Proposal Flow: Semantic Correspondences from Object Proposals

Bumsub Ham, Minsu Cho, Cordelia Schmid, Jean Ponce

Proposal Flow: Semantic Correspondences from Object Proposals

Bumsub Ham, *Member, IEEE,* Minsu Cho, Cordelia Schmid, *Fellow, IEEE* and Jean Ponce, *Fellow, IEEE*

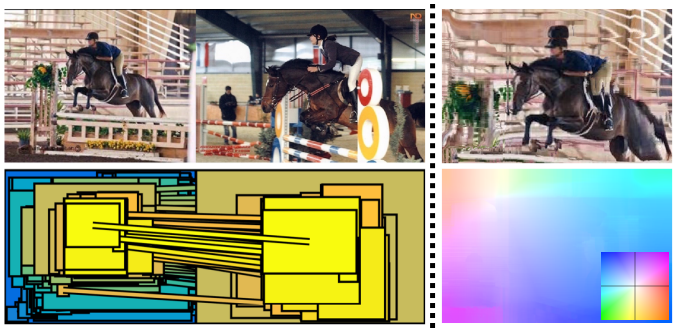
Abstract—Finding image correspondences remains a challenging problem in the presence of intra-class variations and large changes in scene layout. Semantic flow methods are designed to handle images depicting different instances of the same object or scene category. We introduce a novel approach to semantic flow, dubbed proposal flow, that establishes reliable correspondences using object proposals. Unlike prevailing semantic flow approaches that operate on pixels or regularly sampled local regions, proposal flow benefits from the characteristics of modern object proposals, that exhibit high repeatability at multiple scales, and can take advantage of both local and geometric consistency constraints among proposals. We also show that the corresponding sparse proposal flow can effectively be transformed into a conventional dense flow field. We introduce two new challenging datasets that can be used to evaluate both general semantic flow techniques and region-based approaches such as proposal flow. We use these benchmarks to compare different matching algorithms, object proposals, and region features within proposal flow, to the state of the art in semantic flow. This comparison, along with experiments on standard datasets, demonstrates that proposal flow significantly outperforms existing semantic flow methods in various settings.

Index Terms—Semantic flow, object proposals, scene alignment, dense scene correspondence.

1 Introduction

Classical approaches to finding correspondences across images are designed to handle scenes that contain the same objects with moderate view point variations in applications such as stereo matching [1], [2], optical flow [3], [4], [5], and wide-baseline matching [6], [7]. *Semantic flow* methods, such as SIFT Flow [8] for example, on the other hand, are designed to handle a much higher degree of variability in appearance and scene layout, typical of images depicting different instances of the same object or scene category. They have proven useful for many tasks such as object recognition, cosegmentation, image registration, semantic segmentation, and image editing and synthesis [7], [8], [9], [10], [11], [12], [13]. In this context, however, appearance and shape variations may confuse similarity measures for local region matching, and prohibit the use of strong geometric constraints (e.g., epipolar geometry, limited disparity range). Existing approaches to semantic flow are thus easily distracted by scene elements specific to individual objects and image-specific details (e.g., background, texture, occlusion, clutter). This is the motivation for our work, where we use reliable and robust region correspondences to focus on regions containing prominent objects and scene elements rather than clutter and distracting details.

Concretely, we introduce an approach to pairwise semantic flow computation, called *proposal flow*, that establishes region correspondences using object proposals and their geometric relations (Fig. 1). Unlike previous semantic flow algorithms [7], [8], [9], [10], [11], [13], [14], [15], [16], [17], [18], [19], [20], [21], that use regular grid structures for local region generation and matching, we leverage a large number of multi-scale object proposals [22], [23], [24], [25], [26], as now widely used to significantly reduce the search space or false alarms, e.g, for object detection [27], [28] and tracking tasks [29]. Using object proposals for semantic flow has the following advantages: First, we can use diverse spatial supports for prominent objets and parts, and focus on these elements rather than clutter and distracting scene components. Second, we can use geometric relations between objects and parts, which prevents confusing objects with visually similar regions or parts, but quite different geometric configurations. Third, as in the case of object detection, we can reduce the search space for correspondences, scaling well with the size of the image collection. Accordingly, the proposed approach establishes region



(a) Region-based semantic flow. (b) Dense flow field.

Fig. 1. Proposal flow generates a reliable and robust semantic flow between similar images using local and geometric consistency constraints among object proposals, and it can be transformed into a dense flow field. Using object proposals for semantic flow enables focusing on regions containing prominent objects and scene elements rather than clutter and distracting details. (a) Region-based semantic flow between source (left) and target (right) images. (b) Dense flow field (bottom) and image warping using the flow field (top). **(Best viewed in color.)**

- *Bumsub Ham is with the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea. E-mail: mimo@yonsei.ac.kr.*
- *Minsu Cho is with the Department of Computer Science and Engineering, POSTECH, Pohang, Korea. E-mail: mscho@postech.ac.kr.*
- *Cordelia Schmid is with Thoth project-team, Inria Grenoble Rhône-Alpes, Laboratoire Jean Kuntzmann, France. E-mail: cordelia.schmid@inria.fr.*
- *Jean Ponce is with École Normale Supérieure / PSL Research University and Inria. E-mail: jean.ponce@ens.fr.*

correspondences between object proposals by exploiting their visual features and geometric relations in an efficient manner, and generates a region-based semantic flow composed of object proposal matches. We show that this region-based proposal flow can be effectively transformed into a conventional dense flow field. We also introduce new datasets and evaluation metrics that can be used to evaluate both general semantic flow techniques and region-based approaches such as proposal flow. These datasets consist of images containing more clutter and intra-class variation, and are much more challenging than existing ones for semantic flow evaluation. We use these benchmarks to compare different matching algorithms, object proposals, and region features within proposal flow, to the state of the art in semantic flow. This comparison, along with experiments on standard datasets, demonstrates that proposal flow significantly outperforms existing semantic flow methods (including a learning-based approach) in various settings.

Contributions. The main contributions of this paper can be summarized as follows:

- We introduce the proposal flow approach to establishing robust region correspondences between related, but not identical scenes using object proposals (Section 3).
- We introduce benchmark datasets and evaluation metrics for semantic flow that can be used to evaluate both general semantic flow algorithms and region matching methods (Section 4).
- We demonstrate the advantage of proposal flow over state-of-the-art semantic flow methods through extensive experimental evaluations (Section 5).

A preliminary version of this work appeared in [30]. Besides a more detailed presentation and discussion of the most recent related works, this version adds (1) an in-depth presentation of proposal flow; (2) a more challenging benchmark based on the PASCAL 2011 keypoint dataset [31]; (3) a verification of quality of ground-truth correspondence generation for our datasets; (4) an extensive experimental evaluation including a performance analysis with varying the number of proposals and an analysis of runtime, and a comparison of proposal flow with recently introduced state-of-the-art methods and datasets. To encourage comparison and future work, our datasets and code are available online: http://www.di.ens.fr/willow/research/proposalflow.

2 Related work

Correspondence problems involve a broad range of topics beyond the scope of this paper. Here we briefly describe the context of our approach, and only review representative works pertinent for ours.

2.1 Semantic flow

Pairwise correspondence. Classical approaches to stereo matching and optical flow estimate dense correspondences between pairs of nearby images of the same scene [1], [3], [6]. While advances in invariant feature detection and description have revolutionized object recognition and reconstruction in the past 15 years, research on image matching and alignment between images have long been dominated by instance matching with the same scene and objects [32]. Unlike these, several recent approaches to semantic flow focus on handling images containing different scenes and objects. Graph-based matching algorithms [12], [33] attempt to find category-level feature matches by leveraging a flexible graph representation of images, but they commonly handle sparsely sampled or detected features due to their computational complexity. Inspired by classic optical flow algorithms, Liu *et al*. pioneered the idea of dense correspondences across different scenes, and proposed the SIFT Flow [8] algorithm that uses a multi-resolution image pyramid together with a hierarchical optimization technique for efficiency. Kim *et al*. [10] extende the approach by inducing a multi-scale regularization with a hierarchically connected pyramid of grid graphs. Long *et al*. [34] investigate the effect of pretrained ConvNet features on the SIFT Flow algorithm, and Bristow *et al*. [14] propose an exemplar-LDA approach that improves the performance of semantic flow. More recently, Taniai *et al*. [13] have shown that the approach to jointly recovering cosegmentation and dense correspondence outperforms state-of-the-art methods designed specifically for either cosegmentation or correspondence estimation. Zhou *et al*. [21] propose a learning-based method that leverages a 3D model. This approach uses cycle consistency to link the correspondence between real images and rendered views. Choy *et al*. [20] propose to use a fully convolutional architecture, along with a correspondence contrastive loss, allowing faster training by effective reuse of computations. While archiving state-of-the-art performance, these learning-based approaches require a large number of annotated images [20] or 3D models [21] to train the corresponding deep model, and do not consider geometric consistency among correspondences.

Despite differences in graph construction, optimization, and similarity computation, existing semantic flow approaches share grid-based regular sampling and spatial regularization: The appearance similarity is defined at each region or pixel on (a pyramid of) regular grids, and spatial regularization is imposed between neighboring regions in the pyramid models [8], [10], [13], [34]. In contrast, our work builds on generic object proposals with diverse spatial supports [22], [23], [24], [25], [26], and uses an irregular form of spatial regularization based on co-occurrence and overlap of the proposals. We show that the use of local regularization with object proposals yields substantial gains in generic region matching and semantic flow, in particular when handling images with significant clutter, intra-class variations and scaling changes, establishing a new state of the art on the task.

Multi-image correspondence. Besides these pairwise matching methods, recent works have tried to solve a correspondence problem as a joint image-set alignment. Collection Flow [35] uses an optical flow algorithm that aligns each image to its low-rank projection onto a sub-space capturing the common appearance of the image collection. FlowWeb [11] first builds a fully-connected graph with each image as a node, and each edge as flow field between a pair of images, and then establishes globally-consistent correspondences using cycle consistency among all edges. This approach gives state-of-the-art performance, but requires a large number of images for each object category, and the matching results are largely dependent on the initialization quality. Zhou *et al*. [36] also use cycle consistency between sparse features to solve a graph matching problem posed as a low-rank matrix recovery. Carreira *et al*. [37] leverage keypoint annotations to estimate dense correspondences across images with similar viewpoint, and use these pairwise matching results to align a query image to all the other images to perform single-view 3D reconstruction.

While improving over pairwise correspondence results at the expense of runtime, these multi-image methods all use a pairwise method to find initial matches before refining them, (e.g., with cycle consistency [36]). Our correspondence method outperforms current pairwise methods, and its output could be used as a good initialization for multi-image methods.

