Object-aware Dense Semantic Correspondence

Fan Yang[1], Xin Li[1]*, Hong Cheng[2], Jianping Li[1], Leiting Chen[1]
[1]School of Computer Science & Engineering, UESTC
[2]Center for Robotics, School of Automation Engineering, UESTC
fanyang_uestc@hotmail.com, xinli_uestc@hotmail.com, hcheng@uestc.edu.cn

Abstract

This work aims to build pixel-to-pixel correspondences between images from the same visual class but with different geometries and visual similarities. This task is particularly challenging because (i) their visual content is similar only on the high-level structure, and (ii) background clutters keep bringing in noises.

To address these problems, this paper proposes an object-aware method to estimate per-pixel correspondences from semantic to low-level by learning a classifier for each selected discriminative grid cell and guiding the localization of every pixel under the semantic constraint. Specifically, an Object-aware Hierarchical Graph (OHG) model is constructed to regulate matching consistency from one coarse grid cell containing whole object(s), to fine grid cells covering smaller semantic elements, and finally to every pixel. A guidance layer is introduced as the semantic constraint on local structure matching. In addition, we propose to learn the important high-level structure for each grid cell in an "objectness-driven" way as an alternative to handcrafted descriptors in defining a better visual similarity.

The proposed method has been extensively evaluated on various challenging benchmarks and real-world images. The results show that our method significantly outperforms the state-of-the-arts in terms of semantic flow accuracy.

1. Introduction

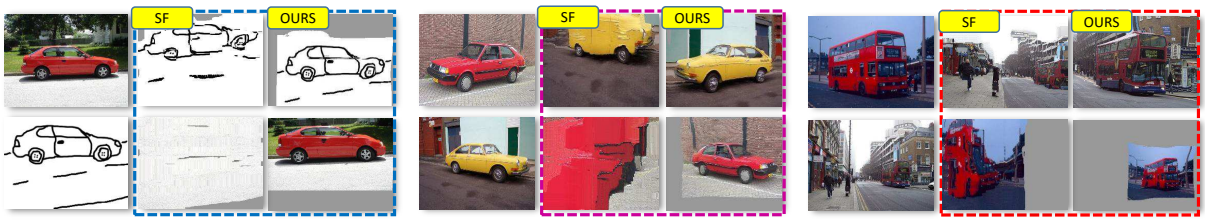
Dense semantic correspondence, which is defined as the correlation between pixels in one image and those in another semantically similar image, is an important problem in computer vision. Many research efforts have been devoted to building dense semantic correspondences due to its wide application in semantic segmentation [1], depth estimation [13], scene parsing [16], co-segmentation [22], salient object detection [31], pose estimation [7], *etc.*

Unlike optical-flow methods [4] [5] designed to analyze the transformation between images from the same scene (*e.g.,* adjacent video sequences), semantic flow methods aim at establishing per-pixel correspondences between visually-related images that may have no spatial or temporal relations. To this end, prior works [18] [12] [26] [14] typically use descriptor similarity metric (L_1 metric) to perform local detection, and then adopt the computational framework of optical flow to achieve global optimization. These methods are based on assumptions that (i) semantically consistent pixels (or regions) share a sufficiently similar low-level structure, and (ii) all pixels (or regions) are equally significant in each image. They can mostly produce accurate results when local handcrafted features (*e.g.,* SIFT descriptors) are reasonably discriminative, and background clutters bring in only a small amount of noise. But what about the cases when the exact appearance of semantically consistent regions is similar only on the high-level structure, or when the noise from background clutters cannot be filtered out by the optimizer? For instance, correspondence methods often need to match images with high intra-class variations (Fig. 1(a)), changes in viewpoint (Fig. 1(b)) or strong background clutters (Fig. 1(c)). In these cases, the L_1 metric is inadequate to estimate the likelihood of semantically consistent regions. Moreover, treating all regions equally also may hurt performance.

In this paper, we propose a novel approach to overcome these drawbacks. The key idea is to build semantic correspondences based on the learned classifier for each discriminative grid cell (object region) and then refine the flow fields by matching local structures. By integrating these two processes into a single model and optimizing it in a coarse-to-fine manner, we can narrow the "semantic gap" and eliminate background clutters. The contributions of this paper are as follows:

1) Object-aware Hierarchical Graph model. We design an object-aware hierarchical graph (OHG) model for dense correspondence with following novelties: (i) a novel algorithm is proposed to construct an object-aware hierarchical architecture for the input image; (ii) a guidance layer is introduced to drive the matching of local structures; (iii) the

*Corresponding author.



(a) Intra-class variations (b) Changes in viewpoint (c) Background clutters

Figure 1. Dense semantic correspondences between (a) images with high intra-class variations; (b) images with changes in viewpoint; and (c) images with strong background clutters. We warp each image to its corresponding target using the estimated dense correspondences to illustrate the result. We also show the results of the most widely used SIFT-Flow (SF) [18] for a comparison.

scale and flip invariant properties are imposed in our model ; and (iv) matching consistency is regulated through hierarchical optimization.

2) "Objectness-driven" visual similarity. Unlike previous methods relying on L_1 metric to measure visual similarity, we propose to train a discriminative classifier for each node in the upper two layers of our model in an "objectness-driven" way — using a large set of "background" images/patches. The learned weight features can capture important high-level visual structures of the object while safely ignoring the local dissimilarity.

