of semantic elements, N^k at level k. To this end, we design several convolutional layers, called the element detection networks, and impose the equivariance constraint [44], [53] on them which enforces the discovered semantic elements to be consistently detectable across source and target images.

Specifically, we pass the constructed cost volumes $\mathbf{C}^k$ through the convolutional layers with parameters $\mathbf{W}^k_G$ to estimate N^k+1 score maps for one background and N^k semantic elements of the object, such that $\hat{\psi}^k = \mathcal{F}(\mathbf{C}^k;\mathbf{W}^k_G) \in \mathbb{R}^{H\times W\times(N^k+1)}$. The softmax layer is then applied at the end of the networks to transform raw score maps into the probability maps through the normalization over N^k+1 channels,

$$\psi^k_i(n) = \exp(\hat{\psi}^k_i(n))/\textstyle\sum_{l=0}^{N^k}\exp(\hat{\psi}^k_i(l)), \tag{5}$$

where $\psi^k_i(n)$ is the probability map of the n^{th} semantic element at k^{th} pyramidal level. Finally, the spatial coordinate of the n^{th} cell $\phi^k(n) = [\phi^k_{\mathbf{x}}(n), \phi^k_{\mathbf{y}}(n)]^T$ is computed as an expected value over the spatial coordinates i weighted by its probability $\psi^k_i(n)$

$$\begin{bmatrix}\phi^k_{\mathbf{x}}(n)\\ \phi^k_{\mathbf{y}}(n)\end{bmatrix} = \frac{1}{\sum_i \psi^k_i(n)}\begin{bmatrix}\sum_i i_{\mathbf{x}}\cdot\psi^k_i(n)\\ \sum_i i_{\mathbf{y}}\cdot\psi^k_i(n)\end{bmatrix}. \tag{6}$$

where $\sum_i \psi^k_i(n)$ is a normalization factor. This operation is fully differentiable and allow us to formulate loss functions with respect to the barycenter coordinate of the cells, similar to [54]. A detailed description of the element detection network is shown in Table 2.

Note that, the existing unsupervised part detection methods [44], [45], [53] are biased on a specific single object category since they only rely on the representations of an image collection of the specific object class ($\mathbf{F}$). In contrast, our element detection network is independent of object category as we utilize the similarity scores across an image pair ($\mathbf{C}$) allowing us to handle general object classes frequently encountered in semantic correspondence problem.

4.4 Affine Transformation Parameter Regression

4.4.1 Cell-Level Affine Transformation Regression

By passing the constrained cost volume $\mathbf{C}^k$ through successive convolutional layers, we regress the affine transformation parameters of those cells ϕ^k at level k, such that $\hat{\mathbf{T}}^k = \mathcal{F}(\mathbf{C}^k;\mathbf{W}^k_R) \in \mathbb{R}^{N^k\times 6}$ where $\mathbf{W}^k_R$ is the regression network parameter and N^k is the number of cells (for PSCNet-UR model, $N^k = 2^{k-1}\times 2^{k-1}$). It should be noted that each grid-level module estimates the residual transformation only, and thus modest number of convolutional layers are sufficient to guarantee the performance, e.g., three to six layers. Batch normalization and the ReLU layers are used after each convolution layer, as described in Table 2.

4.4.2 Pixel-Level Affine Transformation Regression

To localize fine-grained object boundaries, we finally apply a pixel-level module at the end of our networks. Similar to the cell-level modules, it also consists of feature extraction, constrained cost volume construction, and regression network. The main difference is that each pixel forms a cell, hence dense affine field is directly estimated through the

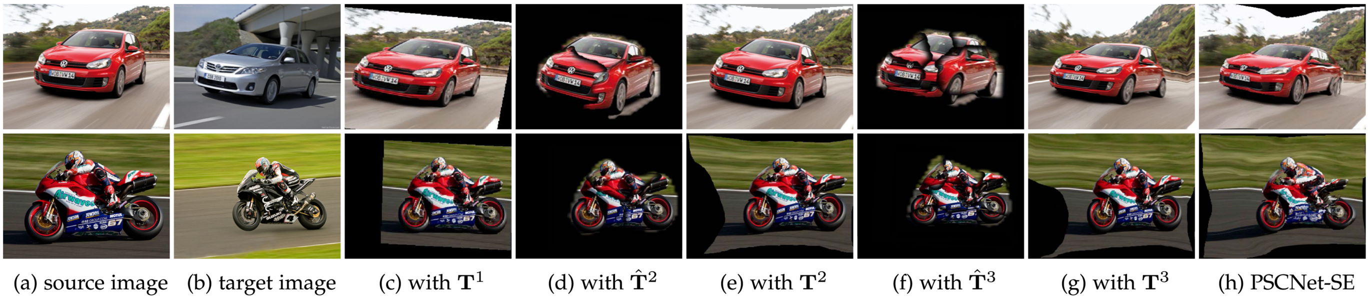


Fig. 8. Qualitative results of the PSCNet-SE at each level: (a) source image, (e) target image, warping result with estimated affine transformation fields (c) $\mathbf{T}^1$, (d) $\hat{\mathbf{T}}^2$, (e) $\mathbf{T}^2$, (f) $\hat{\mathbf{T}}^3$, (g) $\mathbf{T}^3$, and (h) $\mathbf{T}^*$ (PSCNet-SE). The discovered semantic elements at each level are visualized in Fig. 6.

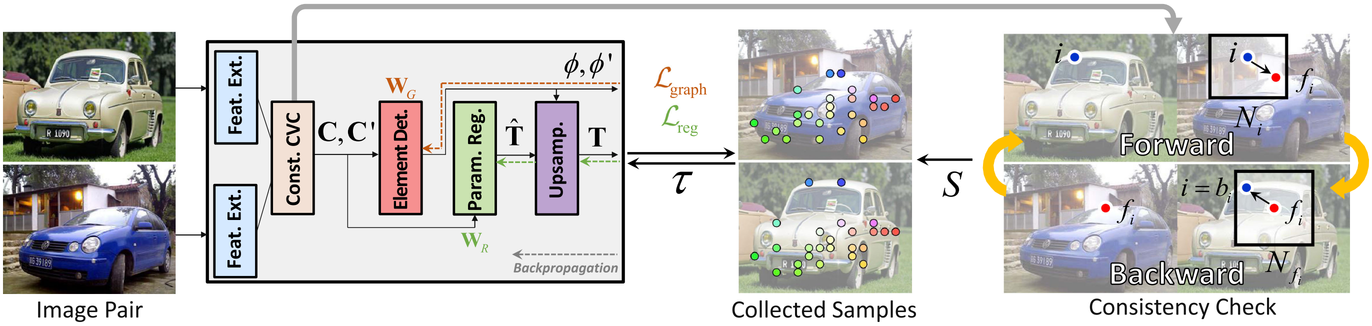


Fig. 9. Visualization of training our networks based on PSCNet-SE model. By applying the correspondence consistency check to the constructed cost volume, tentative positive samples S are collected and utilized for learning the network parameters $\mathbf{W}_G^k$ and $\mathbf{W}_R^k$.

parameter regression networks without the need of element detection networks or upsampling layer. For the regression networks, we employ an encoder-decoder style architecture that has been adopted in many pixel-level prediction tasks such as disparity estimation [8], optical flow [10], or semantic segmentation [55]. Specifically, taking a warped image I^{K+1} as an input, a constrained cost volume $\mathbf{C}^{K+1}$ is computed and the pixel-level affine field is regressed through the encoder-decoder network such that $\mathbf{T}' = \mathcal{F}(\mathbf{C}^{K+1}; \mathbf{W}_R^{K+1})$, where $\mathbf{W}_R^{K+1}$ is the pixel-level regression network parameter.

4.5 Affine Transformation Field Upsampling

While the proposed pyramidal models (PSCNet-UR and PSCNet-SE) produce coarse affine transformation fields in the cell-level modules, a dense transformation field is needed to generate the warped image that is used in the subsequent cell-module, as shown in Fig. 2. A simple nearest neighbor upsampling approach leads to blocky artifacts on the affine transformation fields as shown in Figs. 7d and 7f. To alleviate this, PSCNet-UR employs a bilinear upsampler of [24] for upsampling a coarse grid-wise affine field to the original resolution of the input image I, which is applied behind the affine transformation regression networks.

However, in contrast to PSCNet-UR model, upsampling the affine field of PSCNet-SE model cannot be directly realized with the existing upsampler [24] due to irregularly distributed semantic elements. Instead, inspired by the moving least square (MLS) [56] concept that interpolates a set of sparsely matched points with pointwise different weights, we formulate a differentiable upsampling layer that relies on all the sampling points to densify a sparse affine transformation field.

Formally, given the regressed parameters $\hat{\mathbf{T}}^k$ and their corresponding coordinates ϕ^k, the affine transformation parameters at arbitrary pixel i, namely $\mathbf{T}_i \in \mathbb{R}^{H \times W \times 6}$, can be computed as

$$\mathbf{T}_i^k = \sum\nolimits_n \hat{\mathbf{T}}_{\phi^k(n)}^k \omega(\phi_{\mathbf{x}}^k(n) - i_{\mathbf{x}}) \omega(\phi_{\mathbf{y}}^k(n) - i_{\mathbf{y}}), \tag{7}$$

where the spatially-varying weight function w is formed with coefficient ϵ as

$$w(z) = \exp(-||z||^2/2\epsilon^2). \tag{8}$$

Since the weight function w is linear, the differentiability of this operation with respect to $\hat{\mathbf{T}}^k$ can be easily derived, similar to [24]. As examplified in Figs. 7 and 8, our affine transformation field upsampling layer regularizes the affine field to be smooth, suppressing the artifacts considerably.