2.2 Object proposals and object-centric representations

Object proposals [22], [23], [24], [25], [26] have originally been developed for object detection, where they are used to reduce the search space as well as false alarms. They are now an important component in many state-of-the-art detection pipelines [27], [28] and other computer vision applications, including object tracking [29], action recognition [38], weakly supervised localization [39], and semantic segmentation [40]. Despite their success for object detection and segmentation, object proposals have seldom been used in matching tasks [41], [42]. In particular, while Cho *et al*. [41] have shown that object proposals are useful for region matching due to their high repeatability on salient part regions, the use of object proposals has never been thoroughly investigated in semantic flow computation. The approach proposed in this paper is a first step in this direction, and we explore how the choice of object proposals, matching algorithms, and features affects matching robustness and accuracy.

Recently, object-centric representation has been used to estimate optical flow. In [43], potentially moving vehicles are first segmented from the background, and the flow is estimated individually for every object and the background. Similarly, Sevilla-Lara *et al*. [44] use semantic segmentation to break the image into regions, and compute optical flow differently in different regions, depending on the the semantic class label. The main intuition behind these works is that focusing on regions containing prominent regions, e.g., objects, can help estimate the optical flow field effectively. Proposal flow shares similar idea, but it is designed for semantic flow computation and leverages the geometric relations between objects and parts as well. We show that object proposals are well suited to semantic flow computation, and further using their geometric relations boosts the matching accuracy.

3 Proposal Flow

Proposal flow can use any type of object proposals [22], [23], [24], [25], [26], [45] as candidate regions for matching a pair of images of related scenes. In this section, we introduce a probabilistic model for region matching (Section 3.1), and describe three matching strategies including two baselines and a new one using local regularization (Section 3.2). We then describe our approach to generating a dense flow field from the region matches (Section 3.3).

3.1 A Bayesian model for region matching

Let us suppose that two sets of object proposals $\mathcal{R}$ and $\mathcal{R}'$ have been extracted from images $\mathcal{I}$ and $\mathcal{I}'$ (Fig. 2(a-b)). A proposal r in $\mathcal{R}$ is an image region $r = (f, s)$ with appearance feature f and spatial support s. The appearance feature represents a visual descriptor for the region (e.g., SPM [47], HOG [46], ConvNet [48]), and the spatial support describes the set of all pixel positions in the region (a rectangular box in this work). Given the data $\mathcal{D} = (\mathcal{R}, \mathcal{R}')$, we wish to estimate a posterior probability of the event $r \mapsto r'$ meaning that proposal r in $\mathcal{R}$ matches proposal r' in $\mathcal{R}'$:

$$p(r \mapsto r' \mid \mathcal{D}) = p(f \mapsto f')p(s \mapsto s' \mid \mathcal{D}), \tag{1}$$

where we decouple appearance and geometry, and further assume that appearance matching is independent of the data $\mathcal{D}$. In practice, the appearance term $p(f \mapsto f')$ is simply computed from a similarity between feature descriptors f and f', and the geometric consistency term $p(s \mapsto s' \mid \mathcal{D})$ is evaluated by comparing the spatial supports s and s' in the context of the given data $\mathcal{D}$, as described in the next section. We set the posterior probability $p(r \mapsto r' \mid \mathcal{D})$ as a matching score and assign the best match $\phi(r)$ for each proposal in $\mathcal{R}$:

$$\phi(r) = \underset{r' \in \mathcal{R}'}{\operatorname{argmax}}\, p(r \mapsto r' \mid \mathcal{D}). \tag{2}$$

Using a slight abuse of notation, if $(f', s') = \phi(f, s)$, we will write $f' = \phi(f)$ and $s' = \phi(s)$.

3.2 Geometric matching strategies

We now introduce three matching strategies, using different geometric consistency terms $p(s \mapsto s' \mid \mathcal{D})$.

3.2.1 Naive appearance matching (NAM)

A straightforward way of matching regions is to use a uniform distribution for the geometric consistency term $p(s \mapsto s' \mid \mathcal{D})$ so that

$$p(r \mapsto r' \mid \mathcal{D}) \propto p(f \mapsto f'). \tag{3}$$

NAM considers appearance only, and does not reflect any geometric relationship among regions (Fig. 2(d)).

3.2.2 Probabilistic Hough matching (PHM)

The matching algorithm in [41] can be expressed in our model as follows. First, a three-dimensional location vector (position plus scale) is extracted from the spatial support s of each proposal r[1]. We denote it by a function γ. An offset space $\mathcal{X}$ is defined as a feasible set of offset vectors between $\gamma(s)$ and $\gamma(s')$: $\mathcal{X} = \{\gamma(s) - \gamma(s') \mid r \in \mathcal{R}, r' \in \mathcal{R}'\}$. The geometric consistency term $p(s \mapsto s' \mid \mathcal{D})$ is then defined as

$$p(s \mapsto s' \mid \mathcal{D}) = \sum_{x \in \mathcal{X}} p(s \mapsto s' \mid x)p(x \mid \mathcal{D}), \tag{4}$$

which assumes that the probability $p(s \mapsto s' \mid x)$ that two boxes s and s' match given the offset x is independent of the rest of the data and can be modeled by a Gaussian kernel in the three-dimensional offset space. Given this model, PHM replaces $p(x \mid \mathcal{D})$ with a generalized Hough transform score:

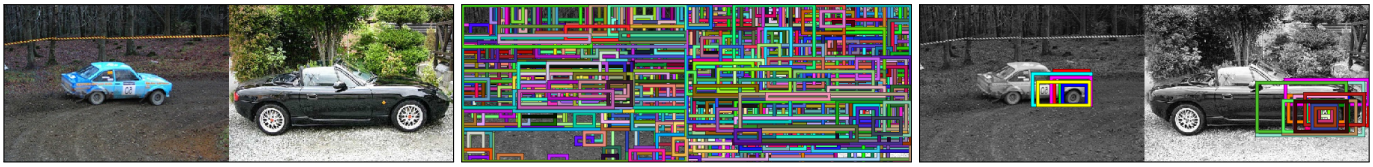
$$h(x \mid \mathcal{D}) = \sum_{(r,r') \in \mathcal{D}} p(f \mapsto f')p(s \mapsto s' \mid x), \tag{5}$$

which aggregates individual votes for the offset x, from *all* possible matches in $\mathcal{D} = \mathcal{R} \times \mathcal{R}'$. Hough voting imposes a spatial regularizer on matching by taking into account a *global* consensus on the corresponding offset [49], [50]. However, it often suffers from background clutter that distracts the global voting process (Fig. 2(e)).

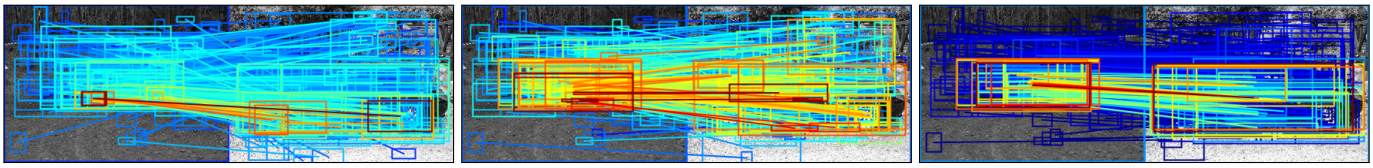
3.2.3 Local offset matching (LOM)

Here we propose a new method to overcome this drawback of PHM [41] and obtain more reliable correspondences. Object proposals often contain a large number of distracting outlier regions from background clutter, and are not perfectly repeatable even for corresponding object or parts across different images (Fig. 2(c)). The global Hough voting in PHM has difficulties with such outlier regions. In contrast, we optimize a translation and scale offset for

1. The location vector consists of center coordinate and area of the spatial supprot s.



(a) Input images. (b) Object proposals [25]. (c) Object proposals near the front wheel.



(d) NAM. (e) PHM [41]. (f) LOM.

Fig. 2. **Top:** (a-b) A pair of images and their object proposals [25]. (c) Multi-scale object proposals contain the same object or parts, but they are not perfectly repeatable across different images. **Bottom:** In contrast to NAM (d), PHM [41] (e) and LOM (f) both exploit geometric consistency, which regularizes proposal flow. In particular, LOM imposes local smoothness on offsets between neighboring regions, avoiding the problem of using a global consensus on the offset in PHM [41]. The matching score is color-coded for each match (red: high, blue: low). The HOG descriptor [46] is used for appearance matching in this example. **(Best viewed in color.)**

each proposal by exploiting only neighboring proposals. That is, instead of averaging $p(s \mapsto s'|x)$ over all feasible offsets $\mathcal{X}$ in PHM, we use one reliable offset optimized for each proposal. This local approach substantially alleviates the effect of outlier regions in matching as will be demonstrated by our experiment results.

The main issue is how to estimate a reliable offset for each proposal r in a robust manner without any information about objects and their locations. One way would be to find the region corresponding to r through a multi-scale sliding window search in $\mathcal{I}'$ as in object detection [51], but this is expensive. Instead, we assume that nearby regions have similar offsets. For each region r, we first define its neighborhood $\mathcal{N}(r)$ as the set of regions with overlapping spatial support:

$$\mathcal{N}(r) = \{\hat{r} \mid s \cap \hat{s} \neq \emptyset, \hat{r} \in \mathcal{R}\}. \tag{6}$$

Using an initial correspondence $\psi(r)$, determined by the best match according to appearance, each neighboring region $\hat{r}$ is assigned its own offset, and all of them form a set of neighbor offsets:

$$\mathcal{X}(r) = \{\gamma(\hat{s}) - \gamma(\psi(\hat{s})) \mid \hat{r} \in \mathcal{N}(r)\}. \tag{7}$$

From this set of neighbor offsets, we estimate a local offset x_r^* for the region r by the geometric median [52][2]:

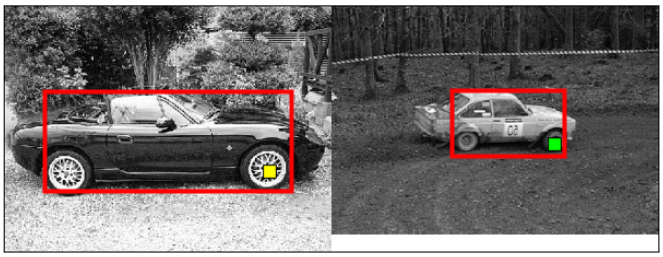
$$x_r^* = \underset{x \in \mathbb{R}^3}{\operatorname{argmin}} \sum_{y \in \mathcal{X}(r)} \|x - y\|_2 , \tag{8}$$

which can be computed using Weiszfeld's algorithm [54] with a form of iteratively re-weighted least squares. In other words, the local offset x_r^* for the region r is estimated by regression using its local neighboring offsets $\mathcal{X}(r)$. Based on the local offset x_r^* optimized for each region, we define the geometric consistency function:

$$g(s \mapsto s'|\mathcal{D}) = p(s \mapsto s'|x_r^*) \sum_{\hat{r} \in \mathcal{N}(r)} p(\hat{f} \mapsto \psi(\hat{f})), \tag{9}$$

which can be interpreted as the fact that the region r in $\mathcal{R}$ is likely to match r' in $\mathcal{R}'$ where its offset $\gamma(s) - \gamma(s')$ is close to the local offset x_r^*, and the region r has many neighboring matches with a high appearance fidelity. By using $g(s \mapsto s'|\mathcal{D})$ as a proxy for $p(s \mapsto s'|\mathcal{D})$, LOM imposes local smoothness on offsets between neighboring regions. This geometric consistency function effectively suppresses matches between clutter regions, while favoring matches between regions that contain objects rather than object parts (Fig. 2(f)).