We evaluate our method on a variety of well-used benchmarks as well as many real-world images. The results show that the proposed method can generate much more accurate dense semantic correspondences than the state-of-the-arts.

The rest of this paper is organized as follows. We first give a brief overview of related works in Section. 2, and then describe our method in detail in Section. 3. Section. 4 provides both qualitative and quantitative results on several public datasets and real-world images. Conclusions are drawn in Section. 5.

2. Related Work

Building per-pixel correspondence is a fundamental task in computer vision. Dense correspondence approaches are originally designed for estimating optical flow fields [4] or depth [21] between two very similar images. Recently, there has been a growing interest in designing methods for dense semantic correspondence estimation. Matching pixels between different scenes goes beyond the same-scene assumption, which makes it a much more challenging task.

As an important step in estimating dense semantic correspondence, SIFT-Flow (SF) [18] [17], for the first time, is proposed to estimate dense correspondences across scene/object appearances. One typical assumption in SIFT-Flow is that visually similar pixels should share same (or sufficiently similar) local structures. Therefore, SIFT-Flow produces semantically meaningful correspondences by matching SIFT descriptors instead of matching raw pixel intensities with the computational framework of optical flow. However, the underlying Dense SIFT (DSIFT) is not robust to geometric variations and background clutters, largely limiting the applicability of SIFT-Flow.

Over the years, a number of methods with better performance in handling different visual variations have been proposed. Many of them focus on designing a more powerful dense descriptor, which represents a quite straightforward solution. Hassner et al. [12] developed an alternative SIFT representation, named Scale-Less SIFT (SLS) descriptor, for the purpose of scale-invariant dense matching, and achieved impressive results in dense semantic correspondences across scales. However, it is difficult to apply SLS in practice due to its high computational complexity. Match-aware SIFT [26] was proposed to address these shortcomings by propagating reliable scales detected from key-points to every pixel, but it relies heavily on key-point matching technology which is unreliable for cross-scene matching. With some modifications to the Scale Invariant Descriptor (SID), Segmentation-aware SID (S-SID) [27] exploited "Soft-Segmentation" masks to counter the background-clutter effect, but it may reduce the discriminative power of original descriptors. Overall, these approaches focus on designing problem-specific dense descriptors. Despite their great performance in matching similar scenes/objects, these methods still suffer many drawbacks caused by the well-known "semantic gap" [19] between low-level descriptors and high-level semantics.

On the other hand, more powerful optimizers have been proposed. Deformable Spatial Pyramid (DSP) [14] was introduced to perform cross-scene/object matching. This method regulates matching consistency through a pyramid graph, where larger spatial nodes mainly handle appearance variations while smaller ones help to localize matches with fine detail. DSP uses SIFT descriptors as underlying representations, so it also suffers from the "semantic gap". Besides, it has two major weaknesses in its structure. First, its regular spatial division of image may create many "bad" patches, *i.e.* patches that do not correspond to visual phrases. Second, background clutters may drastically reduce matching accuracy. Taniai *et al.* [25] proposed to recov-

er co-segmentation and dense correspondence altogether in image pair. Ham *et al.* [11] introduced an efficient method to estimate pixel-wise correspondences by matching object proposals. However, the inherent limitation of handcrafted descriptor still remain unsolved. To solve above drawbacks, Zhou *et al.* [35] used cycle-consistency as a supervisory signal to learn the high-level semantic information for dense correspondences. Although this method benefits from powerful deep CNNs, it is extremely limited in the application for requiring massive training sets of different categories.

Bristow et al. [3] recently proposed a closely related method which learns exemplar LDA classifier with 5×5 spatial support for each pixel. Although we share some similar goals, our work is quite different. First, considering that many small patches (*e.g.,* patches of skin) are plain and thus barely distinguishable, we devise a hierarchical strategy based on our OHG model, in which classifiers are only learned for nodes in upper layers which may correspond to whole object(s) or smaller semantic parts of object(s), and matches for these nodes are then used to guide the localization of each pixel. Second, our method focuses more on object regions so that background clutters can be largely reduced.

3. Approach

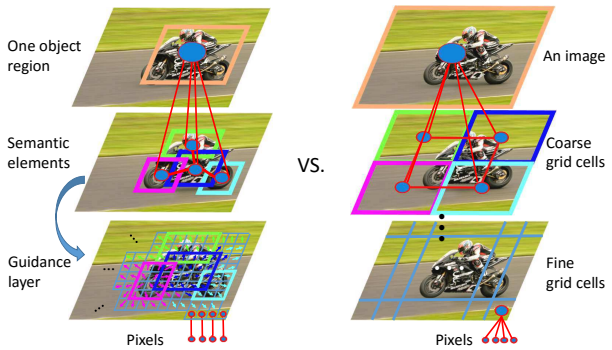
In this section, we firstly give a detailed description of the proposed Object-aware Hierarchical Graph (OHG) model and illustrate why it is superior to previous Deformable Spatial Pyramid (DSP) model. Then, we show how to learn the important high-level structure for each node in the upper two layers by using a large dataset of "background" images/patches, as a way in defining a more reliable visual similarity.