5 Training

5.1 Generating Progressive Supervisions

A major challenge of semantic correspondence with CNNs is the lack of ground-truth correspondence maps for training. A possible approach is to synthesize a set of image pairs transformed by applying random transformation fields as the pseudo pixel-wise ground-truth [18], [19], but this approach cannot reflect the realistic appearance variations and geometric transformations well.

Instead of using synthetically deformed imagery, we propose to generate supervisions directly from the semantically similar image pairs as shown in Figs. 9 and 10, where the correspondence consistency check is applied to the cost volume constructed at each level. Intuitively, the correspondence relation from a source image to a target image should be consistent with that from the target image to the source image. Given the constrained cost volume $\mathbf{C}^k$, the best match f_i^k is computed by searching the maximum score for each point i, $f_i^k = \mathrm{argmax}_j \, \mathbf{C}^k(i,j)$. We also compute the backward best match b_i^k for f_i^k such that $b_i^k = \mathrm{argmax}_m \, \mathbf{C}^k(m, f_i^k)$ to identify that the best match f_i^k is consistent or not. By running this

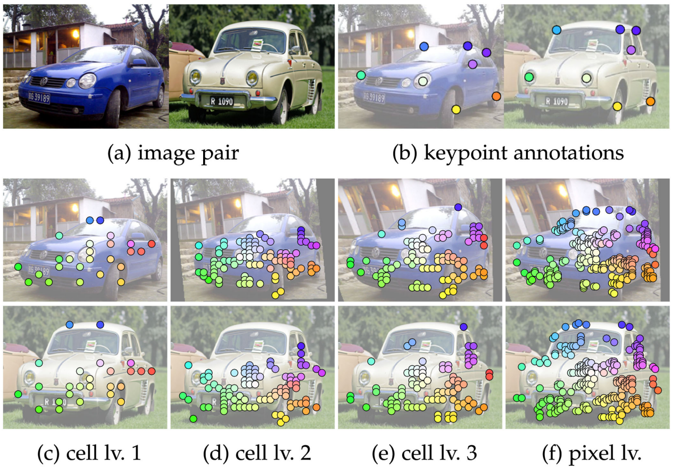
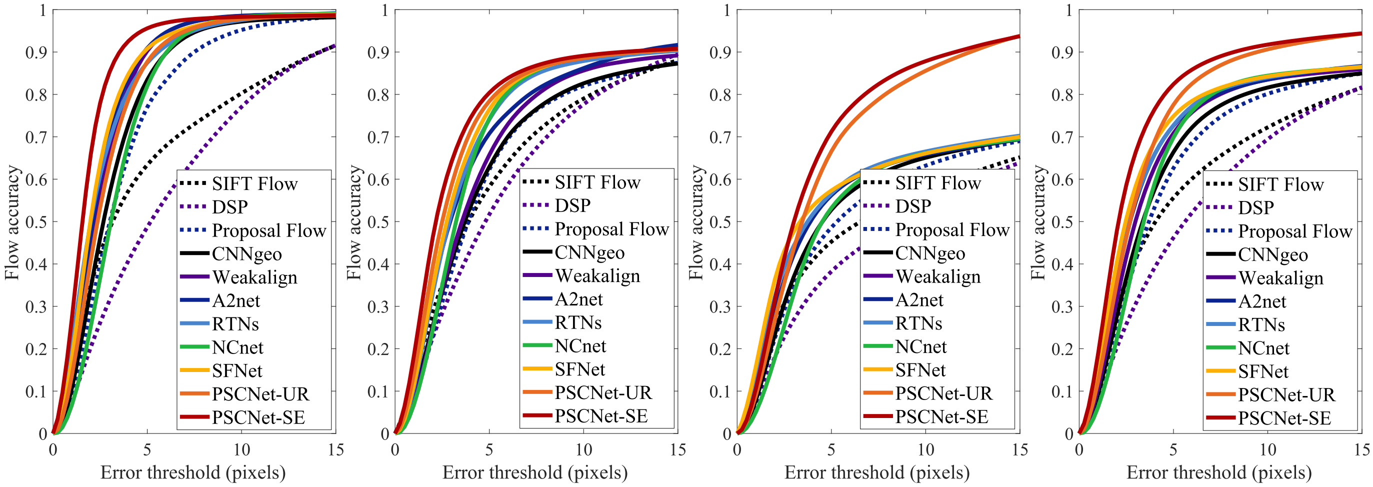

(a) image pair
(b) keypoint annotations
(c) cell lv. 1
(d) cell lv. 2
(e) cell lv. 3
(f) pixel lv.

Fig. 10. Visualization of the generated supervisions at each level: (a) source and target images, (b) keypoint annotations, (c) cell-level 1, (d) cell-level 2, (e) cell-level 3, and (f) pixel level. The tentative positive samples are color-coded where the samples of same color are supposed to match each other. (Best viewed in color.)



(a) FG3DCar (b) JODS (c) PASCAL (d) Average

Fig. 11. Flow accuracy with respect to endpoint error threshold on the TSS benchmark [27]: (a) FG3DCar, (b) JODS, (c) PASCAL, and (d) average.

consistency check along our pyramidal model, we actively collect the tentative positive samples at each level such that $S^k = \{i | i = b_i^k\}$ where sparse correspondence supervisions can be generated as $\tau_i^k = \{b_i^k - f_i^k | i \in S^k\}$. These are further interpolated in a similar way to (7) to provide pixel-wise flow supervisions, i.e., a dense correspondence map τ^k, for the equivariance loss described in Section 5.2.2.

For the accuracy of supervisions, we limit the correspondence candidate regions using binary object masks containing the target object to be matched, which are provided in most benchmarks [29], [57], [58]. Note that the cost required to annotate the object location priors is clearly less than the one required for constructing ground-truth pixel-wise semantic correspondences.

5.2 Objective Functions

5.2.1 Loss for Regression Networks

To train the parameters of our regression networks, the loss function is defined as a L_2 distance between the flows of putative positive samples and the ones computed by applying estimated affine transformation field, such that

$$\mathcal{L}_{\text{reg}} = \sum\nolimits_{i\in S^k} \frac{1}{L} \|\mathbf{T}_i^k \mathbf{i} - f_i^k\|^2, \tag{9}$$

where L is the number of collected positive samples S^k.

TABLE 3
Matching Accuracy Compared to State-of-the-Art Correspondence Techniques on the TSS Benchmark [27] When $T = 5$

Methods	FG3D	JODS	PASC.	Avg.
SIFT Flow [2]	0.632	0.509	0.360	0.500
DSP [3]	0.487	0.465	0.382	0.445
GDSP [48]	0.639	0.374	0.368	0.459
Proposal Flow [12]	0.786	0.653	0.531	0.657
TSS [27]	0.830	0.595	0.483	0.636
FCSS [14]	0.830	0.653	0.494	0.660
DCTM [17]	0.891	0.721	0.610	0.740
SCNet [13]	0.776	0.608	0.474	0.619
CNNgeo [18]	0.835	0.656	0.527	0.673
WeakAlign [20]	0.903	0.764	0.565	0.744
A2Net [19]	0.870	0.670	0.550	0.696
RTNs [21]	0.901	0.782	0.633	0.772
NCNet [22]	0.893	0.771	0.562	0.742
SFNet [23]	0.906	0.787	0.565	0.753
PSCNet-UR	0.895	0.759	0.712	0.788
PSCNet-SE	**0.952**	**0.796**	**0.723**	**0.823**

5.2.2 Losses for Element Detection Networks

For the element detection networks of PSCNet-SE model, we impose three constraints to meet the common desirable characteristics of object elements:

- Each probability map should concentrate on a discriminative local region.
- Different probability maps should highlight the different parts of the object.
- Each probability map should lie within the object.