2. We found that the centroid and mode of the offset vectors in three-dimensional offset space show worse performance than the geometric median. This is because the neighboring regions may include clutter. Clutter causes incorrect neighbor offsets, but the geometric median is robust to outliers [53], providing a reliable local offset.



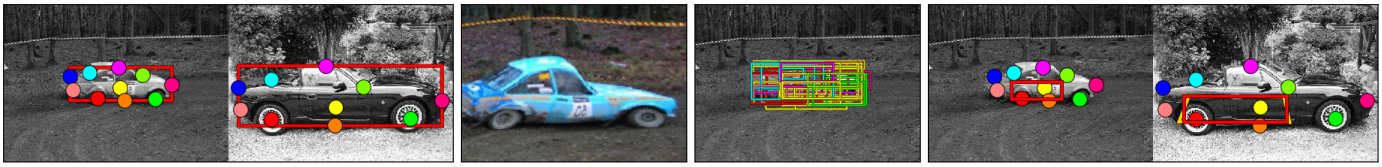
(a) Anchor match and pixel correspondence.

(b) Match visualization. (c) Warped image.

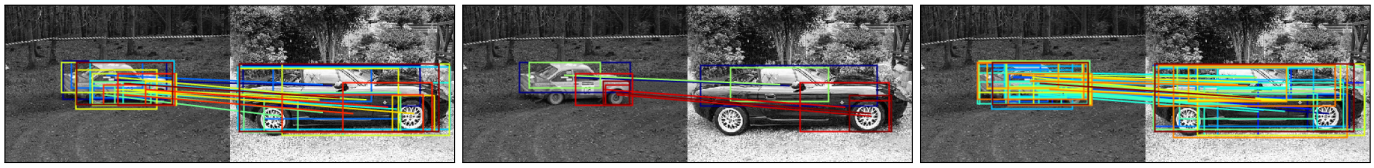
Fig. 3. Flow field generation. (a) For each pixel (yellow point), its anchor match (red boxes) is determined. The correspondence (green point) is computed by the transformed coordinate with respect to the position and size of the anchor match. (b) Based on the flow field, (c) the right image is warped to the left image. The warped object shows visually similar shape to the one in the left image. The LOM method is used for region matching with the object proposals [24] and the HOG descriptor [46]. **(Best viewed in color.)**

3.3 Flow field generation

Proposal flow gives a set of region correspondences between images that can easily be transformed into a conventional dense flow field. Let p denote a pixel in the image $\mathcal{I}$ (yellow point in Fig. 3(a)). For each pixel p, its neighborhood is defined as the region in which it lies, i.e., $\mathcal{N}(p) = \{r \in \mathcal{R} : p \in r\}$. We define an anchor match $(r^*, \phi(r^*))$ as the region correspondence that



(a) Keypoints and object bounding boxes. (b) Warping. (c) $\mathcal{R}_s$. (d) Ground-truth correspondence



(e) NAM. (f) PHM [41]. (g) LOM.

Fig. 4. **Top:** Generating ground-truth regions and evaluating correct matches. (a) Using keypoint annotations, dense correspondences between images are established using TPS warping [55], [56]. (b) Based on the dense correspondences, all pixels in the left image are warped to the right image, showing that the correspondences align two images well. (c) We assume that true matches exist only between the regions near the object bounding box, and thus an evaluation is done with the regions in this subset of object proposals. (d) For each object proposal (red box in the left image), its ground truth is generated automatically by the dense correspondences: We fit a tight rectangle (red box in the right image) of the region formed by the warped object proposal (yellow box in the right image) and use it as a ground-truth correspondence. **Bottom:** Examples of correct matches: The numbers of correct matches are 16, 5, and 38 for NAM (e), PHM [41] (f), and LOM (g), respectively. Matches with an IoU score greater than 0.5 are considered as correct in this example. **(Best viewed in color.)**

has the highest matching score among neighboring regions (red boxes in Fig. 3(a)) where

$$r^* = \underset{r \in \mathcal{N}(p)}{\operatorname{argmax}}\, p(r \mapsto \phi(r) \mid \mathcal{D}). \tag{10}$$

Note that the anchor match contains information on translation and scale changes between objects or part regions. Using the geometric relationships between the pixel p and its anchor match $(r^*, \phi(r^*))$, a correspondence p' in the image $\mathcal{I}'$ (green point in Fig. 3(a)) is obtained by linear interpolation, i.e., computed by the transformed coordinate with respect to the position and size of the anchor match.

The matching score for each correspondence p is set to the value of its anchor match $(r^*, \phi(r^*))$. When the pixels p and q in the image $\mathcal{I}$ are matched to the same pixel p' in the image $\mathcal{I}'$, we select the match with the highest matching score and delete the other one. Finally, joint image filtering [57] is applied under the guidance of the image $\mathcal{I}$ to interpolate the flow field in places without correspondences. Figure 3(b-c) shows examples of the estimated flow field and corresponding warping result between two images: Using the dense flow field, we warp all pixels in the right image to the left image. Our approach using the anchor match aligns semantic object parts well while handling translation and scale changes between objects.

4 Datasets for Semantic Flow Evaluation

Current research on semantic flow lacks appropriate benchmarks with dense ground-truth correspondences. Conventional optical flow benchmarks (e.g., Middlebury [58] and MPI-Sintel [59]) do not feature within-class variations, and ground truth for generic semantic flow is difficult to capture due to its intrinsically semantic nature, manual annotation being extremely labor intensive and somewhat subjective. Existing approaches are thus usually evaluated only with sparse ground truth or in an indirect manner (e.g. mask transfer accuracy) [8], [10], [11], [14], [17], [18]. Such benchmarks only evaluate a small number of matches, that occur at ground-truth keypoints or around mask boundaries in a point-wise manner. To address this issue, we introduce in this section two new datasets for semantic flow, dubbed PF-WILLOW and PF-PASCAL (PF for proposal flow), built using ground-truth object bounding boxes and keypoint annotations, (Fig. 4(a)), and propose new evaluation metrics for region-based semantic flow methods. Note that while designed for region-based methods, our benchmark can be used to evaluate any semantic flow technique. As will be seen in our experiments, it provides a reasonable (if approximate) ground truth for dense correspondences across similar scenes without an extremely expensive annotation campaign. Comparative evaluations on this dataset have also proven to be good predictors for performance on other tasks and datasets, further justifying the use of our benchmark.

Taniai *et al*. have recently introduced a benchmark dataset for semantic flow evaluation [13]. It provides 400 image pairs of 7 object categories, corresponding ground-truth cosegmentation masks, and flow maps that are obtained by natural neighbor interpolation [60] on sparse keypoint matches. In contrast, our datasets use over 2200+ image pairs of up to 20 categories. It is split into two subsets: The first subset features 900 image pairs of 4 object categories, further split into 10 sub-categories according to the viewpoint and background clutter, in order to evaluate the different factors of variation for matching accuracy. The second subset consists of 1300+ image pairs of 20 image categories. In the following, we present our ground-truth generation process in Section 4.1, evaluation criteria in Section 4.2, and datasets in Section 4.3.

4.1 Ground-truth correspondence generation

Let us assume two sets of keypoint annotations at positions k_i and k'_i in $\mathcal{I}$ and $\mathcal{I}'$, respectively, with $i = 1, \ldots, m$. Assuming the objects present in the images and their parts may undergo shape deformation, we use thin plate splines (TPS) [55], [56] to interpolate sparse keypoints (Fig. 4(b)). Concretely, the ground truth is approximated from sparse correspondences using TPS warping. For each region or proposal, its ground-truth match is generated as follows. We assume that each image has a single object and true matches only exist between a subset of regions, i.e., regions around object bounding boxes (Fig. 4(c)): $\mathcal{R}_s = \{r \mid |b \cap r| \,/\, |r| \geq 0.75, r \in \mathcal{R}\}$ where b denotes an object bounding box in the image $\mathcal{I}$, and $|r|$ indicates the area of the

region r. For each region $r \in \mathcal{R}_s$ (e.g., red box in Fig. 4(d) left), the four vertices of the rectangle are warped to the corresponding ones in the image $\mathcal{I}'$ by the TPS mapping function (e.g., yellow box in Fig. 4(d) right). The region formed by the warped points is a correspondence of region r. We fit a tight rectangle for this region and set it as a ground-truth correspondence for the region r (e.g., red box in Fig. 4(d) right).

Note that WarpNet [61] also uses TPS to generate ground-truth correspondences, but it does not consider intra-class variation. In particular, WarpNet constructs a pose graph using a fine-grained dataset (e.g., the CUB-200-2111 [62] of bird categories), computes a set of TPS functions using silhouettes of image pairs that are closest on the graph, and finally transforms each image by sampling from this set of TPS warps. In contrast to this, we directly use TPS to estimate a warping function using ground-truth keypoint annotations.

4.2 Evaluation criteria

We introduce two evaluation metrics for region matching performance in terms of matching precision and match retrieval accuracy. These metrics build on the intersection over union (IoU) score between the region r's correspondence $\phi(r)$ and its ground truth $r^\star$:

$$\text{IoU}(\phi(r), r^\star) = |\phi(r) \cap r^\star| \,/\, |\phi(r) \cup r^\star|. \tag{11}$$

For region matching precision, we propose the probability of correct region (PCR) metric where the region r is correctly matched to its ground truth $r^\star$ if $1 - \text{IoU}(\phi(r), r^\star) < \tau$ (e.g., Fig. 5(a) top), where τ is an IoU threshold. Note that this region-based metric is based on a conventional point-based metric, the probability of correct keypoint (PCK) [63]. In the case of pixel-based flow, PCK can be adopted instead. We measure the PCR metric while varying the IoU threshold τ from 0 to 1. For match retrieval accuracy, we propose the average IoU of k-best matches (dubbed mIoU@k) according to the matching score (e.g., Fig. 5(a) bottom). We measure the mIoU@k metric while increasing the number of top matches k. These two metrics exhibit two important characteristics of matching: PCR reveals the accuracy of overall assignment, and mIoU@k shows the reliability of matching scores that is crucial in match selection.