3.1. Object-aware Hierarchical Graph Model

Unlike traditional spatial pyramid, our object-aware hierarchical architecture starts from one discriminative rectangular region (object region), to several smaller object proposals (semantic elements), to the guidance layer, and finally to every pixel (see Fig. 2). This novel four-layer architecture has two distinct advantages: (i) it largely reduces the noises caused by background clutters from the optimization framework; (ii) it makes training a discriminative classifier for nodes in upper two layers in the graph become possible, because the negative set can be easily guaranteed and obtained — a large set of pure "background" images/patches.

Our model is based on object proposal algorithms, which generate a small number of windows (*e.g.,* 1000) likely to cover all objects in an image. In this paper, our model adopts the Selective Search (SS) proposals [28], but it is not limited to any particular type of proposal algorithm.

We firstly use the computed object proposals to localize one distinctive rectangular region as the coarsest layer. To



(a) Ours (b) DSP

Figure 2. Comparison of graph representations between (a) our OHG and (b) DSP [14]. The blue circle denotes a graph node, and the edges link all neighboring nodes. Our object-aware hierarchical architecture focuses more on object regions while spatial pyramid treats each region equally.

be more specific, given N_P (*e.g.,* $N_P = 500$) proposals for the novel input, we obtain the objectness map Obj by accumulating all object proposals P_r:

$$Obj(p) = \sum_{i=1}^{N_p} Pr_i(p) \tag{1}$$

where $Pr_i(p)$ is the score at position p determined by the i-th proposal, which takes 1 if the pixel p is covered by this proposal, and 0 otherwise. The objectness map Obj tells us how likely each pixel belongs to the object region. Therefore the pixels with low objectness scores usually belong to the background. Inspired by some salient object detection methods [29] [31], we can compute the accumulated objectness value in four directions on the normalized objectness map to exclude those regions. According to the accumulated value, the four sides of the rectangular region can be easily determined by a pre-defined threshold θ ($\theta = 0.1$). The integral image [30] computed from objectness map can be adopted to boost computational efficiency of this step.

We want the second layer to be composed of smaller grid cells containing semantic elements. To this end, we initially determine a number of proposals as candidates C by selecting those whose size is $0.2 \sim 0.4$ time that of the first layer and intersect with the first layer that is higher than 0.6. It is a highly complex task to detect a small set of semantic elements from C. Inspired by [34], we convert this question of semantic element detection to a clustering problem. Given the candidates C, we divide them into clusters and select only one proposal from each group as the output detection. In addition, we favor a small set of proposals that have small overlaps with other selected ones and tend to cover the entire space of the first layer. Hence, we define the following

objective function:

$$\max_{B}\{S(B)-\alpha\cdot O(B)+\beta\cdot U(B)-\gamma\cdot N(B)\} \quad s.t. \quad B\subseteq C \tag{2}$$

Here $S(\cdot)$ is the data term that encourages the selection of proposals that are more likely to belong to one cluster center; $O(\cdot)$ denotes overlap term that penalizes intersection between selected windows. $U(\cdot)$ is coverage term which represents the coverage of the selected windows in the coarsest layer. $N(\cdot)$ denotes the number term, and it penalizes the number of selected regions. By maximizing Eq. 2, we can determine a small subset of proposals so that these selected ones are very likely to be a cluster center, have small overlaps with other selected windows, and cover the entire coarsest grid cell (the first layer).

To be more specific, we use a binary variable z_i to indicate the selection of proposal b_i from candidates C. If b_i is selected, $z_i = 1$ otherwise 0. So, the data term is $\sum_{b_i\subseteq C} S_i z_i$, where S_i is the score reflecting the likelihood of the proposal b_i to be a cluster center. Here we adopt the measure proposed in [34] to compute S_i. For each proposal b_i, $S_i = \sum_{b_j\in C}\max(\log(s_j\cdot K(b_i,b_j))-\tau,0)$, where s_j is the original score of proposal b_j; $K(b_i,b_j)$ is a function that measures the overlaps between b_i and b_j, and here we use the popular Intersection-over-Union (IoU) score [8]; $\tau=\log(\frac{1}{\sum_{j=1}^{n} s_i\cdot K(b_i,b_j)})$ is a normalization constant. Apart from selecting proposals with high possibility of belonging to a cluster center, the proposals selected from C should have small overlaps with other selected ones. The overlap cost is $\sum_{b_i,b_j\in C;i\neq j} K(b_i,b_j)z_i z_j$. Besides, we encourage the selected proposal to cover the entire coarsest grid cell. Therefore, we partition the coarsest grid cell into small tiles. We introduce a binary variable t_m to indicate whether tile m is covered by the selected proposals. If tile m is covered by any selected proposal, $t_m = 1$. Otherwise, $t_m = 0$. Therefore, the coverage term is $\sum_{m\in T} t_m$, where T denotes the set of tiles in the coarsest layer. We tend to select a small set of proposals, because it is time-consuming to train classifiers for a large number of regions. Therefore, we include the number term $\sum_{b_i\in C} z_i$ in our objective function. Combining all the terms, we get the following objective function:

$$\max\{\sum_{b_i\subseteq C}(S_i-\gamma)z_i-\alpha\cdot\sum_{\substack{b_i,b_j\in C\\ i\neq j}}K(b_i,b_j)z_i z_j+\beta\cdot\sum_{m\in T}t_m\} \quad s.t. \quad z_i,z_j=0 \quad or \quad 1 \tag{3}$$

Seeking the solution of Eq. 3 is typically a NP-hard problem. To quickly solve this problem, we adopt the greedy algorithm described in [34]. It starts from an empty solution set, and adds proposals to the solution set until no more proposals can be added to improve the objective function. Then, it removes proposals from the solution set until no more proposals can be removed to further improve the object function. The interactions described above keep iterating until a local optimal solution is found. Finally, only a few proposals are selected to form the semantic element layer of our hierarchical architecture.