The first constraint is formulated as a concentration loss that minimizes the variances of the probability maps with respect to their barycenters

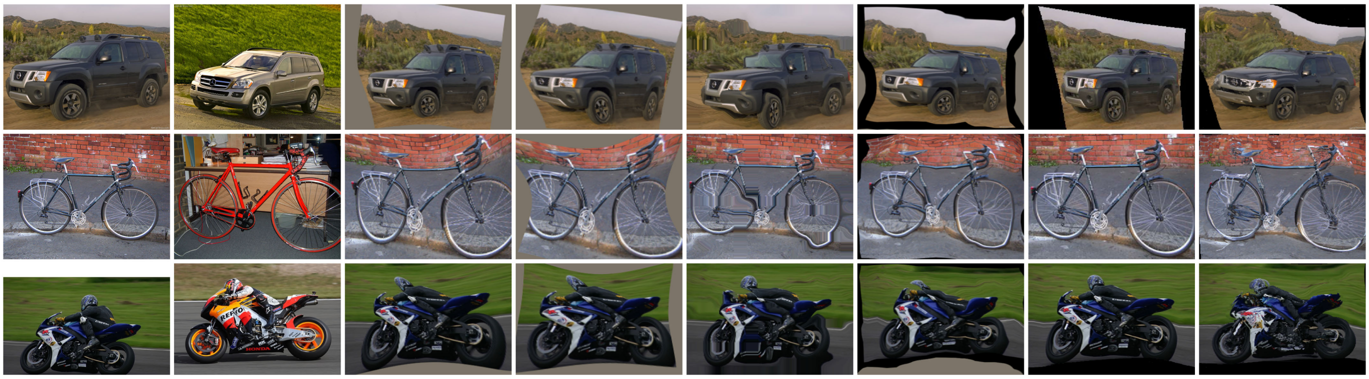
$$\mathcal{L}_{\text{con}} = \sum\nolimits_{n\in N^k} \left(\sum\nolimits_i (i-\phi^k(n))^2 \cdot \frac{\psi_i^k(n)}{\sum_i \psi_i^k(n)} \right). \tag{10}$$

For the second constraint, we define a separation loss that encourages each barycenter of elements to be far away than a margin c, such that

$$\mathcal{L}_{\text{sep}} = \sum\nolimits_n \sum\nolimits_{m\neq n} \max(0, c - ||\phi^k(n)-\phi^k(m)||_2^2). \tag{11}$$

Third, the last constraint is formulated as an objectness loss that encourages the estimated probability maps to lie within the object of interest. The binary mask of the object m in each training image is used to generate the progressive supervisions in Section 5. Here, we design the objectness loss by making use of the mask m, such that

$$\mathcal{L}_{\text{obj}} = \sum\nolimits_n -\log \sum\nolimits_i m_i \cdot \frac{\psi_i^k(n)}{\sum_i \psi_i^k(n)}, \tag{12}$$



(a) source image (b) target image (c) WeakAl. [20] (d) A2Net [19] (e) NCNet [22] (f) SFNet [23] (g) PSCNet-UR (h) PSCNet-SE

Fig. 12. Qualitative results on the TSS benchmark [27]: (a) source image, (b) target image, (c) WeakAlign [20], (d) A2Net [19], (e) NCNet [22], (f) SFNet [23], (g) PSCNet-UR, and (h) PSCNet-SE. The source images were warped to the target images using correspondences.

where $m_i = 1$ means a foreground object, and a background otherwise. Note that this binary mask is only used to compute the loss and not used at inference time.

To automatically discover object elements without the need of ground-truth, we additionally formulate a loss function that imposes equivariance constraint, such that the semantically meaningful regions should be consistently detectable with respect to the collected correspondences τ^k [44], [53]. Specifically, we supply the cost volume $\mathbf{C}^k_{i,j}$ and its reshaped version $\mathbf{C}'^{,k}_{i,j} = \mathbf{C}^k_{j,i}$ to the element detection networks and encourage the barycenter coordinates of object elements on the source and target images, ϕ^k and $\phi'^{,k}$, to be matched with the following loss function:

$$\mathcal{L}_{\text{eq}} = \sum\nolimits_n \|(\phi^k(n) - \phi'^{,k}(n)) - \tau^k_{\phi'^{,k}(n)}\|^2. \tag{13}$$

The final loss for our element detection networks is defined as a weighted sum of four loss functions, such that $\mathcal{L}_{\text{element}} = \lambda_{\text{con}}\mathcal{L}_{\text{con}} + \lambda_{\text{sep}}\mathcal{L}_{\text{sep}} + \lambda_{\text{obj}}\mathcal{L}_{\text{obj}} + \lambda_{\text{eq}}\mathcal{L}_{\text{eq}}$. Note that similar loss functions have been used in the semantic part detection literature [44], [45], [53], but they have limited generalization ability due to the dependence on a particular object category and the usage of synthetically generated ground-truth correspondences. In contrast, our method is independent of object category as we utilize the similarity scores across an image pair ($\mathbf{C}$) rather than the representations of an image collection of the specific object class ($\mathbf{F}$). Furthermore, we leverage more realistic supervisory signals for training by obtaining correspondences from the semantically similar image pairs. Algorithms 1 and 2 provide an overall summary of PSCNet-UR and PSCNet-SE models, respectively.

TABLE 4
Matching Accuracy Compared to State-of-the-Art Correspondence Techniques on the Proposal Flow-WILLOW Benchmark [12]

Methods	PCK		
	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.15$
SIFT Flow [2]	0.247	0.380	0.504
Proposal Flow [12]	0.284	0.568	0.682
FCSS [14]	0.354	0.532	0.681
DCTM [17]	0.381	0.610	0.721
SCNet [13]	0.359	0.601	0.692
CNNgeo [18]	0.312	0.586	0.712
WeakAlign [20]	0.370	0.702	0.799
A2Net [19]	0.363	0.688	0.844
RTNs [21]	0.413	0.719	0.862
NCNet [22]	0.388	0.737	0.857
SFNet [23]	0.385	0.739	0.860
PSCNet-UR	0.381	0.720	0.851
PSCNet-SE	**0.426**	**0.751**	**0.880**

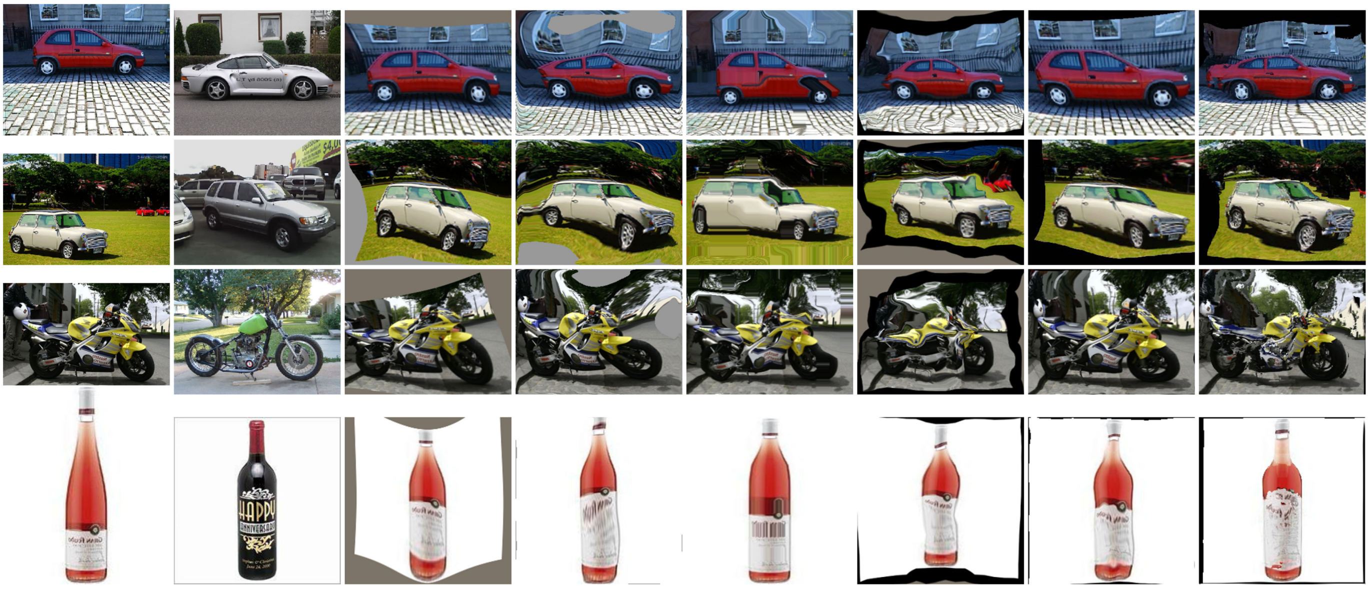
Algorithm 1. PSCNet-UR Framework

```
Input: images $I$, $I'$
Output: network parameters $\mathbf{W}_F$, $\mathbf{W}^k_R$, dense affine field $\mathbf{T}^*$
Parameters: pyramid levels $K$, window $Q^k$, indices $M^k$
Compute convolutional activations of target image $I'$
  for $k = 1 : K$ do
    if $k > 1$ do
      Warp $I^{k-1}$ with $\mathbf{T}^{k-1}$ to compute $I^k$
    end if
    /* Hierarchical Feature Extraction */
    Extract features $\mathbf{F}^k$, $\mathbf{F}'^{,k}$ with $M^k$ from $I^k$, $I'^{,k}$
    /* Constrained Correlation Volume */
    Construct $\mathbf{C}^k$ within the constrained window $Q^k$
    [Only when training]: Collect $S^k$ from $\mathbf{C}^k$ and generate supervisions $\tau$
    /* Affine Geometry Regression */
    Estimate affine transformation parameters $\hat{\mathbf{T}}^k$
    /* Affine Transformation Field Upsampling */
    Compute $\mathbf{T}^k$ by applying bilinear upsampler to $\hat{\mathbf{T}}^k$
  end for
Estimate pixel-level affine fields $\mathbf{T}' = \mathcal{F}(\mathbf{C}^{K+1}; \mathbf{W}^{K+1}_R)$
Compute $\mathbf{T}^*_i$ as $\mathbf{M}(\mathbf{T}^*_i) = \prod_{n \in \{1,\ldots,K\}} \mathbf{M}(\mathbf{T}^n) \cdot \mathbf{M}(\mathbf{T}'_i)$
```

5.3 Training Details

To learn our networks, we adopt a 2-step training technique, similar to [20]. In the first step, our networks were learned with synthetically generated image pairs from the Pascal VOC 2012 segmentation dataset [59]. In the second step, we finetune this pretrained network with semantically similar image pairs provided from the training set of the Proposal Flow-PASCAL dataset [28].