4.3 Dataset construction

We construct two benchmark datasets for semantic flow evaluation: The PF-WILLOW and PF-PASCAL datasets. The original images and keypoint annotations are taken from existing datasets [31], [64].

PF-WILLOW. To generate the PF-WILLOW dataset, we start from the benchmark for sparse matching of Cho *et al.* [64], which consists of 5 object classes (Face, Car, Motorbike, Duck, WineBottle) with 10 keypoint annotations for each image. Note that these images contain more clutter and intra-class variation than existing datasets [10], [11], [17] for semantic flow evaluation, which include mainly images with tightly cropped objects or similar background. We exclude the face class where the number of generated object proposals is not sufficient to evaluate matching accuracy. The other classes are split into sub-classes[3] according to viewpoint or background clutter. We obtain a total of 10 sub-classes. Given these images and regions, we generate ground-truth data between all possible image pairs within each sub-class. The dataset has 10 images for each sub-class, thus 100 images and 900 image pairs in total.

PF-PASCAL. For the PF-PASCAL dataset, we use PASCAL 2011 keypoint annotations [31] for 20 object categories. We select meaningful image pairs for each category that contain a single object with similar poses, resulting in 1351 image pairs in total. The number of image pairs in the dataset varies from 6 for the sheep class to 140 for the bus class, and 67 on average, and each image pair contains from 4 to 17 keypoints and 7.95 keypoints on average. This dataset is more challenging than PF-WILLOW and other existing datasets for semantic flow evaluation.

5 Experiments

In this section we present a detailed analysis and evaluation of our proposal flow approach.

5.1 Experimental details

Object proposals. We evaluate four state-of-the-art object proposal methods: EdgeBox (EB) [26], multi-scale combinatorial grouping (MCG) [22], selective search (SS) [25], and randomized prim (RP) [24]. In addition, we consider three baseline proposals [23]: Uniform sampling (US), Gaussian sampling (GS), and sliding window (SW) (See [23] for a discussion). We use publicly available codes for all proposal methods.

For fair comparison, we use 1,000 proposals for all the methods in all experiments, unless otherwise specified. To control the number of proposals, we use the proposal score: Albeit not all having explicit control over the number of proposals, EB, MCG, and SS provides proposal scores, so we use the top k proposals. For RP, which lacks any control over the number of proposals, we randomly select the proposals. For US, GS, and SW, we can control the number of proposals explicitly [23].

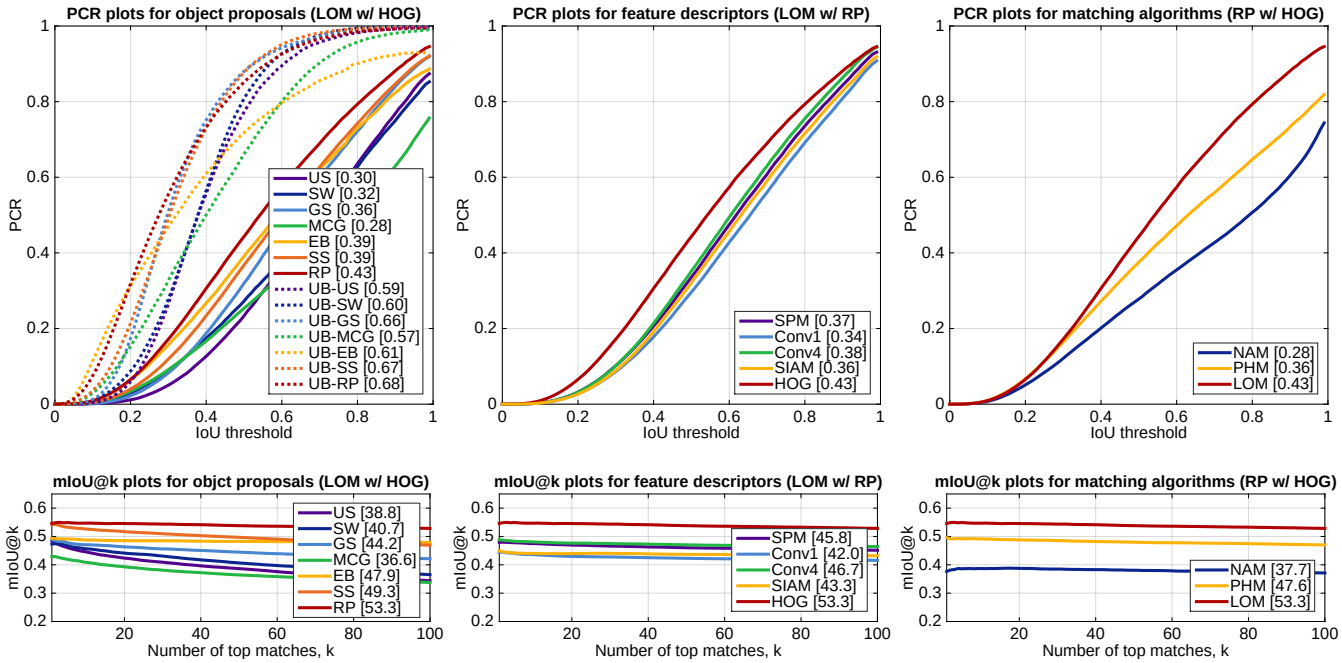
Feature descriptors and similarity. We evaluate four popular feature descriptors: two engineered ones (SPM [47] and HOG [46]) and two learning-based ones (ConvNet [48] and SIAM [65]). For SPM, dense SIFT features [66] are extracted every 4 pixels and each descriptor is quantized into a 1,000 word codebook [67]. For each region, a spatial pyramid pooling [47] is used with 1×1 and 3×3 pooling regions. We compute the similarity between SPM descriptors by the χ^2 kernel. HOG features are extracted with 8×8 cells and 31 orientations, then whitened. For ConvNet features, we use each output of the 5 convolutional layers in AlexNet [48], which is pre-trained on the ImageNet dataset [68]. For HOG and ConvNet, the dot product is used as a similarity metric[4]. For SIAM, we use the author-provided model trained using a Siamese network on a subset of Liberty, Yosemite, and Notre Dame images of the multi-view stereo correspondence (MVS) dataset [69]. Following [65], we compute the similarity between SIAM descriptors by the l_2 distance.

5.2 Proposal flow components

We use the PF benchmarks in this section to compare three variants of proposal flow using different matching algorithms (NAM, PHM, LOM), combined with various object proposals [22], [23], [24], [25], [26], and features [46], [47], [48], [65].

3. They are car (S), (G), (M), duck (S), motorbike (S), (G), (M), wine bottle (w/o C), (w/ C), (M), where (S) and (G) denote side and general viewpoints, respectively. (C) stands for background clutter, and (M) denotes mixed viewpoints (side + general) for car and motorbike classes and a combination of images in wine bottle (w/o C + w/ C) for the wine bottle class.

4. We also tried the χ^2 kernel to compute the similarity between HOG or ConvNet features, and found that using the dot product gives better matching accuracy.



(a) Comparison of object proposals. (b) Comparison of feature descriptors. (c) Comparison of matching algorithms.

Fig. 5. PF-PASCAL benchmark evaluation on region matching precision (top, PCR plots) and match retrieval accuracy (bottom, mIoU@k plots): (a) Evaluation for LOM with HOG [46], (b) evaluation for LOM with RP [24], and (c) evaluation for RP with HOG [46]. The AuC is shown in the legend. **(Best viewed in color.)**

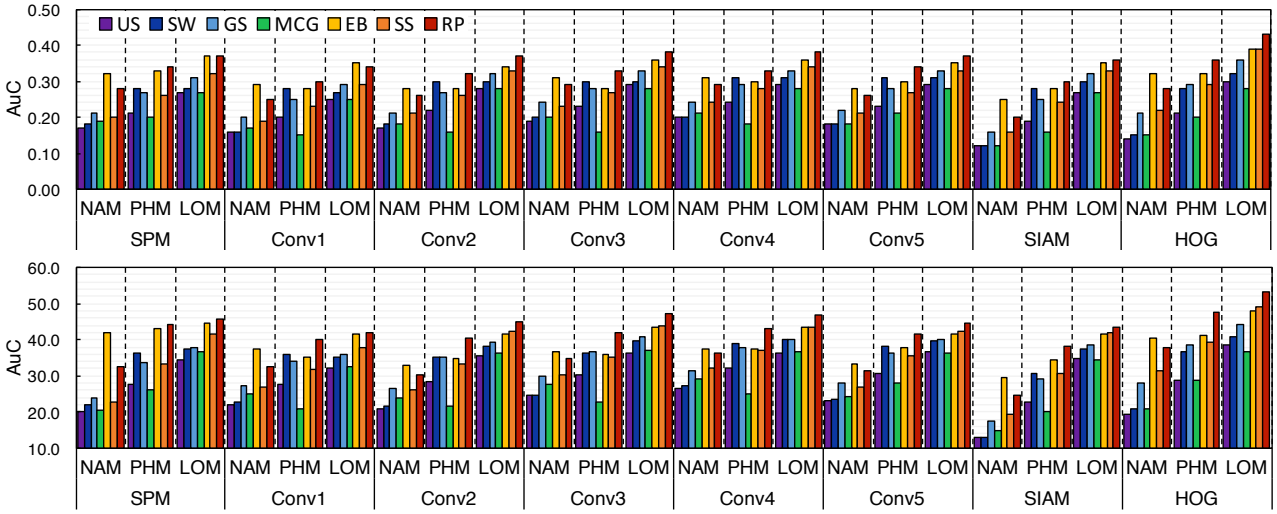


Fig. 6. PF benchmark evaluation on AuCs for PCR and mIoU@k plots with the PF-PASCAL dataset. We can see that combining LOM, RP, and HOG performs best in both metrics and datasets. **(Best viewed in color.)**

Qualitative comparison. Figure 4(e-g) shows a qualitative comparison between region matching algorithms on a pair of images and depicts correct matches found by each variant of proposal flow. In this example, at the IoU threshold 0.5, the numbers of correct matches are 16, 5, and 38 for NAM, PHM [41], and LOM, respectively. This shows that PHM may give worse performance than even NAM when there is much clutter in background. In contrast, the local regularization in LOM alleviates the effect of such clutter.

Quantitative comparison on PF-PASCAL. Figure 5 summarizes the matching and retrieval performance on average for all object classes with a variety of combination of object proposals, feature descriptors, and matching algorithms. Figure 5(a) compares different types of object proposals with fixed matching algorithm and feature descriptor (LOM w/ HOG). RP gives the best matching precision and retrieval accuracy among the object proposals. An upper bound on precision is measured for object proposals (around a given object) in the image $\mathcal{I}$ using corresponding ground truths in image $\mathcal{I}'$, that is the best matching accuracy we can achieve with each proposal method. To this end, for each region r in the image $\mathcal{I}$, we find the region r' in the image $\mathcal{I}'$ that has the highest IoU score given the region r's ground-truth correspondence $r^\star$ in the image $\mathcal{I}'$, and use the score as an upper bound precision. The upper bound (UB) plots show that RP generates more

TABLE 1
AuC performance for PCR plots on the PF-PASCAL dataset (RP w/ HOG).