The upper two layers of our hierarchical architecture only focus on the important object regions. We believe that the matches of these regions provide useful priors that can help better estimating the correspondences for all pixels. Hence, we have one additional layer underneath the second layer, which is referred to as dense guidance layer, to bridge the gap between region-wise semantic and pixel-wise low-level correspondences. Based on the matches of nodes in the upper-two layers, the guidance layer is automatically generated by using a graphical model which has been used in colorization [15] and scale estimation [26]. Specifically, we initialize the translation vector $w^g(p) = (u_p, v_p)$ at pixel p in object regions with its corresponding region-wise translation vector. Then the known translation vectors $w^g(p)$ are propagated to all unknown pixels by minimizing the following objective function, as follows,

$$J(w^g)=\sum_p(w^g(p)-\sum_{p,q\in N}w_{pq}w^g(q))^2 \tag{4}$$

where w_{pq} is a weighting function. In contrast to [15] [26], we define a linear relationship between intensities and flows, rather than between intensities and colors or scales. We assume that neighboring pixels with similar intensities have similar flows (translation vectors). Thus, the weighting function is written as follows,

$$w_{pq}\propto 1+\frac{1}{\sigma_p^2}(G(p)-\mu_p)(G(q)-\mu_q) \tag{5}$$

where $G(p)$ and $G(q)$ denotes the intensities of pixels p and q respectively. σ_p and μ_p are the mean and variance of the intensities in the neighboring region of pixel p. w_{pq} will be large if $G(p)$ is similar to $G(q)$ and vice versa. Finally, the dense guidance layer is automatically generated by solving Eq. 4 using normalized cuts [23].

The bottom layer in our hierarchical architecture is the pixel layer. We use a graph to represent the proposed object-aware hierarchical architecture, where each grid cell is a node. All neighboring nodes (grid cells with overlaps) and parent-child nodes are connected by edges (see Fig. 2). In the bottom layer, each pixel is linked only to its parent node.

3.2. Matching Objective

We design two objective functions for our Object-aware Hierarchical Graph (OHG) model. For the nodes in the up-

Figure 3. Comparison of initial features and learned features. In each case, the learned features boost the gradients belonging to important parts of a given region while the initial features only "upweight" the regions with large gradient changes.

per two layers, we impose greater regularization and define a better visual similarity to improve their robustness to appearance variations. For the nodes in the pixel-wise layer (bottom layer), we design a reduced objective function and simply use a L_1 metric for measuring visual similarity, in order to boost computational efficiency.

First, we introduce the objective function for the upper layers. Let $\mathcal{I}_S$ and $\mathcal{I}_T$ denote the source image and target image, respectively. In order to make our model robust to horizontal flipping, a common visual variation in real-world images, we compute the horizontal flipping translation $\mathcal{F}$ based on the target image $\mathcal{I}_T$ in advance. We use $w_i^{'}$ to denote the temporary translation of node i, and the final translation of a node in the upper layer is denoted as w_i, where $w_i = w_i^{'} + f_i \cdot \mathcal{F}_i$, and $f_i = 1$ or 0 is the flip variable. Our matching objective function is given as:

$$E(w^{'}, s, f) = \sum_i D_i(w_i^{'}, s_i, f_i) + \lambda \sum_{i,j \in N} V_{i,j}(w_i^{'}, w_j^{'}) + \mu \sum_{i,j \in N} S_{i,j}(s_i, s_j) + \nu \sum_{i,j \in N} F_{i,j}(f_i, f_j) \tag{6}$$

where D_i is a data term; $V_{i,j}(w_i^{'}, w_j^{'}) = min(||w_i^{'} - w_j^{'}||_1, \varepsilon)$ is a spatial smoothness term. Unlike the original DSP objective function, we add a scale smoothness term $S_{i,j} = ||s_i - s_j||_1$ and a flip smoothness term $F_{i,j} = ||f_i - f_j||_1$. N denotes pairs of nodes linked by graph edges.

Different from previous works that simply use a L_1 metric to evaluate the matching likelihood, our data term is defined as,

$$D_i(w_i^{'}, s_i, f_i) = \phi(W_i^T X(i^{'})) \tag{7}$$

$$i^{'} = s_i(i + F \cdot f_i + w_i^{'}) \tag{8}$$

where W_i is a SVM classifier trained to measure the visual similarity between node i in the source image $\mathcal{I}_S$ and patch $i^{'}$ with the state $(w_i^{'}, s_i, f_i)$ in target image $\mathcal{I}_T$. $X(i^{'})$ is the feature vector extracted from patch $i^{'}$ in the target image, and $\phi(\cdot)$ maps the detection scores into the range $[0, 3]$.