We used the Adam optimizer [60] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. To determine the weighting parameters $\{\lambda_{\text{con}}, \lambda_{\text{sep}}, \lambda_{\text{obj}}, \lambda_{\text{eq}}\}$, we used the grid search and chose the ones that produce the best result on the validation split of



(a) source image (b) target image (c) WeakAl. [20] (d) RTNs [21] (e) NCNet [22] (f) SFNet [23] (g) PSCNet-UR (h) PSCNet-SE

Fig. 13. Qualitative results on Proposal Flow-WILLOW benchmark [12]: (a) source image, (b) target image, (c) WeakAlign [20], (d) RTNs [21], (e) NCNet [22], (f) SFNet [23], (g) PSCNet-UR, and (h) PSCNet-SE. The source images were warped to the target images using correspondences.

ProposalFlow-PASCAL dataset [28] as $\{1, 10, 1, 100\}$. We referred to the ablation study of other equivariance-based methods [44], [45], [53] when setting the initial values of parameters for the grid search. The margin c is chosen similarly using the validation split of the ProposalFlow-PASCAL dataset [28] as 0.03. We initialize each regression networks to estimate affine transformation parameters as $[\mathbf{I}_{2\times2}, \mathbf{0}_{2\times1}]$ before the training starts. M-estimator sample and consensus (MSAC) [61] is applied to the putative positive samples of level 1 (i.e., S^1), where the maximum number of trials is set to 5,000.

6 Experimental Results

6.1 Experimental Settings

For the feature extraction networks in each module, we used the ImageNet pretrained ResNet-101 [51] with their network parameters. According to the convergence analysis in Section 6.4.1, we used three cell-level modules ($K = 3$) and set the number of semantic elements N^k to $\{1, 3, 9\}$. For the sampling indices M^k in the feature extraction step, we sampled convolutional activations after intermediate pooling layers, such as $\{''C_{5-3}'', ''C_{5-3}, C_{4-3}'', ''C_{5-3}, C_{4-3}, C_{3-3}''\}$. The length of search window r^k is set to the ratio of the whole search space, i.e., the feature map of the target image, decreasing as the level goes deeper such that $\{1, 1/10, 1/15\}$.

TABLE 5
Matching Accuracy Compared to State-of-the-Art Correspondence Techniques on the Proposal Flow-PASCAL Benchmark [28]

Methods	PCK		
	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.15$
SIFT Flow [2]	0.292	0.584	0.762
Proposal Flow [12]	0.314	0.625	0.795
FCSS [14]	0.329	0.659	0.798
DCTM [17]	0.342	0.696	0.802
SCNet [13]	0.362	0.722	0.820
CNNgeo [18]	0.410	0.695	0.804
WeakAlign [20]	0.490	0.748	0.840
A2Net [19]	0.428	0.708	0.833
RTNs [21]	0.552	0.759	0.852
NCnet [22]	0.523	0.789	0.860
SFNet [23]	0.500	0.787	**0.889**
PSCNet-UR	0.558	0.776	0.844
PSCNet-SE	**0.598**	**0.803**	0.885

Algorithm 2. PSCNet-SE Framework

```
Input: images $I$, $I'$
Output: network parameters $\mathbf{W}_F$, $\mathbf{W}_G^k$, $\mathbf{W}_R^k$, dense affine field $\mathbf{T}^*$
Parameters: pyramid levels $K$, window $Q^k$, numbers $N^k$, indices $M^k$
Compute convolutional activations of target image $I'$
for $k = 1 : K$ do
    Step 2-5 in Algorithm 1.
    Construct $\mathbf{C}'^{,k}$ as $\mathbf{C}^k_{j,i} = \max(0, \mathbf{F}'^{,k}_j \cdot \mathbf{F}^k_i)$, where $i \in Q^k_j$.
    /* Element Detection */
    Compute barycenter coordinates $\phi^k$, $\phi'^{,k}$ from $\mathbf{C}^k$, $\mathbf{C}'^{,k}$
    /* Affine Geometry Regression */
    Estimate affine transformation parameters $\hat{\mathbf{T}}^k$
    /* Affine Transformation Field Upsampling */
    Compute dense affine transformation field $\mathbf{T}^k$ using (7)
end for
Estimate pixel-level affine fields $\mathbf{T}' = \mathcal{F}(\mathbf{C}^{K+1}; \mathbf{W}_R^{K+1})$
Compute $\mathbf{T}^*_i$ as $\mathbf{M}(\mathbf{T}^*_i) = \prod_{n\in\{1,\ldots,K\}} \mathbf{M}(\mathbf{T}^n) \cdot \mathbf{M}(\mathbf{T}'_i)$
```

In the following, we comprehensively evaluated our method in comparison with the latest methods including CNNgeo [18], WeakAlign [20], A2Net [19], RTNs [21], NCNet [22], and SFNet [23]. Note that all of our baseline methods [18], [19], [20], [21], [22], [23] employed pre-trained VGGNet [62] or ResNet [51] as a backbone network. Some of them [20], [21], [22] used the image pairs of ProposalFlow-PASCAL dataset [28] as their training data, while

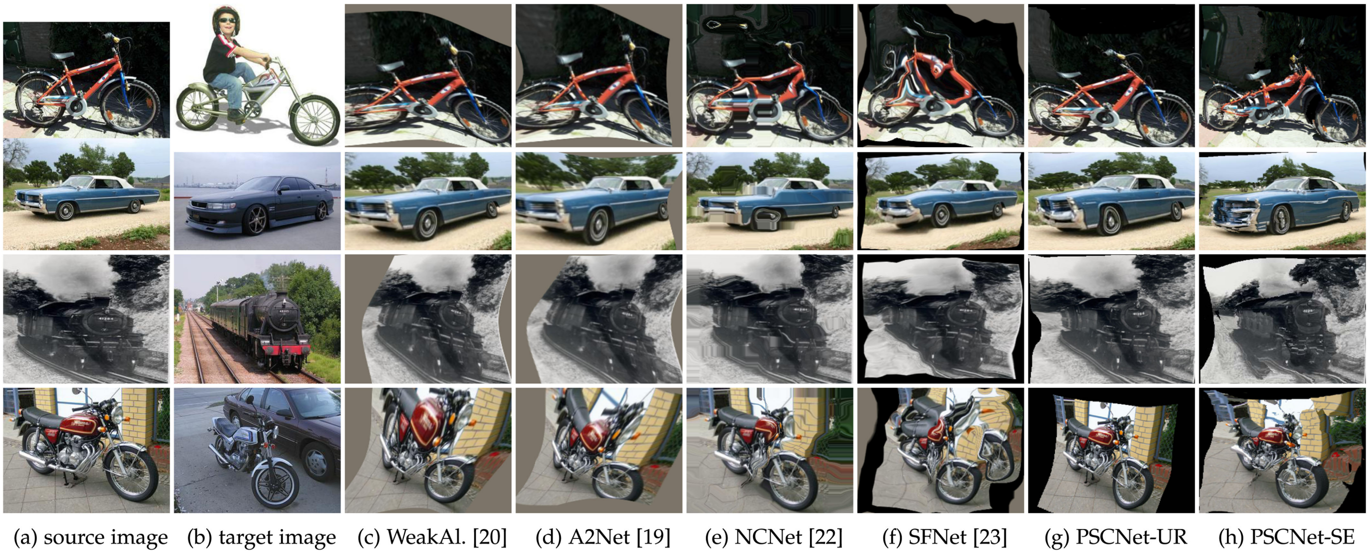


Fig. 14. Qualitative results on Proposal Flow-PASCAL benchmark [28]: (a) source image, (b) target image, (c) WeakAlign [20], (d) A2Net [19], (e) NCNet [22], (f) SFNet [23], (g) PSCNet-UR, and (h) PSCNet-SE. The source images were warped to the target images using correspondences.

others [18], [19], [23] generate synthetic image pairs by applying random transformations to a single image. Among the baselines, solely [20] employed 2-step training technique that first learns with synthetically generated image pairs and then fine-tunes on real image pairs of standard dataset.

6.2 Matching Results

6.2.1 TSS Benchmark

We evaluated PSCNet-UR and PSCNet-SE compared to other state-of-the-art methods on the TSS benchmark [27], which consists of 400 image pairs divided into three groups: FG3DCar, JODS, and PASCAL. Flow accuracy was measured by computing the proportion of foreground pixels with an absolute flow endpoint error that is smaller than a certain threshold T, after resizing images so that its larger dimension is 100 pixels.

Fig. 11 shows the flow accuracy with varying error threshold T. Table 3 summarizes the matching accuracy for state-of-the-art techniques at the fixed threshold ($T = 5$ pixels). Fig. 12 shows qualitative results by warping source images with the estimated correspondence fields.