Methods	aero	bike	bird	boat	bot	bus	car	cat	cha	cow	tab	dog	hor	mbik	pers	plnt	she	sofa	trai	tv	Avg.
LOM	0.52	0.56	0.34	0.39	0.47	0.61	0.58	0.34	0.43	0.43	0.27	0.36	0.46	0.48	0.31	0.34	0.35	0.37	0.52	0.50	0.43
Upper bound	0.70	0.72	0.63	0.66	0.71	0.77	0.73	0.63	0.72	0.69	0.57	0.67	0.70	0.72	0.66	0.62	0.53	0.65	0.73	0.78	0.68

TABLE 2
AuC performance for PCR plots on the PF-WILLOW dataset (RP w/ HOG).

Methods	car(S)	car(G)	car(M)	duc(S)	mot(S)	mot(G)	mot(M)	win(w/o C)	win(w/ C)	win(M)	Avg.
LOM	0.61	0.50	0.45	0.50	0.42	0.40	0.35	0.69	0.30	0.47	0.47
Upper bound	0.75	0.69	0.69	0.72	0.70	0.70	0.67	0.80	0.68	0.73	0.71

consistent regions than other proposal methods, and is adequate for region matching. RP shows higher matching precision than other proposals especially when the IoU threshold τ is low. The evaluation results for different features (LOM w/ RP) are shown in Fig. 5(b). The HOG descriptor gives the best performance in matching and retrieval. The CNN features in our comparison come from AlexNet [48] trained for ImageNet classification. Such CNN features have a task-specific bias to capture discriminative parts for classification, which may be less adequate for patch correspondence or retrieval than engineered features such as HOG. Similar conclusions are found in recent papers [34], [70]. See, for example, Table 3 in [70] where SIFT outperforms all AlexNet features (Conv1-5). Among ConvNet features, the fourth and first convolutional layers (Conv4 and Conv1) show the best and worst performance, respectively, while other layers perform similar to SPM. This confirms the finding in [71], which shows that Conv4 gives the best matching performance among ImageNet-trained ConvNet features. The SIAM feature is designed to compute patch similarity, and thus it can be used as a replacement for any task involving SIFT. This type of feature descriptor using Siamese or triplet networks such as [65], [71], [72] works well in finding correspondences between images containing the same object with moderate view point changes, e.g., as in the stereo matching task. But, we can see that this feature descriptor is less adequate for semantic flow, i.e., finding correspondences of different scenes and objects. The main reason is that the training dataset [69] does not feature intra-class variations. We will show that the dense version of our proposal flow also outperforms a learning-based semantic flow method in Section 5.3. Figure 5(c) compares the performance of different matching algorithms (RP w/ HOG), and shows that LOM outperforms others in matching as well as retrieval.

Figure 6 shows the area under curve (AuC) for PCR (top) and mIoU@k (bottom) plots on average for all object classes with all combinations of object proposals, feature descriptors, and matching algorithms. This suggests that combining LOM, RP, and HOG performs best in both metrics. In Table 1, we show AuCs of PCR plots for each class of the PF-PASCAL dataset (RP w/ HOG). We can see that rigid objects (e.g., bus and car) show higher matching precision than deformable ones (e.g., person and bird).

Quantitative comparison on PF-WILLOW. We perform the same experiments with the PF-WILLOW dataset: We can achieve higher matching precision and retrieval accuracy than for the challenging PF-PASCAL dataset. The behavior of PCR, mIoU@k, and AuCs is almost the same as the one for the PF-PASCAL dataset shown in Figs. 5 and 6, so we omit these results. They can be found on our project webpage for completeness. In Table 2, we show AuCs of PCR plots for each sub-class. From this table, we can see that 1) higher matching precision is achieved with objects having a similar pose (e.g., mot(S) vs. mot(M)), 2) performance decreases for deformable object matching (e.g., duck(S) vs. car(S)), and 3) matching precision can increase drastically by eliminating background clutters (e.g., win(w/o C) vs. win(w/ C)), which verifies out motivation of using object proposals for semantic flow.

Effect of the number of proposals. In Fig. 7, we show the AuCs of PCR (left) and mIoU@k (center) plots, on the PF-PASCAL (top) and PF-WILLOW (bottom), as a function of the number of object proposals. We see that 1) upper bounds on matching precision of all proposals are continuously growing, except MCG, as the number of proposal increases, and 2) matching precision and retrieval accuracy of proposal flow are increasing as well, but at a slightly slower rate. On the one hand, as the number of proposals is increasing, the number of inlier proposals, i.e., regions around object bounding boxes $|\mathcal{R}_s|$, is increasing, and thus we can achieve a higher upper bound. On the other hand, the number of outlier proposals, i.e., $|\mathcal{R}|$ - $|\mathcal{R}_s|$, is increasing as well, which prevents us from finding correct matches. Overall, matching precision and retrieval accuracy increase with the number of proposals (except for MCG), and start to saturate around 1000 proposals. We hypothesize that this is related to the fraction of inliers over all proposals, i.e., $|\mathcal{R}_s|/|\mathcal{R}|$, which may decrease in the case of MCG. To verify this, we plot this fraction as a function of the number of object proposals (Fig. 7, right). We can see that the fraction of MCG is drastically decreasing as the number of proposals, which means that MCG generates more and more outlier proposals corresponding, e.g., to background clutter. The reason is that high recall is the main criteria when designing most object proposal methods, but MCG is designed to achieve high precision with a small number of proposals [22].

5.3 Flow field

To compare our method with state-of-the-art semantic flow methods, we compute a dense flow field from our proposal flows (Section 3.3), and evaluate image alignment between all pairs of images in each subset of the PF-PASCAL and PF-WILLOW datasets. We also compare the matching accuracy with existing datasets: Clatech-101 [73], PASCAL parts [11], and Taniai's datasets [13]. In each case, we compare the proposal flow to the state of the art. For proposal flow, we use a SS method and HOG descriptors, unless otherwise specified, and use publicly available codes for all compared methods.

