Robust Image Alignment with Multiple Feature Descriptors and Matching-Guided Neighborhoods

Kuang-Jui Hsu[1,2] Yen-Yu Lin[1] Yung-Yu Chuang[2]
[1]Academia Sinica, Taiwan [2]National Taiwan University, Taiwan

Abstract

This paper addresses two issues hindering the advances in accurate image alignment. First, the performance of descriptor-based approaches to image alignment relies on the chosen descriptor, but the optimal descriptor typically varies from image to image, or even pixel to pixel. Second, the neighborhood structure for smoothness enforcement is usually predefined before alignment. However, object boundaries are often better discovered during alignment. The proposed approach tackles the two issues by adaptive descriptor selection and dynamic neighborhood construction. Specifically, we associate each pixel to be aligned with an affine transformation, and integrate the learning of the pixel-specific transformations into image alignment. The transformations serve as the common domain for descriptor fusion, since the local consensus of each descriptor can be estimated by accessing the corresponding affine transformation. It allows us to pick the most plausible descriptor for aligning each pixel. On the other hand, more object-aware neighborhoods can be produced by referencing the consistency between the learned affine transformations of neighboring pixels. The promising results on popular image alignment benchmarks manifests the effectiveness of our approach.

1. Introduction

Image alignment aims to densely identify pixel correspondences across images. It is an active and fundamental topic in computer vision, because it is essential to a broad set of applications, such as scene parsing [23], common object discovery [30], image denoising [3], image enhancement [12] and depth estimation [17]. The major challenge that image alignment techniques must face is the large photometric and geometric variations between images to be aligned. Such unfavorable variations significantly degrade the performance of conventional optical flow approaches in producing the correspondence field for image alignment.

To address this problem, *SIFT flow* [24], a pioneering descriptor-based method for image alignment, adopts the model of optical flow, and generates the flow field by matching the SIFT features [25] instead of raw pixel features. Despite the effectiveness, there are still two main limitations in most descriptor-based methods. First, existing descriptors are designed with the trade-off between *distinctiveness* and *invariance*. The effectiveness of a descriptor for alignment crucially depends on the types and the degrees of variations between the images to be aligned. In other words, different features could work better for different variations. Most descriptor-based methods adopt a specific descriptor, and does not take this issue into account. Second, the neighborhood used to enforce the smoothness of the flow field are often predefined by using the four-neighbor rule, the bilateral filter [35], and so on. The constructed neighborhood carries only the intra-image information, and neglects the object-background configuration, which can be better revealed during the process of alignment.

To address the two limitations, we present an approach that leverages *multiple complementary descriptors* and *matching-guided neighborhoods*, and can produce flow maps of high quality. Specifically, we associate each pixel with a learnable affine matrix, which specifies the flow field within that pixel's neighborhood. On the one hand, the affine matrix serves as a common domain for descriptor fusion, because the local consensus of a descriptor can be measured by the gap between the flows estimated by the affine matrix and that descriptor. It allows us to pick the most plausible descriptor in a *pixel-specific* manner. On the other hand, motivated by the observation that correct correspondences within the same object often undergo coherent transformations, we then determine the neighborhood of a pixel by measuring its consistency with its neighbors in terms of affine matrices. The yielded neighborhood is more object-aware, and hence facilitates image alignment.

As an illustration, Figure 1 shows the alignment results on two cases, `face` and `Leuven`, by using SIFT flow, its variants (replacing the SIFT descriptor with the GB descriptor [5] and the LIOP descriptor [38], respectively), and our approach. The strong coherence in shape presents in the case of `face`, so the shape-based descriptor, geometric blur, gives good results. For the case of `Leuven`, better