TABLE 6
Matching Accuracy Compared to State-of-the-Art Correspondence Techniques on the Caltech-101 Dataset [29]

Methods	LT-ACC	IoU	LOC-ERR
SIFT Flow [2]	0.75	0.48	0.32
DSP [3]	0.77	0.47	0.35
Proposal Flow [12]	0.78	0.50	0.25
FCSS [14]	0.80	0.50	0.21
DCTM [17]	0.84	0.53	**0.18**
SCNet [13]	0.79	0.51	0.25
CNNgeo [18]	0.83	0.61	0.25
WeakAlign [20]	0.85	0.63	0.24
A2Net [19]	0.80	0.57	0.25
RTNs [21]	0.86	0.65	0.21
NCNet [22]	0.85	0.60	0.22
SFNet [23]	0.88	**0.67**	0.21
PSCNet-UR	0.87	0.65	0.21
PSCNet-SE	**0.90**	**0.67**	0.21

Our method outperforms especially when the error thershold is small. This clearly demonstrates the advantage of our coarse-to-fine approach in terms of both localization precision and semantic invariance. As shown in Figs. 11, 12, and Table 3, our results have shown highly improved performance qualitatively and quantitatively compared to the methods [18], [19], [20] that rely on global transformation parameters, particularly in capturing fine-grained object details. Moreover, in contrast to the methods [21], [22], [23] that estimate locally-varying transformation fields without explicit consideration of global deformation, our methods naturally impose the smoothness constraint on the affine transformation fields through the proposed spatial pyramid models. Additionally, the improved performance of PSCNet-SE compared to PSCNet-UR reveals the effectiveness of the new pyramid model that considers semantic structure of an object.

6.2.2 Proposal Flow-WILLOW Benchmark

We also evaluated our methods on the Proposal Flow-WILLOW benchmark [12], which provides 900 image pairs of 4 object sub-classes with 10 keypoint annotations for each image. For the evaluation metric, we used the probability of correct keypoint (PCK) between flow-warped keypoints and the ground truth [12], [63]. The warped keypoints are deemed to be correctly predicted if they lie within $\alpha \cdot \max(h_b, w_b)$ pixels of the ground-truth keypoints for $\alpha \in [0, 1]$, where h_b and w_b are the height and width of the object bounding box, respectively.

The PCK values were measured for different correspondence techniques in Table 4 and Fig. 13 shows qualitative results by warping source images with the estimated correspondence fields. Our method exhibits outperforming performance compared to the state-of-the-art correspondence techniques. Our PSCNet-SE method is especially effective in the presence of severe appearance and shape variations compared to other methods.

6.2.3 Proposal Flow-PASCAL Benchmark

We also evaluated our methods on the Proposal Flow-PASCAL benchmark [28], which contains 1,351 image pairs for

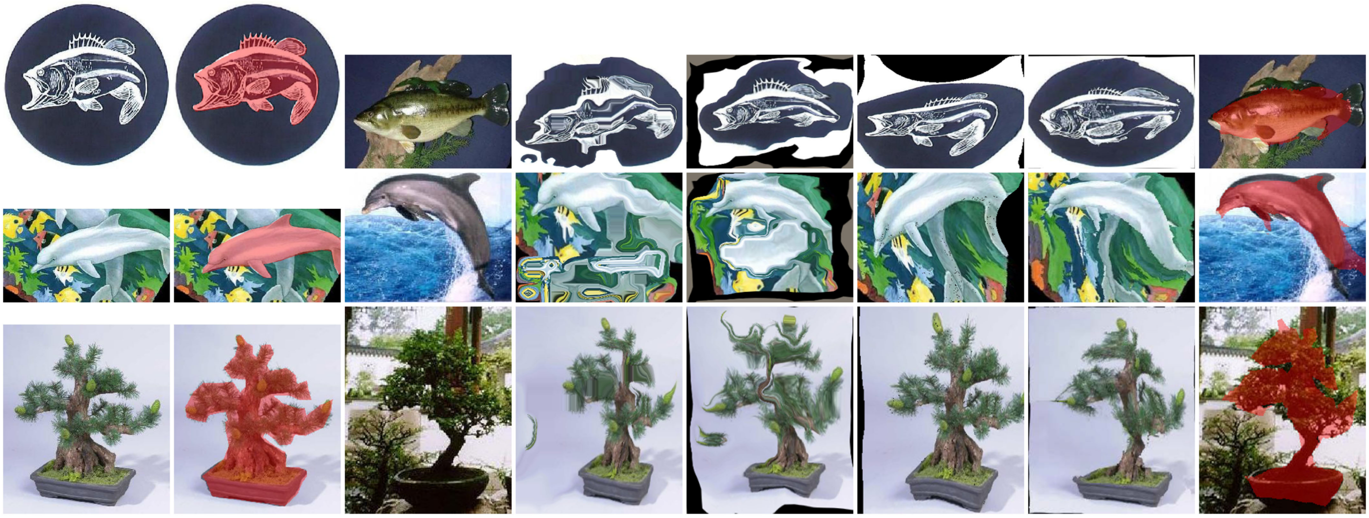

(a) source image (b) source mask (c) target image (d) NCNet [22] (e) SFNet [23] (f) PSCNet-UR (g) PSCNet-SE (h) Trans. mask

Fig. 15. Qualitative results on Caltech 101 benchmark [29]: (a) source image, (b) source mask, (c) target image, (d) NCNet [22], (e) SFNet [23], (f) PSCNet-UR, (g) PSCNet-SE, and (h) Transferred mask. The source images and their masks were warped to the target images using correspondences.

TABLE 7
Per-Class Matching Accuracy on SPair-71k Dataset [30] Compared to State-of-the-Art Correspondence Techniques

Methods		aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dog	horse	moto	person	plant	sheep	train	tv	all
Fine-tuned	CNNGeo [18]	23.4	16.7	40.2	14.3	36.4	27.7	26.0	32.7	12.7	27.4	22.8	13.7	20.9	21.0	17.5	10.2	30.8	34.1	20.6
	A2Net [19]	22.6	18.5	42.0	16.4	37.9	30.8	26.5	35.6	13.3	29.6	24.3	16.0	21.6	22.8	20.5	13.5	31.4	36.5	22.3
	WeakAlign [20]	22.2	17.6	41.9	15.1	**38.1**	27.4	**27.2**	31.8	12.8	26.8	22.6	14.2	20.0	22.2	17.9	10.4	32.2	35.1	20.9
	NCNet [22]	17.9	12.2	32.1	11.7	29.0	19.9	16.1	39.2	9.9	23.9	18.8	15.7	17.4	15.9	14.8	9.6	24.2	31.1	20.1
	SFNet [64]	**26.9**	17.2	45.5	14.7	38.0	22.2	16.4	**55.3**	13.5	33.4	**27.5**	**17.7**	20.8	21.1	16.6	15.6	32.3	35.9	26.3
	PSCNet-UR	23.7	18.1	43.3	16.4	36.8	27.9	27.0	32.5	13.7	27.0	23.1	14.9	21.2	23.4	20.9	13.4	32.8	36.1	23.9
	PSCNet-SE	25.9	**18.9**	**47.7**	**17.5**	38.0	**28.8**	27.1	39.9	**14.9**	**33.0**	25.8	16.6	**24.4**	**26.1**	**21.1**	**15.9**	**33.5**	**37.1**	**26.5**
Transferred	CNNGeo [18]	21.3	15.1	34.6	12.8	31.2	26.3	24.0	30.6	11.6	24.3	20.4	12.2	19.7	15.6	14.3	9.6	28.5	28.8	18.1
	A2Net [19]	20.8	17.1	37.4	13.9	33.6	29.4	26.5	34.9	12.0	26.5	22.5	13.3	21.3	20.0	16.9	11.5	28.9	31.6	20.1
	WeakAlign [20]	23.4	17.0	41.6	14.6	**37.6**	28.1	26.6	32.6	12.6	27.9	23.0	13.6	21.3	22.2	17.9	10.9	31.5	34.8	21.1
	NCNet [22]	24.0	16.0	45.0	13.7	35.7	25.9	19.0	50.4	14.3	32.6	27.4	**19.2**	21.7	20.3	**20.4**	13.6	33.6	**40.4**	26.4
	SFNet [64]	27.3	17.2	**47.2**	14.7	36.7	21.4	16.5	**56.4**	13.6	**32.9**	25.4	17.4	19.9	19.5	15.9	15.9	33.2	35.1	26.0
	PSCNet-UR	24.8	17.5	43.2	14.3	36.7	29.1	27.0	42.8	12.4	29.5	25.7	15.3	23.8	21.5	18.1	12.6	33.8	36.8	24.1
	PSCNet-SE	**28.3**	**17.7**	45.1	**15.1**	37.5	**30.1**	**27.5**	47.4	**14.6**	32.5	**26.4**	17.7	**24.9**	**24.5**	19.9	**16.9**	**34.2**	37.9	**27.0**
Fully-sup.	HPF []	25.2	18.9	52.1	15.7	38.0	22.8	19.1	52.9	17.9	33.0	32.8	20.6	24.4	27.9	21.1	15.9	31.5	35.6	28.2

TABLE 8
Matching Accuracy Compared to State-of-the-Art Correspondence Techniques on SPair-71k Dataset [30] That are Released After the Time of Submission (Sep. 2019)

Methods		Venue	Supervision	PCK
				$\alpha = 0.1$
Weakly supervised	PSCNet-SE	-	PF-PASCAL (I)	27.0
	DHPF [42]	ECCV'20	PF-PASCAL (I)	28.5
	GSF [65]	ECCV'20	PF-PASCAL (I)	36.1
Fully supervised	HPF [30]	ICCV'19	SPair-71k (K)	28.2
	DHPF [42]	ECCV'20	SPair-71k (K)	37.3
	SCOT [43]	CVPR'20	SPair-71k (K)	35.6