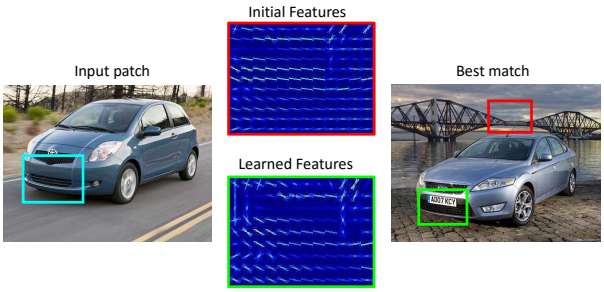


Figure 4. Comparison of patch matching using initial feature and learned feature. The learned feature "upweights" the gradients of important regions, therefore it is much more robust to appearance variations.

After our method finishes matching all nodes in the upper two layers, we use Eq. 4 to generate the dense guidance flow w^g to drive the matching of each node in pixel-wise layer. To accelerate the matching process, we neither link neighboring nodes in pixel-wise layer nor train a classifier for each pixel. We only use the HOG feature [10] for each pixel p to describe its local structure. Therefore, the objective function is given as:

$$E^{'}(w^f) = \sum_p D_p^{'}(w_p^f) + \sum_{p,q \in N^{PC}} V_{p,q}(w_p^f, w_q^g) \tag{9}$$

where the first term $D^{'}$ adopts L_1 metric to measure the HOG distance and the second term ensures that the final flow vector w^f is driven by w^g. N^{PC} denotes parent-child nodes.

3.2.1 Optimization

We initialize the solution by using the optimization framework of DSP [14] for the hierarchical graph built based on all nodes in the upper two layers. Because our model keeps only a few nodes (usually less than 10 discriminative patches) in the upper two layers, the node matching here is very fast. In the pixel-wise layer, each pixel is a node. For better efficiency, we propose a two-stage optimization strategy. Eq. 9 has two terms: data term and guidance term. In Stage 1, we consider only the guidance term. Since the guidance term favors the consistency between final flow and guidance flow, we directly generate the initial location of each pixel by using the guidance flow. In Stage 2, we refine the initial flow within a 10-pixel radius search region by using a reduced object function of Eq. 9 as $E^{'}(w^f) = \sum_i D_p^{'}(w_p^f)$.

3.3. "Objectness-driven" Visual Similarity

The L_1 norm for descriptor distance is adopted by most previous dense semantic correspondence methods for mea-

suring visual similarity. However, handcrafted descriptors tend to capture the minor details rather than important high-level structure of the given region (see Fig. 3). Therefore, directly adopting handcrafted descriptors as underlying representations and using L_1 metric alone for measuring visual similarity usually cause errors in challenging cases, *e.g.,* matching across high appearance variations (see Fig. 4).

To overcome this drawback, we train a discriminative classifier to capture important part of each feature representation, and then use the learned detectors to meature the visual similarity. Specifically, we model the appearance of each node with a HOG template [10], and employ the linear Support Vector Machine (SVM) [6] to discover which parts of the representation are most visually important and which parts can be safely ignored. The visual similarity, therefore, can be defined as follows:

$$S(i, i^{'}) = W_i^T X(i^{'}) \tag{10}$$

where W_i is the learned weight vector for node i in the source image $\mathcal{I}_S$, and $X(i^{'})$ denotes the feature vector of its corresponding target $i^{'}$ in the another image $\mathcal{I}_T$.

Since what we want to capture is the most important structure of a given patch, we can train an exemplar-specific classifier for each node in the upper two layers in a "objectness-driven" way. We hypothesize that the most important features of an object region are also the features exhibiting high "objectness", which best discriminate this region against the "background" samples. Therefore, we train a SVM classifier with a single positive example and millions of "background" images/patches, similar to [20] [24]. To improve its robustness to small transformations, we expand the positive set by performing *slight transformations* (*e.g.,* a shift of less than 5 pixels in different directions). We denote the positive set as P_o and the negative set as N_E, and the feature vector for each patch as X. The weight vector W_i for node i is computed as follows:

$$\begin{aligned}\Omega(W_i) =& C_1 \sum_{x_p \in P_o} h(W_i^T X_p) + C_2 \sum_{x_n \in N_E} h(-W_i^T X_n) \\ &+ ||W_i^T||^2\end{aligned} \tag{11}$$

where $h(x) = max(0, 1 - x)$ is the hinge loss function. C_1 and C_2 are regularization parameters. Since the solution only depends on a small set of "hard" negative support vectors [10], we use the hard-negative mining approach [20] to cope with millions of negative windows. As we learn each classifier independently, a careful calibration phase is required so that the outputs are comparable. We follow the calibration process in [2]. Because there are only a few classifiers in our model, we find that running all classifiers on a set of 2000 "background" patches achieves good results.

Table 1. **Quantitative results on JR dataset.** FAcc denotes flow accuracy rate for an error threshold of 5 pixels. SAcc is segmentation accuracy by using Intersection-over-Union (IoU) scores. "*" represents the final version of the proposed method used in this paper. Our method consistently shows best scores.