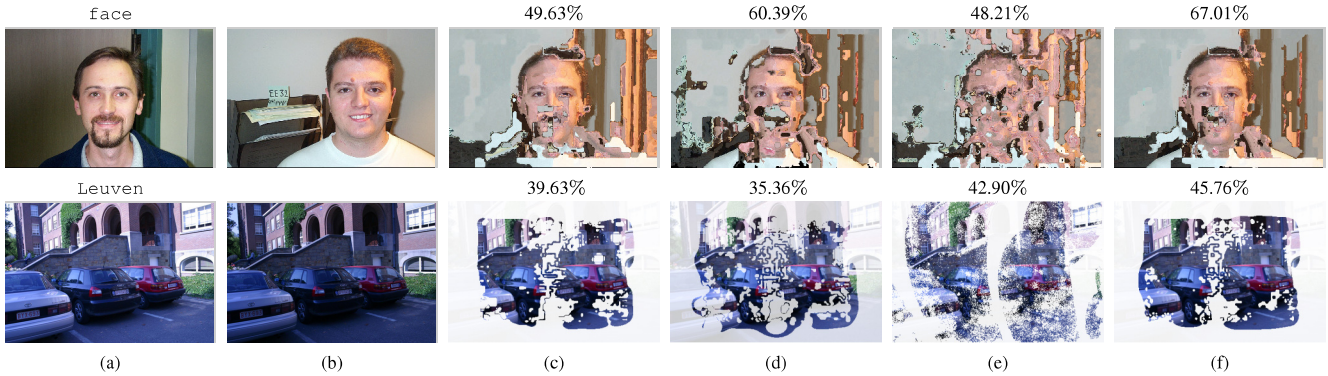

Figure 1. Image alignment on two image pairs, face and Leuven. (a) & (b) Images to be aligned. (c) ∼ (f) Alignment results using different approaches, including (c) SIFT flow, (d) the variant of SIFT flow by replacing the descriptor with GB [5], (e) the variant of SIFT flow by replacing the descriptor with LIOP [38], and (f) our approach. On the top of each result, we show its score (intersection-over-union for face and the correct ratio for Leuven). Note that, for Leuven, only correct matches are shown.

performance is achieved by using LIOP descriptor owing to its robustness to the change of lighting conditions. Our approach can make the most of multiple descriptors with the matching-guided neighborhoods, and results in visually and quantitatively better flow fields in both cases.

In this paper, we integrate the learning of the pixel-specific affine matrix and object-aware neighborhood into the process of image alignment, and cast them as a joint optimization problem. Through the iterative optimization, the object-aware neighborhoods are gradually revealed by the learned affine matrices, while the alignment results would be progressively improved owing to the better neighborhoods. Our approach is comprehensively evaluated on four benchmarks of image alignment, including VGG dataset [28], MSRC dataset [30, 32], Caltech dataset [18] and LMO dataset [18, 23] for different tasks. The results show that our approach effectively produces object-aware neighborhoods and selects a proper descriptor for each pixel, thus leading to a remarkable performance boost.

2. Related work

We review previous work relevant to the development of our approach by categories in this section.

Local feature descriptors. The design of local feature descriptors [28] has gained significant progress. Various descriptors have been developed to be robust to noises as well as invariant to variations in image correspondence. For example, *SIFT (scale-invariant feature transform)* [25] characterizes image regions in the gradient domain, and is invariant to scale and orientation changes. *LIOP (local intensity order pattern)* [38] encodes ordinal information, and is robust to the changes of lighting conditions. *DAISY* [34] is featured with fast feature extraction, while keeping invariant to viewpoint changes. *GB (geometric blur)* [5] matches shapes with deformation by using spatially varying kernels. In addition, diverse visual cues have been explored in descriptor construction, such as color characteristics [1, 37], shapes [4], self-similarities [10, 31], and local symmetries [14]. These descriptors are designed with the trade-off between distinctiveness and invariance. Thus, the optimal descriptor for alignment varies from pixel to pixel. Our approach addresses this issue by taking multiple complementary descriptors into account.