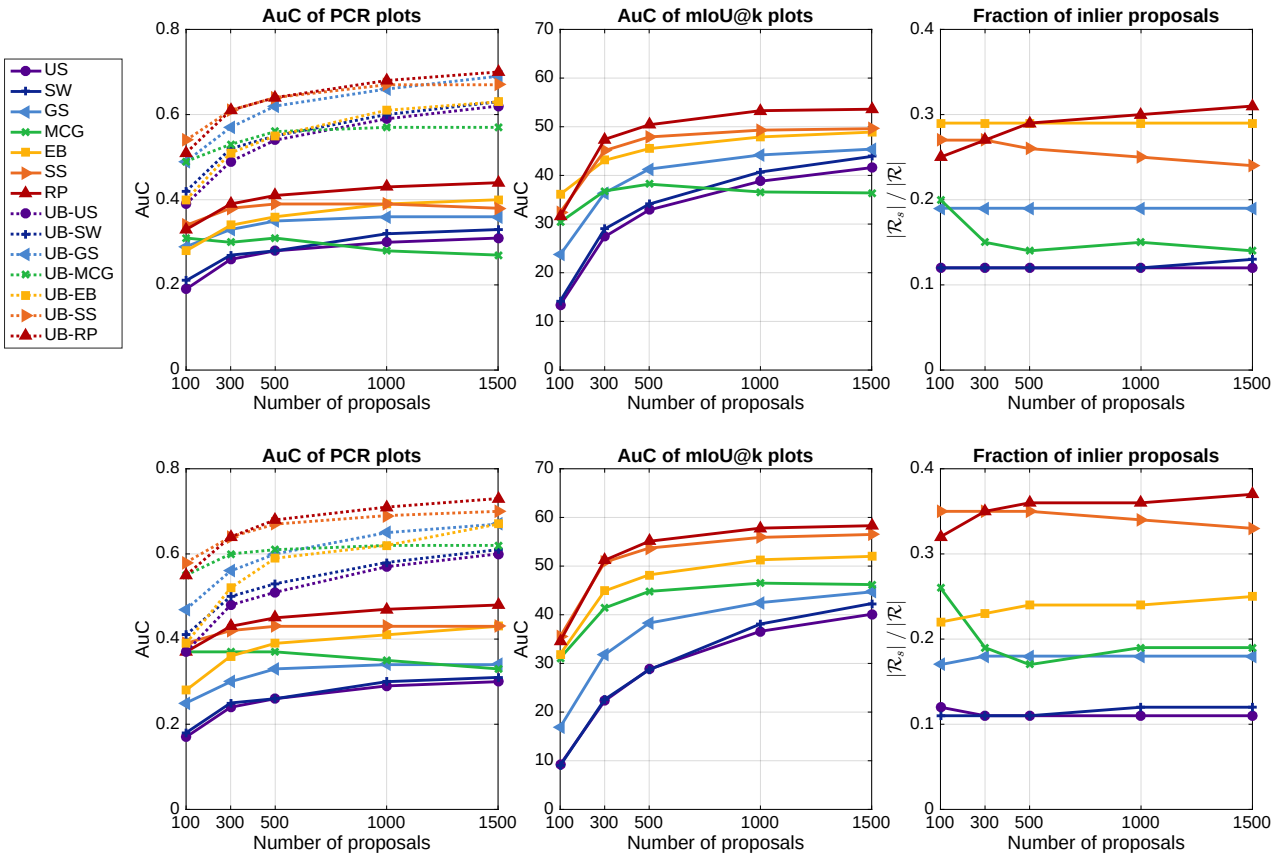


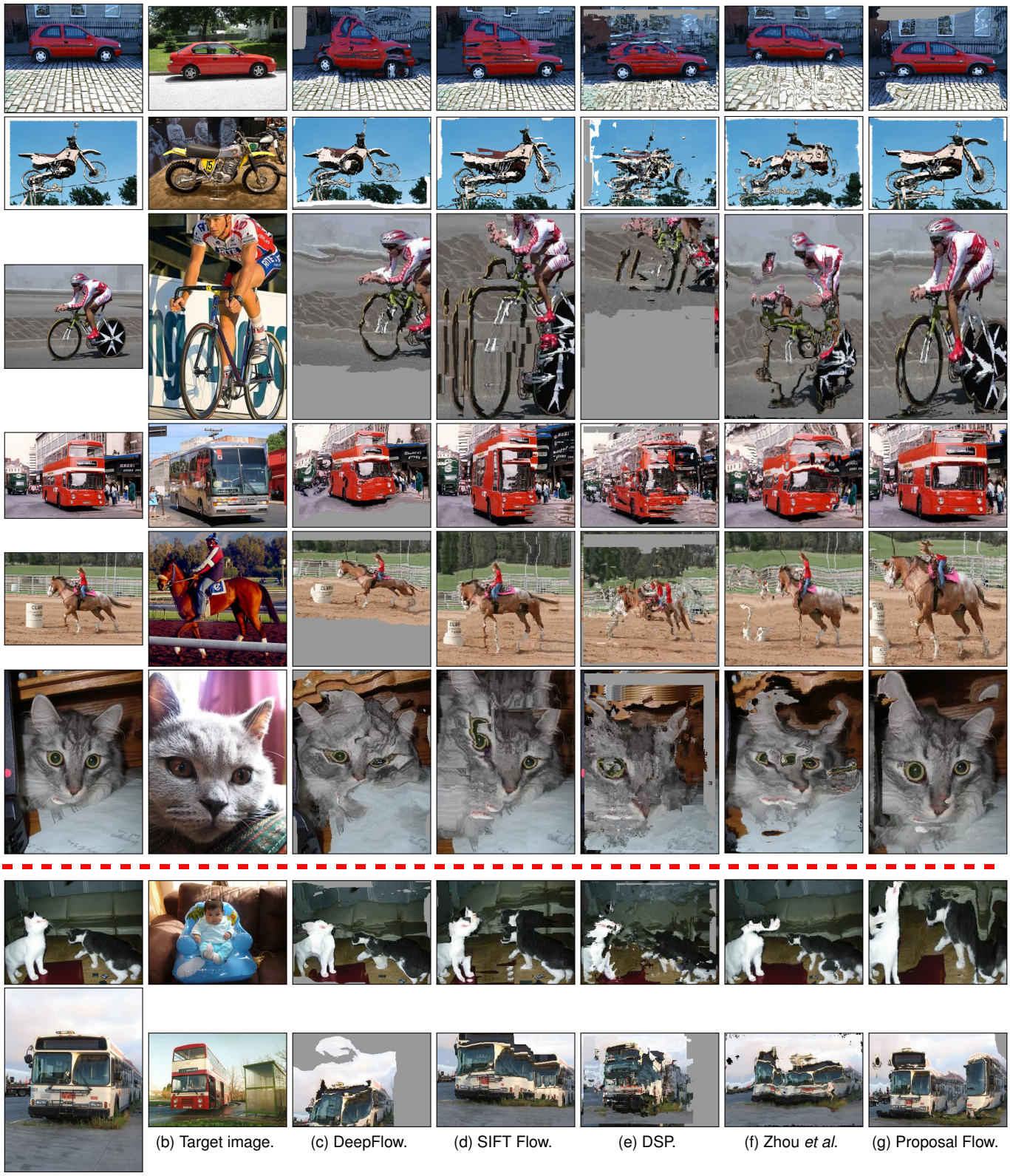
Fig. 7. AuCs for PCR and mIoU@k plots and fraction of inlier proposals over all proposals on the PF-PASCAL (top) and PF-WILLOW (bottom). We can see that matching precision (left, PCR plots) and retrieval accuracy (center, mIoU@k plots) are slightly increasing, except MCG. The MCG is designed to obtain high precision with small number of proposals, so the fraction $|\mathcal{R}_s|/|\mathcal{R}|$ (right) decreases as the number of proposals. The LOM method is used for region matching with the HOG descriptor. **(Best viewed in color.)**

TABLE 3
PCK ($\alpha = 0.1$) comparison for dense flow field on the PF dataset (PF-PASCAL / PF-WILLOW).

Methods	SW [23]	MCG [22]	EB [26]	SS [25]	RP [24]
NAM	0.29/0.44	0.27/0.46	**0.37/0.51**	0.36/0.52	0.37/0.54
PHM	**0.37/0.48**	0.35/0.48	0.35/0.45	0.42/0.55	0.42/0.54
LOM	0.35/0.42	**0.38/ 0.49**	**0.37**/0.45	**0.45/0.56**	**0.44/0.55**
DeepFlow [4]	0.21/0.20				
GMK [12]	0.27/0.27				
SIFT Flow [8]	0.33/0.38				
DSP [10]	0.30/0.37				
Zhou *et al.* [21]	0.30/0.41				

Matching results on PF datasets. We test five object proposal methods (SW, MCG, EB, SS, RP). For an evaluation metric, we use PCK between warped keypoints and ground-truth ones [34], [63]. Ground-truth keypoints are deemed to be correctly predicted if they lie within $\alpha\max(h, w)$ pixels of the predicted points for α in $[0, 1]$, where h and w are the height and width of the object bounding box, respectively. Table 3 shows the average PCK ($\alpha = 0.1$) over all object classes. In our benchmark, all versions of proposal flow significantly outperform SIFT Flow [8], DSP [10], and DeepFlow [4], and proposal flow with PHM and LOM gives better performance than the learning-based method [21]. LOM with SS or RP outperforms other combination of matching and proposal methods, which coincides with the results in Section 5.2. Tables 4 and 5 show the average PCK ($\alpha = 0.1$) over each object class on the PF-PASCAL and PF-WILLOW, respectively. This shows that proposal flow consistently outperforms other methods for all object classes except for table and sheep classes in both datasets. We can also see that the learning-based method [21] does not generalize to other object classes that are not contained in the PASCAL training set (e.g., duc(S)), and are not robust to the outliers (e.g., wine (w/ c)). Figure 8(top) gives a qualitative comparison with the state of the art on the PF-WILLOW and PF-PASCAL datasets. The better alignment found by proposal flow here is clearly visible. Specifically, proposal flow is robust to clutter and translation and scale changes between objects. Figure 8(bottom) shows failure examples of (from top to bottom) sofa and cat classes on the PF-PASCAL dataset, where we see proposal flow does not handle image pairs that contain severe occlusion and objects having similar shape. Our current (unoptimized) `MATLAB` implementation takes on average 8.8 seconds on 2.5 GHz CPU for computing dense flow field using LOM w/ SS and HOG. Table 6 shows runtime comparisons.

Matching results on Caltech-101. We evaluate our approach on the Caltech-101 dataset [73]. Following the experimental protocol in [10], we randomly select 15 pairs of images for each object class, and evaluate matching accuracy with three metrics: Label transfer accuracy (LT-ACC) [74], the IoU metric, and the localization error (LOC-ERR) of corresponding pixel positions. For LT-ACC, we transfer the class label of one image to the other using dense correspondences, and count the number of correctly labeled pixels. Similarly, the IoU score is measured between the transferred label and ground truth. Table 7 compares quantitatively the matching accuracy of proposal flow to the state



(a) Source image. (b) Target image. (c) DeepFlow. (d) SIFT Flow. (e) DSP. (f) Zhou *et al.* (g) Proposal Flow.

Fig. 8. Examples of dense flow field. (a-b) Source images are warped to the target images using the dense correspondences estimated by (c) DeepFlow [4], (d) SIFT Flow [8], (e) DSP [10], (f) Zhou *et al.* [21], and (g) Proposal Flow (LOM w/ RP and HOG). **Top:** Compared to the existing methods, proposal flow is robust to background clutter, and translation and scale changes between objects. The first two images are from the PF-WILLOW and remaining ones are from the PF-PASCAL. **Bottom:** Failure examples of (from top to bottom) sofa and bus classes on the PF-PASCAL dataset. Proposal flow is hard to deal with images containing (from top to bottom) severe occlusion and similarly shaped objects.

TABLE 4
PCK ($\alpha = 0.1$) comparison for dense flow field on the PF-PASCAL dataset (SS w/ HOG).

Methods	aero	bike	bird	boat	bot	bus	car	cat	cha	cow	tab	dog	hor	mbik	pers	plnt	she	sofa	trai	tv	Avg.
LOM	**0.75**	**0.76**	**0.34**	**0.41**	**0.55**	**0.71**	**0.73**	**0.32**	**0.41**	**0.41**	0.21	**0.27**	**0.38**	**0.57**	**0.29**	**0.17**	0.33	**0.34**	**0.54**	**0.46**	**0.45**
DeepFlow [4]	0.55	0.31	0.10	0.19	0.24	0.36	0.31	0.12	0.22	0.10	0.23	0.07	0.11	0.32	0.10	0.08	0.07	0.20	0.31	0.17	0.21
GMK [12]	0.61	0.49	0.15	0.21	0.29	0.47	0.52	0.14	0.23	0.23	0.24	0.09	0.13	0.39	0.12	0.16	0.10	0.22	0.33	0.22	0.27
SIFT Flow [8]	0.61	0.56	0.20	0.34	0.32	0.54	0.56	0.26	0.29	0.21	**0.33**	0.17	0.23	0.43	0.18	0.17	0.17	0.31	0.41	0.34	0.33
DSP [10]	0.64	0.56	0.17	0.27	0.38	0.51	0.55	0.20	0.23	0.24	0.19	0.15	0.23	0.41	0.15	0.11	0.18	0.27	0.35	0.28	0.30
Zhou *et al*. [21]	0.58	0.35	0.15	0.27	0.36	0.40	0.42	0.23	0.26	0.29	0.22	0.20	0.13	0.33	0.16	0.18	**0.48**	0.27	0.34	0.28	0.30

TABLE 5
PCK ($\alpha = 0.1$) comparison for dense flow field on the PF-WILLOW dataset (SS w/ HOG).

Methods	car(S)	car(G)	car(M)	duc(S)	mot(S)	mot(G)	mot(M)	win(w/o C)	win(w/ C)	win(M)	Avg.
LOM	**0.86**	**0.60**	**0.53**	**0.64**	**0.49**	**0.25**	**0.29**	**0.91**	**0.37**	**0.65**	**0.56**
DeepFlow [4]	0.33	0.13	0.22	0.20	0.20	0.08	0.13	0.46	0.08	0.18	0.20
GMK [12]	0.48	0.25	0.34	0.27	0.31	0.12	0.15	0.41	0.17	0.18	0.27
SIFT Flow [8]	0.54	0.37	0.36	0.32	0.41	0.20	0.23	0.83	0.16	0.33	0.38
DSP [10]	0.46	0.30	0.32	0.25	0.31	0.15	0.14	0.85	0.25	0.64	0.37
Zhou *et al*. [21]	0.77	0.34	0.52	0.42	0.34	0.19	0.20	0.78	0.19	0.38	0.41

TABLE 6
Runtime comparison for dense flow field on the PF-PASCAL dataset (SS w/ HOG).

Methods	Time (s)
NAM	4.6 ± 1.0
PHM	5.4 ± 1.1
LOM	8.8 ± 1.3
DeepFlow [4]	$4.7 \pm 0.6^{\dagger}$
GMK [12]	$2.4 \pm 0.3^{\dagger}$
SIFT Flow [8]	$4.2 \pm 0.8^{\dagger}$
DSP [10]	$4.8 \pm 0.8^{\dagger}$

† We used author provided `MEX` implementations.