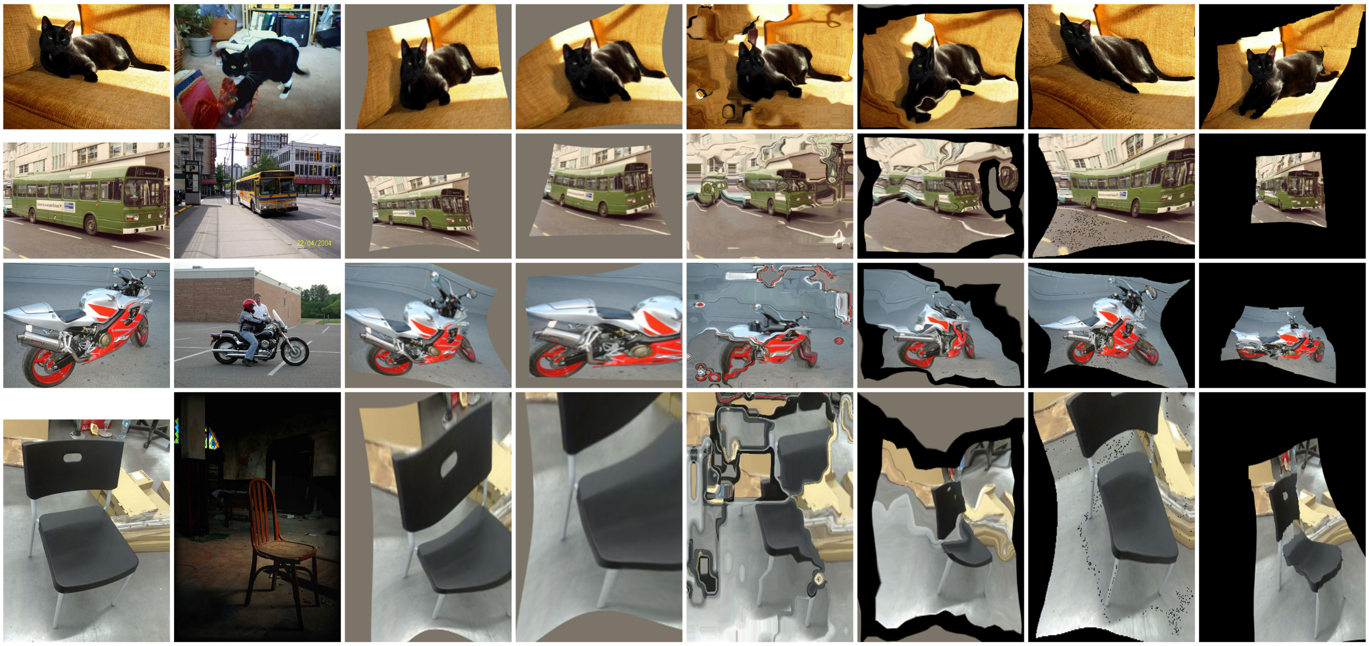
We denote "I" and "K" by the used type of supervision such that image pair and keypoints, respectively.

20 object categories with PASCAL keypoint annotations. For the evaluation metric, we used the PCK between flow-warped keypoints and the ground truth [12] as in the experiments on the Proposal Flow-WILLOW benchmark [12].

The PCK values were measured for different correspondence techniques in Table 5 and Fig. 14 shows qualitative results for dense flow estimation. Our method exhibits outstanding performance compared to state-of-the-art dense correspondence estimation methods. Our PSCNet-SE method again was found to be reliable especially under challenging correspondence settings.

6.2.4 Caltech-101 Dataset

The evaluation was also performed on the Caltech-101 dataset [29] with the image pairs used in [20] which provides the



(a) source image (b) target image (c) WeakAl. [20] (d) A2Net [19] (e) NCNet [22] (f) SFNet [23] (g) PSCNet-UR (h) PSCNet-SE

Fig. 16. Qualitative results on SPair-71k dataset [30] [27]: (a) source image, (b) target image, (c) WeakAlign [20], (d) A2Net [19], (e) NCNet [22], (f) SFNet [23], (g) PSCNet-UR, and (h) PSCNet-SE. The source images were warped to the target images using correspondences.

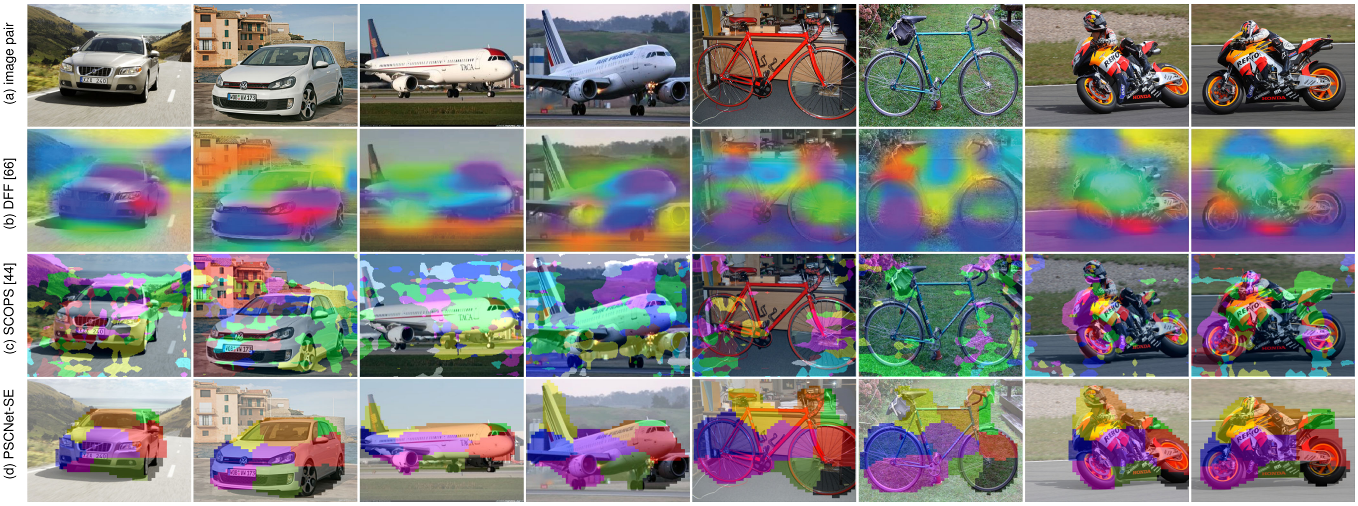


Fig. 17. Visualization of the detected 9 semantic elements on the TSS benchmark [27]: For a given (a) image pair, the semantic elements detected with (b) DFF [66], and (c) PSCNet-SE. The elements with the same color are supposed to match each other.

images of 101 object categories with ground-truth object masks. For the evaluation, we used the 1,515 image pairs used in [13], [20], i.e., 15 image pairs for each object category. Following the experimental protocol in [52], matching accuracy was evaluated with three metrics: the label transfer accuracy (LT-ACC), the intersection-over-union (IoU) metric, and the localization error (LOC-ERR) of corresponding pixel positions. Table 6 summarizes the matching accuracy compared to state-of-the-art methods. As shown in Fig. 15 and Table 6, our PSCNet-UR and PSCNet-SE are competitive to the state-of-the-art techniques in terms of LT-ACC and IoU metrics.

Note that compared to other benchmarks described above, the Caltech-101 dataset [29] provides image pairs from more diverse classes, enabling the performance evaluation under more general correspondence settings.

6.2.5 SPair-71k Dataset

The evaluation was also performed on the SPair-71k benchmark [30] that includes 70,958 image pairs of 18 object categories from PASCAL 3D+ [67] and PASCAL VOC 2012 [57], providing 12,234 pairs for testing. This benchmark is more challenging than other datasets described above [12], [27], [28], [29] as the provided image pairs cover diverse variations in terms of viewpoint, scale, truncation, and occlusion. For the evaluation metric, we used the PCK with respect to the object bounding box by setting the threshold to 0.1.

Table 7 reports the quantitative performance with respect to different object categories. The qualitative results are visualized in Fig. 16. In terms of average PCK score reported

TABLE 9
Part-Level IoU Compared to State-of-the-Art Co-Segmentation Technique on the TSS Benchmark [27][1]

Methods	Mask	N	FG3D	JODS	PASC.	Avg.
DFF [66]	✗	3	58.1	49.6	46.3	51.3
		9	49.8	41.6	39.4	43.6
SCOPS [44]	✓	4	21.7	17.8	19.5	19.7
		8	18.1	16.6	17.2	17.1
PSCNet-SE	✓	3	**63.8**	**57.8**	**55.7**	**59.2**
		9	**61.8**	**55.6**	**49.9**	**55.8**
wo/ $\mathcal{L}_{\mathrm{con}}$		3	59.2	52.6	49.6	53.8
		9	58.3	49.4	45.3	51.0
wo/ $\mathcal{L}_{\mathrm{sep}}$		3	60.6	54.9	52.9	56.1
		9	58.7	52.1	47.3	52.7
wo/ $\mathcal{L}_{\mathrm{obj}}$		3	61.2	55.6	53.6	56.8
		9	58.3	53.9	48.4	53.5
wo/ $\mathcal{L}_{\mathrm{eq}}$		3	57.4	52.0	50.0	53.3
		9	55.6	50.4	44.9	50.3

We denote "Mask" and "N" by the additionally used object mask information and the number of semantic elements, respectively.