Methods(%)	FG3DCar		JODS		PASCAL	
	FAcc	SAcc	FAcc	SAcc	FAcc	SAcc
SF [18]	63.37	75.50	52.22	56.91	45.27	72.72
DSP [14]	48.69	72.69	46.53	61.50	38.22	69.94
DFF [32]	49.46	59.33	30.41	48.31	22.45	53.58
UFL [33]	36.90	65.68	34.71	51.18	23.78	62.21
PF [11]	79.13	75.18	64.40	59.67	48.52	67.25
JR [25]	82.97	73.04	59.48	54.11	48.31	67.68
Ours w/o SVM	69.51	74.76	55.10	57.22	61.62	71.73
Ours*	**87.46**	**85.58**	**70.78**	**68.44**	**72.92**	**78.21**

4. Experiments

Implementation The proposed method is implemented in MATLAB. We represent each node in the upper two layers with a rigid HOG template [10], and the HOG features are computed over an image pyramid. The LIBSVM [6] is used to train each node's weight vector W. We create negative samples by using images from the PASCAL VOC 2007 dataset [8] and filtering out patches containing objects. For the pixel layer, we use HOG [10] features with 25×25 spatial support as underlying represents. We set $\alpha = 0.5$, $\beta = 0.3$ and $\gamma = 0.5$ in Eq. 2; We set $\lambda = 0.3$, $\mu = 0.2$ and $\upsilon = 0.4$ in Eq. 6. We use the regularization parameter $C_1 = 0.1$ and $C_2 = 0.01$ in Eq. 11. The values of these parameters are fixed in the following experiments. We will release the code and results online.

Methods of comparison We compare the proposed method with currently strongest methods including SF [18], DSP [14], DFF [32], UFL [33], MATCH [26], PF [11] and JR [25]. In total, we make comparisons with 7 leading methods. In all cases, we use the code and recommended parameter settings published by the authors of each method.

4.1. Results on JR dataset

The JR dataset [25] is designed for evaluating the accuracy of dense semantic correspondence. It includes three subsets with different difficulty levels. FG3DCar is a the simplest subset. It contains 195 image pairs from the category of vehicle. The challenge is to handle appearance variations. JODS includes 81 image pairs of airplanes, horses, and cars. The images in JODS have large intra-class variations, as well as changes in scale and viewpoint. PASCAL is the most challenging subset. It contains 124 image pairs from different object categories (e.g., bicycle, motorbike and train). These images contain objects with high appearance variations, large changes in viewpoint, strong background clutters, as well as flipping variations.

We evaluate the pixel-wise flow accuracy on JR dataset. Following the experimental protocol in [25], we compute

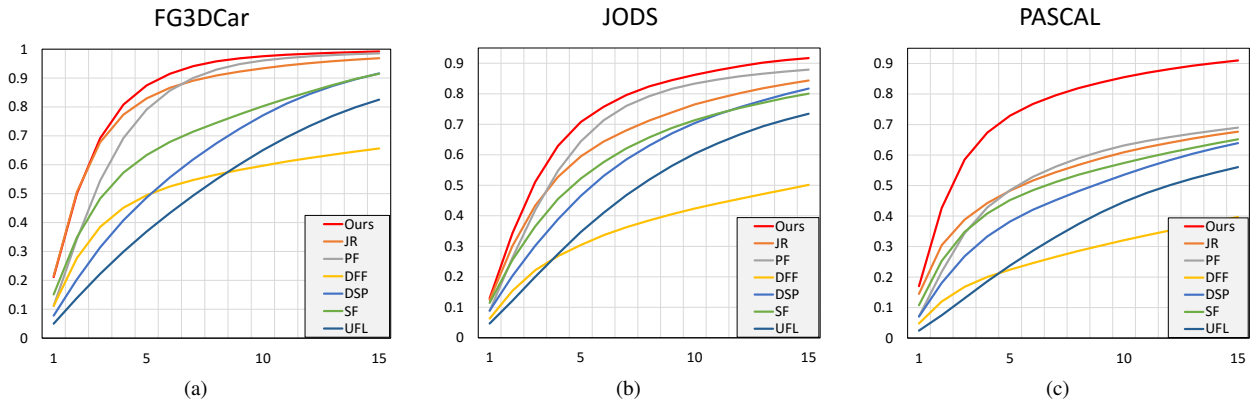


Figure 5. **Average flow accuracies with varying thresholds on JR dataset.** Our method consistently outperforms the state-of-the-arts. On the most challenging PASCAL, our method can still perform well, while other methods have been shown to fail.