TABLE 7
Matching accuracy on the Caltech-101 dataset (HOG).

Proposals	Methods	LT-ACC	IoU	LOC-ERR
SW [23]	LOM	**0.78**	0.47	**0.25**
SS [25]	NAM	0.68	0.44	0.41
	PHM	0.74	0.48	0.32
	LOM	**0.78**	**0.50**	**0.25**
RP [24]	NAM	0.70	0.44	0.39
	PHM	0.75	0.48	0.31
	LOM	**0.78**	**0.50**	0.26
DeepFlow [4]		0.74	0.40	0.34
GMK [12]		0.77	0.42	0.34
SIFT Flow [8]		0.75	0.48	0.32
DSP [10]		0.77	0.47	0.35

of the art. It shows that proposal flow using LOM outperforms other approaches, especially for the IoU score and the LOC-ERR of dense correspondences. Note that compared to LT-ACC, these metrics evaluate the matching quality for the foreground object, separate from irrelevant scene clutter. Our results verify that proposal flow focuses on regions containing objects rather than scene clutter and distracting details, enabling robust image matching against outliers.

Matching results on Taniai's Benchmark. We also evaluate flow accuracy on the dataset provided by [13] that consists of 400 image pairs of three groups: FG3DCar (195 image pairs of vehicles from [77]), JODS (81 image pairs of airplanes, horses, and cars from [78]), and PASCAL (124 image pairs of bicycles, motorbikes, buses, cars, trains from [79]). Matching accuracy is measured by the percentage of pixels in the ground-truth foreground region that have an error measure below a certain threshold. To this end, we compute the Euclidean distance between estimated and true flow vectors in a normalized scale where the larger dimensions of images are 100 pixels. Here, we use a threshold of 5 pixels following the work of [13]. We summarize average matching accuracy for each group in the Table 8. The method of [21] uses convolutional neural networks (CNNs) to learn dense correspondence. Since there is no previous dataset available for training the networks for semantic flow, it leverages a 3D model to use the known synthetic-to-synthetic matches as ground truth, allowing cycle consistency to propagate the correct match information from synthetic to real images. The method of [13] leverages an additional cosegmentation to estimate dense correspondence. This is a similar idea to ours in that excluding background regions when estimating correspondences improves the matching accuracy. In the FG3DCar dataset, this method [13] shows better performance then ours. But, overall, our method achieves the best performance on average over all datasets, and even outperforms the learning based method of [21].

TABLE 8
Matching accuracy on the Taniai's dataset (SS w/ HOG).

Methods	FG3DCar	JODS	PASCAL	Avg.
LOM	0.79	**0.65**	**0.53**	**0.66**
DFF [7]	0.50	0.30	0.22	0.31
DSP [10]	0.49	0.47	0.38	0.45
SIFT Flow [8]	0.63	0.51	0.36	0.50
Zhou *et al*. [21]	0.72	0.51	0.44	0.56
Taniai *et al*. [13]	**0.83**	0.60	0.48	0.64

Matching results on PASCAL parts. We use the dataset provided by [11] where the images are sampled from the PASCAL part dataset [80]. Following [11], we first measure part matching accuracy using human-annotated part segments. For this experiment, we measure the weighted IoU score between transferred segments and ground truths, with weights determined by the pixel area of each part (Table 9). To evaluate alignment accuracy, we measure the PCK metric ($\alpha = 0.05$) using keypoint annotations for the 12 rigid PASCAL classes [81] (Table 9). We use the same set of images as in the part matching experiment. Proposal

TABLE 10
PCK performance for a leave-one-out validation on the PF-WILLOW dataset.

Classes	car(S)	car(G)	car(M)	duc(S)	mot(S)	mot(G)	mot(M)	win(w/o C)	win(w/ C)	win(M)	Avg.
PCK	0.95	0.96	0.99	0.93	0.88	0.89	0.91	1.00	1.00	1.00	0.95

TABLE 11
PCK performance for a leave-one-out validation on the PF-PASCAL dataset.

Classes	aero	bike	bird	boat	bot	bus	car	cat	cha	cow	tab	dog	hor	mbik	pers	plnt	she	sofa	trai	tv	Avg.
PCK	0.74	0.89	0.69	0.91	0.92	0.90	0.85	0.83	0.76	0.81	0.73	0.75	0.74	0.75	0.84	0.83	0.73	0.83	0.73	0.86	0.80

TABLE 9
Matching accuracy on the PASCAL parts (SS w/ HOG).

Methods	IoU	PCK
NAM	0.35	0.13
PHM	0.39	**0.17**
LOM	**0.41**	**0.17**
Congealing [75]	0.38	0.11
RASL [76]	0.39	0.16
CollectionFlow [35]	0.38	0.12
DSP [10]	0.39	**0.17**
FlowWeb [11]	**0.43**	**0.26**

flow does better than existing approaches on images that contain clutter (e.g., background, instance-specific texture, occlusion), but in this dataset [11], such elements are confined to only a small portion of the images (See, for example, Fig. 4 in [11]), compared to the PF and the Caltech-101 [73] datasets. This may be a reason that, for the PCK metric, our approach gives similar results to other methods. FlowWeb [11] gives better results than ours, but relies on a cyclic constraint across multiple images (at least, three images[5]). FlowWeb uses the output of DSP [10] as initial correspondences, and refines them with the cyclic constraint. Since our method clearly outperforms DSP, using FlowWeb as a post processing would likely increase performance.

For more examples and qualitative results, see our project webpage: http://www.di.ens.fr/willow/research/proposalflow.

5.4 Quality of generated ground-truth correspondence

Of course, our "ground truth" for the PF datasets is only approximate, since it is obtained by interpolation. We evaluate its quality using a leave-n-out validation: When generating ground-truth dense correspondences using TPS warping as in Section 4.1, we leave out n randomly selected keypoints per each pair (e.g., n among 10 keypoints in the PF-WILLOW dataset), and then evaluate PCK ($\alpha = 0.1$) between the approximated correspondences (using TPS warps) of the leave-out keypoints and their ground-truth annotations. The average PCK of 10 trials over all object classes is shown in Fig. 9. The number in parentheses denotes the number of ground-truth keypoints. For the PF-PASCAL, each image pair has a different number of keypoints. We see that using more keypoint annotations improves the quality of generated ground truth. Note that perfect score would be 1.0. In Tables 10 and 11, we show the PCK results for a leave-*one*-out validation. The average PCK scores are 0.95 and 0.80 on the PF-WILLOW and PF-PASCAL, respectively. These numbers are quite reasonable, and validate the use of our ground-truth data using TPS.

5. FlowWeb [11] uses 100 images to find correspondences for one pair of images. That is, a single output of DSP [10] is refined using 9900 pairs of matches.

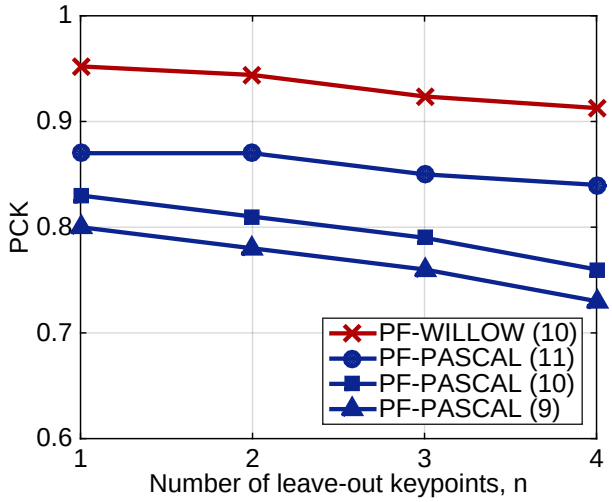


Fig. 9. Verification of ground-truth data using a leave-n-out validation. This shows the average PCK of 10 trials over all object classes. For this experiment, we leave out n randomly selected keypoints per each pair, and then measure PCK scores between the estimated correspondences (using TPS warps) of the leave out keypoints and their ground-truth annotations. **(Best viewed in color.)**

5.5 Object proposals vs. sliding windows

Our experiments show that proposal flow outperforms state-of-the-art methods such as SIFT flow [8], DSP [10], and DeepFlow [5]: Note that these methods all employ a sort of sliding window strategies for matching (i.e., regular sampling with a fixed stride, and in particular, DeepFlow [5] with stride 1). Figures 5(a) and 6 evaluate SW within our approach, where we make proposals by placing windows on a regular grid across predefined 5 scales and 5 aspect ratios with a uniform stride (following [23]). The PCR and mIoU@k plots show that object proposals clearly outperform SW with the same number of regions. In Table 7, we can see that 1) the proposal flow method with SW already outperforms competing algorithms, and 2) it further benefits from the use of SS to go from 0.47 to 0.50 in terms of the IoU metric. Note that this metric focuses on the foreground matching quality [10], implying that the use of object proposals helps in matching foreground regions. The advantage can be clearly seen with more cluttered images. For example, LOM with SW and SS on the PF-WILLOW gives PCK ($\alpha = 0.1$) of 0.42 and 0.56, respectively, as shown in Table 3. The superior performance comes from the effective use of geometric contextual information as well as that of object proposals.

6 Discussion

We have presented a robust region-based semantic flow method, called proposal flow, and shown that it can effectively be mapped to pixel-wise dense correspondences. We have also introduced

the PF datasets for semantic flow, and shown that they provide a reasonable benchmark for a semantic flow evaluation without extremely expensive manual annotation of full ground truth. Our benchmarks can be used to evaluate region-based semantic flow methods and also pixel-based ones, and experiments with the PF datasets demonstrate that proposal flow substantially outperforms existing semantic flow methods. Experiments with Caltech-101, the PASCAL parts, and Taniai's datasets further validate these results.

Proposal flow has benefited from the use of learning-based descriptors for semantic correspondences [82], or learning geometric matching [83]. Although these approaches boost the performance of proposal flow, they still use hand-crafted object proposals. In future work, we will explore models and architectures to learn regions to match.

ACKNOWLEDGMENTS

This work was supported in part by ERC grants VideoWorld and Allegro, and the Institut Universitaire de France. The work of B. Ham was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2017R1C1B2005584). Part of this work was done while B. Ham and M. Cho were with Inria, Paris.