TABLE 10
Object-Level IoU Compared to State-of-the-Art Co-Segmentation Technique on the TSS Benchmark [27]

Methods	Mask	N	FG3D	JODS	PASC.	Avg.
DFF [66]	✗	3	83.2	70.9	66.1	73.4
		9	71.9	59.4	56.3	62.3
SCOPS [45]	✓	4	51.7	47.6	44.9	48.1
		8	48.3	47.0	45.8	47.0
PSCNet-SE	✓	3	**91.2**	**82.6**	**79.6**	**84.5**
		9	**88.3**	**79.4**	**71.3**	**79.7**

We denote "Mask" and "N" by the additionally used object mask information and the number of semantic elements, respectively.

TABLE 11
Ablation Study for Different Components of Our Network Architecture

Methods	FG3D	JODS	PASC.	Avg.
PSCNet-UR wo/pool.	0.862	0.648	0.661	0.691
PSCNet-UR wo/Q^k	0.850	0.637	0.659	0.682
PSCNet-UR wo/up.	0.869	0.704	0.680	0.751
PSCNet-UR	**0.895**	**0.759**	**0.712**	**0.788**
PSCNet-SE wo/pool.	0.890	0.737	0.689	0.739
PSCNet-SE wo/Q^k	0.883	0.716	0.667	0.735
PSCNet-SE wo/up.	0.916	0.751	0.689	0.785
PSCNet-SE	**0.952**	**0.796**	**0.723**	**0.823**

We evaluated their flow accuracy on the TSS benchmark [27] when $T = 5$.

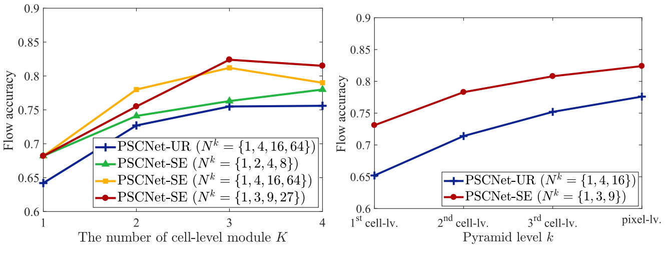
TABLE 12
Ablation Study for Different Loss Functions When Training PSCNet-SE

$\mathcal{L}_{\mathrm{con}}$	$\mathcal{L}_{\mathrm{sep}}$	$\mathcal{L}_{\mathrm{obj}}$	$\mathcal{L}_{\mathrm{eq}}$	FG3D	JODS	PASC.	Avg.
-	✓	✓	✓	0.881	0.683	0.663	0.742
✓	-	✓	✓	0.892	0.702	0.677	0.757
✓	✓	-	✓	0.927	0.735	0.684	0.782
✓	✓	✓	-	0.862	0.678	0.651	0.739
✓	✓	✓	✓	**0.952**	**0.796**	**0.723**	**0.823**

We evaluated their flow accuracy on the TSS benchmark [27] when $T = 5$.

in Table 7, the results of PSCNet exhibits a competitive performance to the state-of-the-art techniques with or without finetuning on Spair-71k dataset [30], indicating that our coarse-to-fine framework is effective in resolving large variations. In the presence of non-rigid deformations in bird, dog, and cat classes, PSCNet-UR has shown limited performance since its regular spatial division of an image often have difficulties in dealing with background clutters and complex transformations. In contrast, PSCNet-SE yields a large PCK gain for non-rigid classes in Table 7, demonstrating the benefits of a pyramid model that concentrates more on the semantic parts of an object. Note that, taking advantages from the active usage of ground-truth annotations in SPair-71k benchmark [41], HPF [30] yields better performances on non-rigid object classes such as bird and cat where our weakly-supervised learning might be fragile in presence of large geometry differences.

We also summarized current state-of-the-art performances on SPair-71k dataset [41] in Table 8 that are reported after the submission of this manuscript (Sep. 2019). To cover wide range of intra-class variations of SPair-71k dataset [41], the keypoint annotations provided from the training set are actively utilized, yielding better performance.



(a) (b)

Fig. 18. Convergence analysis of PSCNet-UR and PSCNet-SE on the TSS benchmark [27]: (a) with differnet numbers of cell-level module K, and (b) with different numbers of pyramid level k when K is fixed to 3.

6.3 Co-Part Segmentation Results

We also conducted an evaluation of our element detection networks used in the PSCNet-SE model on the TSS benchmark [27], comparing with the state-of-the-art method [66] for object part segmentation task. The discovered semantic elements are visualized in Fig. 17, where the probability maps ψ estimated from our element detection networks are color-coded. Following the current best practice [45], [46] to examine the quality of the detected parts in an unsupervised manner, we measure the performance with the IoU metric of part-aggregated segmentation mask in Table 10. We further report the part-level localization accuracy by evaluating the degree of consistency between the detected elements across image pair. To this end, we warp the elements on the source image to the ones of target image using the ground-truth dense correspondences provided in TSS benchmark [27] and then compute the average IoU scores

[1] We note that the publicly released models of SCOPS [45] detect only 4 or 8 object parts trained on bird and human face category, respectively.

TABLE 13
Runtime Comparison for the Images of Size 256×256 on TSS Benchmark [27]

Methods	WeakAl. [20]	A2Net [19]	RTNs [21]	NCNet [22]	SFNet [23]	PSCNet-UR	PSCNet-SE	
							Sequantial	Parallel
Runtime (ms)	143	157	380	243	172	220	420	248

over them. We simply excluded the regions where the the ground-truth correspondences are not given.

As reported in Tables 9 and 10, mIoU accuracy of our element detection networks outperforms DFF [66] by a large margin thanks to the various constraints formulated in Section 5.2.2 and the generated supervisory signals that reflect realistic appearance and geometric variations. The qualitative results in Fig. 17 show that the discovered semantic elements by our method are robust to various appearance and viewpoint variations, while SCOPS [45] is inherently limited in handling diverse object classes, such as car, horse, motorbike, and train in TSS benchmark.

6.4 Ablation Study

6.4.1 Convergence Analysis

We first analyze the convergence of our methods on the TSS benchmark [27]. All the flow accuracies are measured at the fixed threshold ($T = 5$ pixels). Fig. 18a shows the flow accuracy of PSCNet-UR and PSCNet-SE model for different numbers of cell-level module K. While matching accuracies were improved by enlarging the number of cell-level module until $K = 3$, we observe that using more modules (e.g., $K = 4$) reduces matching accuracy since the fine division of the semantic elements may lack of the contextual information of an object. The performances of PSCNet-SE model with respect to the number of cells N^k indicate the trade-off between the number of semantic elements and their matching ambiguities. Based on these experiments, we set $K = 3$ and $N^k = \{1, 3, 9\}$. Fig. 18b shows the tendency of flow accuracy in the intermediate results when the number of cell-level module is fixed to 3.

As expected, after estimating the global affine transformation robust to geometric variations at level 1, the localization ability has been improved progressively as the level goes deeper.

6.4.2 Network Architecture

To examine the effects of our components, we report the qualitative assessment in Table 11 when one of our components is removed from the network architecture; the pooling of multi-scale feature maps (wo/pool.), the constrained search range (wo/Q^k), and the upsampling of coarse affine transformation field (wo/up.). The evaluations were conducted on the TSS benchmark [27] at the fixed threshold ($T = 5$ pixels). As shown in Table 11, the flow accuracies of "wo/pool." and "wo/Q^k" highlight the importance of exploiting different levels of features to resolve local ambiguities at the feature extraction stage, and reducing the matching ambiguity with the constrained search spaces at the cost volume construction stage, respectively. Additionally, the results of "wo/up." reveal the significance of the affine transformation field regularization.

6.4.3 Loss Function

To validate the effectiveness of the utilized loss functions described in Section 5.2.2, we conduct a series of ablation studies on TSS benchmark [27] when learned with different loss functions. In Tables 9 and 12, we report the performances on co-part segmentation task and semantic correspondence task, respectively. As reported in both tables, the gain of mean IoU scores and flow accuracy with respect to $\mathcal{L}_{\mathrm{eq}}$ demonstrates that the equivariance constraint plays the most important role in identifying consistently meaningful parts and providing well-defined object structure for constructing pyramid model. On the other hand, with respect to $\mathcal{L}_{\mathrm{obj}}$, the modest degradation indicates that our element detection networks can still capture the objectness to some extent without using $\mathcal{L}_{\mathrm{obj}}$. We attribute this to the formulation of element detection networks that normalize the probability scores over $N^k + 1$ channel (one background and N^k semantic elements of the object). As the objective of $\mathcal{L}_{\mathrm{eq}}$ imposes equivariance constraint on N^k scores, the normalization automatically encourages the remaining score to highlight inconsistent regions, i.e., background.