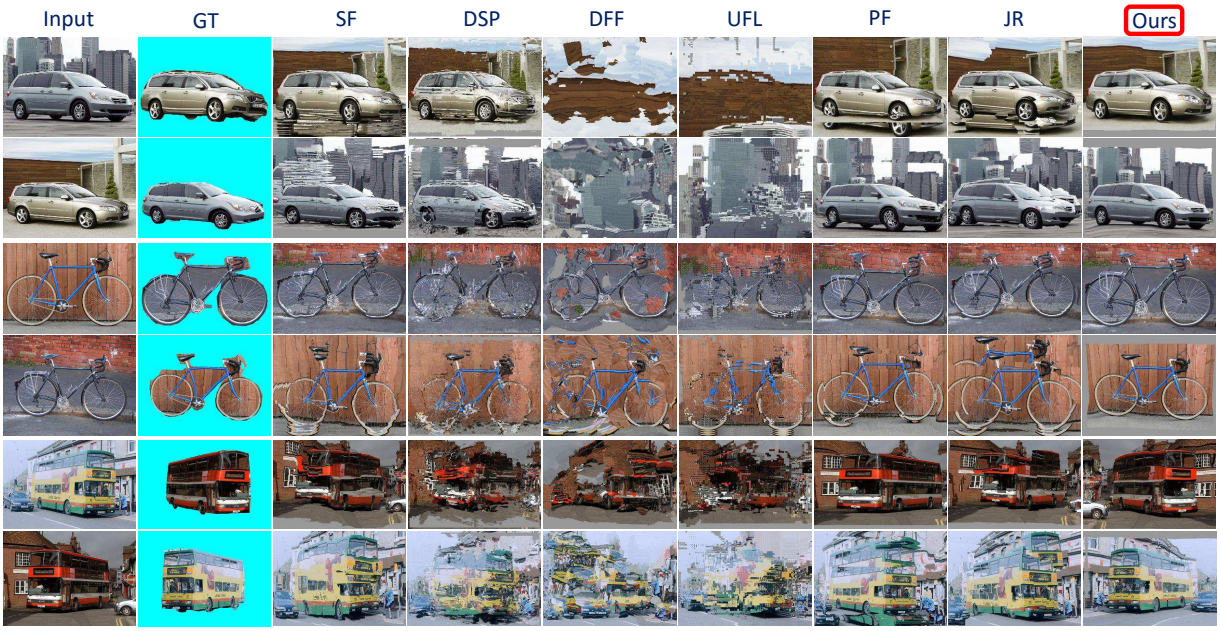


Figure 6. **Qualitative results on JR dataset.** We show some example results from different subsets of JR Benchmark.

the average flow accuracy with an error threshold of 5 pixels on each subset. As shown in Tab. 1, our method reports the highest dense correspondence accuracy on all subsets. Specifically, on FG3DCar and JODS, we achieve an improvement of 5.41% and 9.91% respectively in terms of pixel-wise flow accuracy over the currently strongest methods. On the most challenging PASCAL, we outperform the best existing methods by 50.29% in term of flow accuracy. That is because 1) our method is able to handle flip variations, eliminate background clutters, and 2) the trained features are much more robust to appearance variations. Similar to [25], we also plot the percentage of correct correspondences using varying thresholds (see Fig. 5).

Additionally, according to the computed flow fields, we transfer the groundtruth mask of one image to the other in each pair to measure the IoU score [10] for a more balanced comparison. The annotation transfer results are also listed in Tab. 1. Our method also achieves the highest score with a significant improvement of 13.35% on FG3DCar, 11.28% on JODS and 8.44% on PASCAL over previously best results. Some sample results are shown in Fig. 6.

4.2. Results on Caltech-101 dataset

Caltech-101 dataset [9] includes 101 object categories. In each category, there are more than 50 images containing object(s) in different locations and scales, and exhibiting high appearance variations. It also provides ground-truth pixel labels for the foreground object. Although it has no

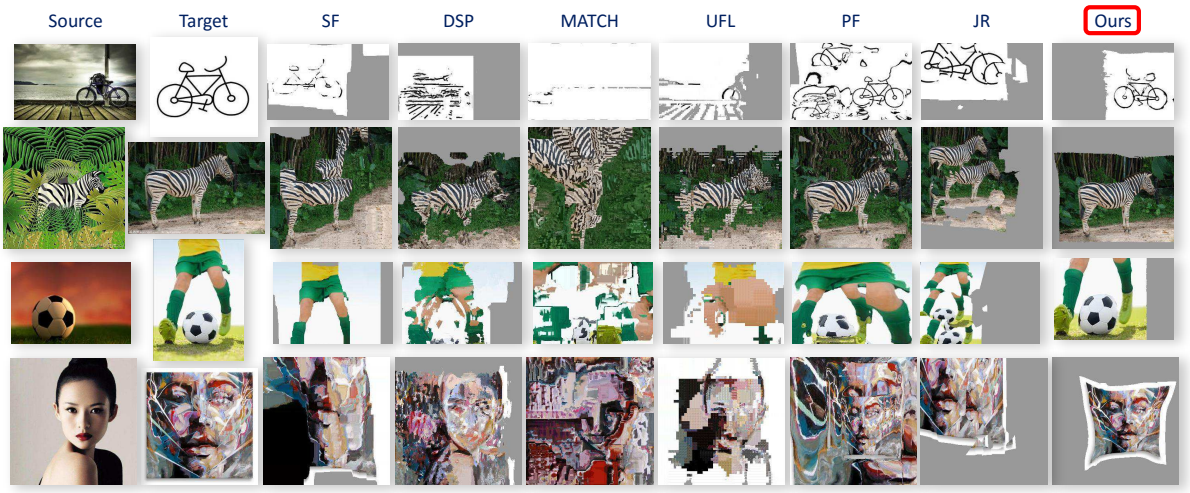


Figure 7. **Qualitative results on real-world images.** We warp *Target* to *Source* using the estimated dense correspondences to illustrate the result. Good results should be those whose colors and textures keep unchanged, and shapes close to their corresponding contents.

groundtruth flow fields, we can transfer the segmentation mask to evaluate the performance of dense semantic correspondence method on images with appearance variations like many previous works did [14] [11] [33].

Following the evaluation methods in many pervious works [14] [11] [33], we randomly select 15 pairs of images from each object category and evaluate the annotation transfer accuracy using three different metrics including Label Transfer Accuracy (LT-ACC), IoU metric and the Localization Error (LOC-ERR). The results of all methods are summarized in Tab. 2. Our method also achieves the highest score.