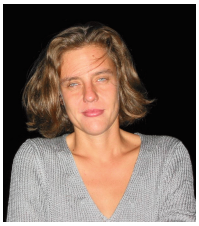
REFERENCES

[1] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *TPAMI*, vol. 15, no. 4, pp. 353–363, 1993.
[2] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *CVPR*, 2011.
[3] B. K. Horn and B. G. Schunck, "Determining optical flow: A retrospective," *Artificial Intelligence*, vol. 59, no. 1, pp. 81–87, 1993.
[4] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "DeepMatching: Hierarchical deformable dense matching," *IJCV*, pp. 1–24, 2015.
[5] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *ICCV*, 2013.
[6] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.
[7] H. Yang, W.-Y. Lin, and J. Lu, "Daisy filter flow: A generalized discrete approach to dense correspondences," in *CVPR*, 2014.
[8] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *TPAMI*, vol. 33, no. 5, pp. 978–994, 2011.
[9] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," *ACM TOG*, vol. 30, no. 4, p. 70, 2011.
[10] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," in *CVPR*, 2013.
[11] T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros, "FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences," in *CVPR*, 2015.
[12] O. Duchenne, A. Joulin, and J. Ponce, "A graph-matching kernel for object categorization," in *ICCV*, 2011.
[13] T. Taniai, S. N. Sinha, and Y. Sato, "Joint recovery of dense correspondence and cosegmentation in two images," in *CVPR*, 2016.
[14] H. Bristow, J. Valmadre, and S. Lucey, "Dense semantic correspondence where every pixel is a classifier," in *ICCV*, 2015.
[15] T. Hassner, V. Mayzels, and L. Zelnik-Manor, "On SIFTs and their scales," in *CVPR*, 2012.
[16] J. Hur, H. Lim, C. Park, and S. C. Ahn, "Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation," in *CVPR*, 2015.
[17] W. Qiu, X. Wang, X. Bai, Z. Tu *et al.*, "Scale-space SIFT flow," in *WACV*, 2014.
[18] M. Tau and T. Hassner, "Dense correspondences across scenes and scales," *TPAMI*, vol. 38, no. 5, pp. 875–888, 2016.
[19] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer, "Dense segmentation-aware descriptors," in *CVPR*, 2013.
[20] C. B. Choy, M. Chandraker, J. Gwak, and S. Savarese, "Universal correspondence network," in *NIPS*, 2016.
[21] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros, "Learning dense correspondence via 3D-guided cycle consistency," in *CVPR*, 2016.
[22] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *CVPR*, 2014.
[23] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *TPAMI*, 2015.
[24] S. Manen, M. Guillaumin, and L. Van Gool, "Prime object proposals with randomized Prim's algorithm," in *ICCV*, 2013.
[25] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, no. 2, pp. 154–171, 2013.
[26] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014.
[27] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.
[28] H. Kaiming, Z. Xiangyu, R. Shaoqing, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*, 2014.
[29] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *CVPR*, 2016.
[30] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow," in *CVPR*, 2016.
[31] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3D human pose annotations," in *ICCV*, 2009.
[32] D. A. Forsyth and J. Ponce, "Computer vision: A modern approach (2nd edition)," *Computer Vision: A Modern Approach*, 2011.
[33] M. Cho and K. M. Lee, "Progressive graph matching: Making a move of graphs via probabilistic voting," in *CVPR*, 2012.
[34] J. L. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?" in *NIPS*, 2014.
[35] I. Kemelmacher-Shlizerman and S. M. Seitz, "Collection flow," in *CVPR*, 2012.
[36] X. Zhou, M. Zhu, and K. Daniilidis, "Multi-image matching via fast alternating minimization," in *ICCV*, 2015.
[37] J. Carreira, A. Kar, S. Tulsiani, and J. Malik, "Virtual view networks for object reconstruction," in *CVPR*, 2015.
[38] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with r*CNN," in *ICCV*, 2015.
[39] R. G. Cinbis, J. Verbeek, and C. Schmid, "Multi-fold MIL training for weakly supervised object localization," in *CVPR*, 2014.
[40] J. Dai, K. He, and J. Sun, "BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *ICCV*, 2015.
[41] M. Cho, S. Kwak, C. Schmid, and J. Ponce, "Unsupervised object discovery and localization in the wild: Part-based matching using bottom-up region proposals," in *CVPR*, 2015.
[42] H. Jiang, "Matching bags of regions in RGBD images," in *CVPR*, 2015.
[43] M. Bai, W. Luo, K. Kundu, and R. Urtasun, "Exploiting semantic information and deep matching for optical flow," in *ECCV*, 2016.
[44] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black, "Optical flow with semantic segmentation and localized layers," in *CVPR*, 2016.
[45] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.
[46] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.
[47] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.
[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
[49] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *IJCV*, vol. 77, no. 1-3, pp. 259–289, 2008.
[50] S. Maji and J. Malik, "Object detection using a max-margin hough transform," in *CVPR*, 2009.
[51] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *CVPR*, 2008.
[52] H. P. Lopuhaa and P. J. Rousseeuw, "Breakdown points of affine equivariant estimators of multivariate location and covariance matrices," *The Annals of Statistics*, pp. 229–248, 1991.
[53] P. T. Fletcher, S. Venkatasubramanian, and S. Joshi, "Robust statistics on riemannian manifolds via the geometric median," in *CVPR*, 2008.
[54] R. Chandrasekaran and A. Tamir, "Open questions concerning weiszfeld's algorithm for the fermat-weber location problem," *Mathematical Programming*, vol. 44, no. 1-3, pp. 293–295, 1989.
[55] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE TPAMI*, no. 6, pp. 567–585, 1989.

[56] G. Donato and S. Belongie, "Approximate thin plate spline mappings," in *ECCV*, 2002.
[57] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," in *CVPR*, 2015.
[58] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *IJCV*, vol. 92, no. 1, pp. 1–31, 2011.
[59] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *ECCV*, 2012.
[60] R. Sibson *et al.*, "A brief description of natural neighbour interpolation," *Interpreting multivariate data*, vol. 21, pp. 21–36, 1981.
[61] A. Kanazawa, D. W. Jacobs, and M. Chandraker, "WarpNet: Weakly supervised matching for single-view reconstruction," in *CVPR*, 2016.
[62] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The CALTECH-UCSD birds-200-2011 dataset," 2011.
[63] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE TPAMI*, vol. 35, no. 12, pp. 2878–2890, 2013.
[64] M. Cho, K. Alahari, and J. Ponce, "Learning graphs to match," in *ICCV*, 2013.
[65] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *ICCV*, 2015.
[66] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[67] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei, "Co-localization in real-world images," in *CVPR*, 2014.
[68] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *CVPR*, 2009.
[69] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *TPAMI*, vol. 33, no. 1, pp. 43–57, 2011.
[70] M. Paulin *et al.*, "Local convolutional features with unsupervised training for image retrieval," in *ICCV*, 2015.
[71] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *CVPR*, 2015.
[72] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *CVPR*, 2015.
[73] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *TPAMI*, vol. 28, no. 4, pp. 594–611, 2006.
[74] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *TPAMI*, vol. 33, no. 12, pp. 2368–2382, 2011.
[75] E. G. Learned-Miller, "Data driven image models through continuous joint alignment," *TPAMI*, vol. 28, no. 2, pp. 236–250, 2006.
[76] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *TPAMI*, vol. 34, no. 11, pp. 2233–2246, 2012.
[77] Y.-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis, "Jointly optimizing 3D model fitting and fine-grained classification," in *ECCV*, 2014.
[78] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *CVPR*, 2013.
[79] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *ICCV*, 2011.
[80] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun *et al.*, "Detect what you can: Detecting and representing objects using holistic models and body parts," in *CVPR*, 2014.
[81] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3D object detection in the wild," in *WACV*, 2014.
[82] S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn, "FCSS: Fully convolutional self-similarity for dense semantic correspondence," in *CVPR*, 2017.
[83] K. Han, R. S. Rezende, B. Ham, K.-Y. K. Wong, M. Cho, C. Schmid, and J. Ponce, "SCNet: Learning semantic correspondence," *CoRR*, vol. abs/1705.04043, 2017.

Bumsub Ham is an an Assistant Professor of Electrical and Electronic Engineering at Yonsei University in Seoul, Korea. He received the B.S. and Ph.D. degrees in Electrical and Electronic Engineering from Yonsei University in 2008 and 2013, respectively. From 2014 to 2016, he was Post-Doctoral Research Fellow with Willow Team of INRIA Rocquencourt, École Normale Supérieure de Paris, and Centre National de la Recherche Scientifique. His research interests include computer vision, computational photography, and machine learning, in particular, regularization and matching, both in theory and applications.

Minsu Cho is an Assistant Professor of computer science and engineering at POSTECH in Pohang, South Korea. He obtained his PhD degree in Electrical Engineering and Computer Science from Seoul National University in 2012. Before joining POSTECH in 2016, he worked as an Inria starting researcher in the ENS/Inria/CNRS Project team WILLOW at cole Normale Superiure, Paris, France. His research lies in the areas of computer vision and machine learning, especially in the problems of object discovery, weakly-supervised learning, and graph matching.



Cordelia Schmid holds a M.S. degree in Computer Science from the University of Karlsruhe and a Doctorate, also in Computer Science, from the Institut National Polytechnique de Grenoble (INPG). Her doctoral thesis received the best thesis award from INPG in 1996. Dr. Schmid was a post-doctoral research assistant in the Robotics Research Group of Oxford University in 1996–1997. Since 1997 she has held a permanent research position at INRIA Grenoble Rhone-Alpes, where she is a research director and directs an INRIA team. Dr. Schmid is the author of over a hundred technical publications. She has been an Associate Editor for IEEE PAMI (2001–2005) and for IJCV (2004–2012), editor-in-chief for IJCV (2013—), a program chair of IEEE CVPR 2005 and ECCV 2012 as well as a general chair of IEEE CVPR 2015 and ECCV 2020. In 2006, 2014 and 2016, she was awarded the Longuet-Higgins prize for fundamental contributions in computer vision that have withstood the test of time. She is a fellow of IEEE. She was awarded an ERC advanced grant in 2013, the Humbolt research award in 2015 and the Inria & French Academy of science Grand Prix in 2016.

Jean Ponce is a Professor at École Normale Supérieure (ENS) and PSL Research University in Paris, France, where he leads a joint ENS/INRIA/CNRS research team, WILLOW, that focuses on computer vision and machine learning. Prior to this, he served for over 15 years on the faculty of the Department of Computer Science and the Beckman Institute at the University of Illinois at Urbana-Champaign. Dr. Ponce is the author of over 150 technical publications, including the textbook "Computer Vision: A Modern Approach", in collaboration with David Forsyth. He is a member of the Editorial Boards of Foundations and Trends in Computer Graphics and Vision, the International Journal of Computer Vision, and the SIAM Journal on Imaging Sciences. He was also editor-in-chief of the International Journal on Computer Vision (2003-2008), an Associate Editor of the IEEE Transactions on Robotics and Automation (1996-2001), and an Area Editor of Computer Vision and Image Understanding (1994-2000). Dr. Ponce was Program Chair of the 1997 IEEE Conference on Computer Vision and Pattern Recognition and served as General Chair of the year 2000 edition of this conference. In 2003, he was named an IEEE Fellow for his contributions to Computer Vision, and he received a US patent for the development of a robotic parts feeder. In 2008, he served as General Chair for the European Conference on Computer Vision.