6.4.4 Runtime Analysis

We report the runtime of our models in comparison to the state-of-the-art methods based on the global transformation [19], [20] or locally-varying transformation [21], [22], [23]. All evaluations were performed with a Nvidia GTX 1080Ti and Intel Core i7-3770 CPU at 3.40 GHz. For PSCNet-UR model, the runtime takes on average 220 milliseconds for the resized images of 256×256 from TSS benchmark [27]. For comparison, PSCNet-SE takes 420 milliseconds when implemented to pass the cost volume sequentially to the sub-networks of PSCNet-SE, i.e., affine transformation regression networks and element detection networks. To expedite the runtime of PSCNet-SE, we split those sub-networks onto two different GPUs and then feed-forward cost volume in a parallel manner. This allows us to significantly reduce the execution time of PSCNet-SE to 248 ms, closing the gap to PSCNet-UR. As shown in Table 13, PSCNet-UR and PSCNet-SE are slower than the methods that estimate global transformation parameter, but yield a significantly better matching performance.

7 Conclusion

We presented a deep architecture, called pyramidal semantic correspondence networks (PSCNet), that estimates locally-varying affine transformation fields across semantically similar images. While existing methods suffer from the trade-off between the precise localization ability and the robustness to the semantic variations, we acheive both thanks to the proposed pyramidal model. Experimental results on various benchmarks demonstrate the effectiveness of our two models that divide an image in a form of quad-tree rectangles or into multiple semantic elements of an object.

References

[1] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," in *Proc. Special Interest Group Comput. Graph. Interactive Techn. Conf.*, 2011, Art. no. 70.

[2] C. Liu, J. Yuen, and A. Torralba, "SIFT Flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 815–830, May 2011.

[3] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2307–2314.

[4] H. Yang, W. Y. Lin, and J. Lu, "DAISY filter flow: A generalized discrete approach to dense correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3406–3413.

[5] T. Zhou, Y. J. Lee, S. X. Yu, and A. A. Efros, "FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1191–1200.

[6] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1, pp. 7–42, 2002.

[7] J. Zbontar, Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2287–2318, 2016.

[8] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6602–6611.

[9] D. Butler, J. Wulff, G. Stanley, and M. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 611–625.

[10] P. Fischer *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2758–2766.

[11] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2462–2470.

[12] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3475–3484.

[13] K. Han *et al.*, "SCNet: Learning semantic correspondence," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1849–1858.

[14] S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn, "FCSS: Fully convolutional self-similarity for dense semantic correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 616–625.

[15] J. L. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 1601–1609.

[16] N. Ufer and B. Ommer, "Deep semantic feature matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6914–6923.

[17] S. Kim, D. Min, S. Lin, and K. Sohn, "DCTM: Discrete-continuous transformation matching for semantic flow," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4539–4548.

[18] I. Rocco, R. Arandjelović, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 39–48.

[19] P. Hongsuck Seo, J. Lee, D. Jung, B. Han, and M. Cho, "Attentive semantic alignment with offset-aware correlation kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 349–364.

[20] I. Rocco, R. Arandjelović, and J. Sivic, "End-to-end weakly-supervised semantic alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6917–6925.

[21] S. Kim, S. Lin, S. Jeon, D. Min, and K. Sohn, "Recurrent transformer networks for semantic correspondence," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 6126–6136.

[22] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1651–1662.

[23] J. Lee, D. Kim, J. Ponce, and B. Ham, "SFNet: Learning object-aware semantic correspondence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2278–2287.

[24] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[25] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[26] S. Jeon, S. Kim, D. Min, and K. Sohn, "PARN: Pyramidal affine regression networks for dense semantic correspondence," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 351–366.

[27] T. Taniai, S. N. Sinha, and Y. Sato, "Joint recovery of dense correspondence and cosegmentation in two images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4246–4255.

[28] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow: Semantic correspondences from object proposals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 7, pp. 1711–1725, Jul. 2018.

[29] F.-F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[30] J. Min, J. Lee, J. Ponce, and M. Cho, "Hyperpixel flow: Semantic correspondence with multi-layer neural features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3394–3403.

[31] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[32] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.

[33] F. Yang, X. Li, H. Cheng, J. Li, and L. Chen, "Object-aware dense semantic correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4151–4159.

[34] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[35] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2414–2422.

[36] D. Novotny, D. Larlus, and A. Vedaldi, "AnchorNet: A weakly supervised network to learn geometry-sensitive features for semantic matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2867–2876.

[37] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Self-supervised learning of geometrically stable features through probabilistic introspection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3637–3645.

[38] C.-H. Lin and S. Lucey, "Inverse compositional spatial transformer networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2252–2260.

[39] S. Huang, Q. Wang, S. Zhang, S. Yan, and X. He, "Dynamic context correspondence network for semantic alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2010–2019.

[40] S. Li, K. Han, T. W. Costain, H. Howard-Jenkins, and V. Prisacariu, "Correspondence networks with adaptive neighbourhood consensus," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10 196–10 205.

[41] J. Min, J. Lee, J. Ponce, and M. Cho, "SPair-71k: A large-scale benchmark for semantic correspondence," 2019, *arXiv:1908.10543*.

[42] J. Min, J. Lee, J. Ponce, and M. Cho, "Learning to compose hypercolumns for visual correspondence," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 346–363.

[43] Y. Liu, L. Zhu, M. Yamada, and Y. Yang, "Semantic correspondence as an optimal transport problem," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4463–4472.

[44] J. Thewlis, H. Bilen, and A. Vedaldi, "Unsupervised learning of object landmarks by factorized spatial embeddings," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5916–5925.

[45] W.-C. Hung, V. Jampani, S. Liu, P. Molchanov, M.-H. Yang, and J. Kautz, "SCOPS: Self-supervised co-part segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 869–878.

[46] S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu, "Unsupervised part segmentation through disentangling appearance and shape," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8351–8360.

[47] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 41–48.

[48] J. Hur, H. Lim, C. Park, and S. C. Ahn, "Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1392–1400.

[49] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "DeepMatcing: Hierarchical deformable dense matching," *Int. J. Comput. Vis.*, vol. 120, pp. 300–323, 2016.

[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[52] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2307–2314.

[53] Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee, "Unsupervised discovery of object landmarks as structural representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2694–2703.

[54] A. Kendall *et al.*, "End-to-end learning of geometry and context for deep stereo regression," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 66–75.

[55] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1520–1528.

[56] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 533–540, 2006.

[57] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisseman, "The PASCAL Visual Object Classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[58] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasum, and A. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1979–1986.

[59] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[61] P. H. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[63] J. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 1601–1609.

[64] J. Lee, D. Kim, W. Lee, J. Ponce, and B. Ham, "Learning semantic correspondence exploiting an object-level prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 3, 2020, doi: 10.1109/TPAMI.2020.3013620.

[65] S. Jeon, D. Min, S. Kim, J. Choe, and K. Sohn, "Guided semantic flow," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 631–648.

[66] E. Collins, R. Achanta, and S. Susstrunk, "Deep feature factorization for concept discovery," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 336–352.

[67] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3D object detection in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2014, pp. 75–82.

Sangryul Jeon (Student Member, IEEE) received the BS degree in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2016. He is currently working toward the joint MS and PhD degrees in electrical and electronic engineering at Yonsei University, Seoul, Korea. His current research interests include 2D/3D computer vision and machine learning, in particular image alignment and representation learning.

Seungryong Kim (Member, IEEE) received the BS and PhD degrees from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea, in 2012 and 2018, respectively. From 2018 to 2019, he was postdoctoral researcher in Yonsei University, Seoul, Korea. From 2019 to 2020, he has been post-doctoral researcher in the School of Computer and Communication Sciences, École Polytechnique Féd érale de Lausanne (EPFL), Lausanne, Switzerland. Since 2020, he has been an assistant professor with the Department of Computer Science and Engineering, Korea University, Seoul, Korea. His current research interests include 2D/3D computer vision, computational photography, and machine learning.

Dongbo Min (Senior Member, IEEE) received the BS, MS, and PhD degrees from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, in 2003, 2005, and 2009, respectively. From 2009 to 2010, he was a postdoctoral researcher with Mitsubishi Electric Research Laboratories, Cambridge, Massachusetts. From 2010 to 2015, he was with the Advanced Digital Sciences Center, Singapore. From 2015 to 2018, he was an assistant professor in the Department of Computer Science and Engineering, Chungnam National University, Daejeon, South Korea. Since 2018, he has been in the Department of Computer Science and Engineering, Ewha Womans University, Seoul, South Korea. His current research interests include computer vision, deep learning, and video processing,

Kwanghoon Sohn (Senior Member, IEEE) received the BE degree in electronic engineering from Yonsei University, Seoul, Korea, in 1983, the MSEE degree in electrical engineering from the University of Minnesota, Minneapolis, Minnesota, in 1985, and the PhD degree in electrical and computer engineering from North Carolina State University, Raleigh, North Carolina, in 1992. He was a senior member of the research engineer with the Satellite Communication Division, Electronics and Telecommunications Research Institute, Daejeon, Korea, from 1992 to 1993, and a postdoctoral fellow with the MRI Center, Medical School of Georgetown University, Washington, DC, in 1994. He was a visiting professor with Nanyang Technological University, Singapore, from 2002 to 2003. He is currently an Underwood distinguished professor with the School of Electrical and Electronic Engineering, Yonsei University, Korea. His research interests include 3D image processing and computer vision.