4.3. Results on real-world images

Fig. 7 offers the image reconstruction results obtained by warping the target image back to the source image using the estimated dense correspondences. These real-world image pairs are only semantically related and extremely challenging. A good resulting image should have the shapes and locations of the source image and the colors and textures of the target image. Recent approaches have been shown to fail in many challenging real-world image pairs, but our approach obtains very good qualitative results.

4.4. Run-time

The average running time of each method is tested on a PC with an i7 2.50 GHz CPU and 8 GB RAM. Our method is implemented by using MATLAB with unoptimized codes. It takes an average of 98.7 seconds for our method to handle each image pair (400×300). It takes longer than some high efficient methods, e.g., SF [18] and DSP [14], yet it achieves the best performance. Compared with some highly robust methods, such as JR [25], our method is much more efficient. We believe that a parallel implementation of our method will greatly improve its computational efficiency, because training weighted vector for each node is totally independent.

Table 2. **Results on the Caltech-101 dataset.**

Methods	LT-ACC	IoU	LOC-ERR
SF [18]	0.70	0.46	0.35
DSP [14]	0.75	0.51	0.32
DFF [32]	0.62	0.42	0.40
MATCH [26]	0.73	0.44	0.38
UFL [33]	0.67	0.46	0.43
PF [11]	0.79	0.52	0.26
JR [25]	0.76	0.48	0.33
Ours	**0.81**	**0.55**	**0.19**

5. Conclusion

In this paper, we have proposed Object-aware Hierarchical Graph (OHG) model, a novel framework for semantic dense correspondence estimation. Different from existing works, we estimate dense correspondences from semantic to low-level by training a discriminative classifier for each node in the upper two layers and guiding the matching of local structures. A better visual similarity is also defined in this paper. Our method achieves satisfactory results on two challenging benchmarks. In the future, we will improve the accuracy of our dense correspondence estimation by acquiring other invariant properties, e.g., rotation invariant.

Acknowledgements. The research was partly supported by the NSFC (No.6157021026) and a MSRA research project. Y. Fan' participation was supported by the NSFC (No.61370073). L. Xin' participation was supported in part by the "863" program (No.2015AA016010) and the MSTP of Dongguan (No.2015215102).

References

[1] E. Ahmed, S. Cohen, and B. Price. Semantic object selection. In *CVPR*, pages 3150–3157, 2014.

[2] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3d chairs: Exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, June 2014.

[3] H. Bristow, J. Valmadre, and S. Lucey. Dense semantic correspondence where every pixel is a classifier. In *ICCV*, December 2015.

[4] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *CVPR*, pages 41–48, 2009.

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36, 2004.

[6] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman. Personalizing human video pose estimation. In *CVPR*, 2016.

[8] M. Everingham. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[9] L. Feifei, R. Fergus, and P. Perona. One-shot learning of object categories. *TPAMI*, 28(4):594–611, 2006.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

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

[12] T. Hassner, V. Mayzels, and L. Zelnik-Manor. On sifts and their scales. In *CVPR*, pages 1522–1528, June. 2012.

[13] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *TPAMI*, 36(11):2144–2158, 2014.

[14] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, pages 2307–2314, 2013.

[15] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *TOG*, 23(3):689–694, 2004.

[16] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *TPAMI*, 33(12):2368–2382, 2011.

[17] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 33(5):978–994, 2011.

[18] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman. Sift flow: dense correspondence across difference scenes. In *ECCV*, pages 28–42, 2008.

[19] Y. Lu, L. Zhang, J. Liu, and Q. Tian. Constructing concept lexica with small semantic gaps. *IEEE Transactions on Multimedia*, 12(4):288–299, 2010.

[20] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[21] K. Prazdny. Egomotion and relative depth map from optical flow. *Biological Cybernetics*, 36(1):87–102, 1981.

[22] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, June 2013.

[23] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000.

[24] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. *TOG*, 30(6), 2011.

[25] T. Taniai, S. N. Sinha, and Y. Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR*, June 2016.

[26] M. Tau and T. Hassner. Dense correspondences across scenes and scales. *TPAMI*, 38(5):875–888, 2016.

[27] E. Trulls, I. Kokkinos, and F. Sanfeliu, A. andMoreno-Noguer. Dense segmentation-aware descriptors. In *CVPR*, pages 2890–2897, 2013.

[28] J. R. Uijlings, K. E. Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[29] T. Van Nguyen and J. Sepulveda. Salient object detection via augmented hypotheses. In *IJCAI*, 2015.

[30] P. A. Viola and M. J. Jones. Robust real-time face detection. In *ICCV*, volume 2, page 747. Citeseer, 2001.

[31] L. Xin, Y. Fan, C. Leiting, and C. Hongbin. Saliency transfer: An example-based method for salient object detection. In *IJCAI*, 2016.

[32] H. Yang, W.-Y. Lin, and J. Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *CVPR*, pages 3406–3413, June 2014.

[33] C. Zhang, C. Shen, and T. Shen. Unsupervised feature learning for dense correspondences across scenes. *IJCV*, 116(1):90–107, 2016.

[34] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Měch. Unconstrained salient object detection via proposal subset optimization. In *CVPR*, 2016.

[35] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *CVPR*, pages 117–126, 2016.