Dense image correspondence. In contrast to sparse matching [8, 9, 14], methods of this class aim to identify the matched regions between pairs of images in a dense manner. The major challenges are the large photometric and geometric variations between images. To address this issue, *SIFT flow* [24] adopts the computational model of optical flow, but discovers the correspondences by matching the SIFT features instead of raw features. HaCohen *et al.* [12] proposed to fit a color model and aggregate consistent matching regions with locally adaptive constraints. Kim *et al.* [18] developed the *deformable spatial pyramid* model which further improves SIFT flow in both efficiency and accuracy. Barnes *et al.* [2] proposed a fast but approximate method, *PatchMatch*, for finding the nearest patches across images. It was further generalized to search patches with various scales and rotation angles [3]. Yang *et al.* [41] presented *DAISY filter flow*, which combines the DAISY descriptor [34], filter-based flow inference [16], and PatchMatch [2], to efficiently estimate the correspondence field with significant variations. To be more robust to scales, Hassner *et al.* [13] used a set of SIFTs with multiple scales, and Kokkinos and Yuille [20] adopted multiple scale filters in construing features. These two methods are robust to the scales, but less robust to other intra-object variations. All these methods adopt a single descriptor. They may have sub-optimal performance, when the adopted descriptor cannot handle the variations between the image pair.

Feature fusion. Since different descriptors characterize diverse visual cues, using multiple descriptors has been a feasible way for improving the correspondence performance. Research results [7, 33, 39] have shown that integrating sparse feature matching into dense image correspondence yields better performance, especially when large displacement presents. Fixed weights are used to merge the evidences extracted on sparsely detected interest points and densely sampled points in these methods, neglecting the fact the optimal features are often image-dependent. Concerning this issue, adaptive feature fusion has been implemented in recent studies [19, 21, 40]. Lempitsky *et al.* [21] proposed *FusionFlow*, which compiles the flow map by adaptively fusing multiple flow proposals obtained by different flow estimation methods with various parameter settings. Xu *et al.* [40] presented a selective model, in which the color and gradient constraints are adaptively combined to deal with outliers. Kim *et al.* [19] developed a locally varying data term, in which multiple types of data models are merged to best reduce local ambiguity. However, features extracted by diverse descriptors or models are usually of different dimensions and with different scales of statistics. Thus, fusion by directly combining the respective features or comparing distances may be difficult. We overcome this issue by using the pixel-specific affine matrices as the common domain for descriptor fusion. The local consistency across heterogeneous descriptors can be measured by their compatibility with the corresponding affine matrix. This property allows us to select an appropriate descriptor for aligning each pixel.

Neighborhood construction. The smoothness term implements spatial regularization, and plays a key role in dense image correspondence. It is usually computed based on the neighborhoods of pixels. There exist a number of approaches for neighborhood construction, such as the four- or eight-connected rule, Gaussian filter [29], bilateral filter [35], guided filter [16]. In contrast to the *local* approaches [16, 29, 35], Yang [42] proposed a non-local method for cost aggregation in matching, and showed its robustness in textureless regions. In addition, research effort [15, 26, 43] has been made on the construction of adaptive supports. However, the foregoing methods compile the neighborhoods by accessing the single image, and suffer from the inter-object ambiguity and intra-object variations. The generated neighborhoods may be inconsistent with the true object boundaries. Our approach instead integrates neighborhood construction into image alignment. More object-aware neighborhoods are produced by leveraging the object-background configuration revealed during image alignment. Trulls *et al.* [36] added the object-aware information into the feature descriptors for image alignment. Lin *et al.* [22] integrated the bilateral functions into sparse matching to enforce the global flow coherence and handle noisy cases. Unlike aforementioned approaches, ours dynamically estimates local neighborhoods by referring to flow fields during alignment.

3. The proposed approach

We introduce the problem definition, the formulation, the optimization process and implementation details of our approach in this section.

3.1. Problem definition

We aim to align two given images I_1 and I_2 by finding a plausible flow map $W = \{\mathbf{w}_i\}_{i=1}^N$, where $\mathbf{w}_i = [u_i \ v_i \ 0]^\top$ is the flow vector at the ith pixel, $\mathbf{p}_i = [x_i \ y_i \ 1]^\top$, and N is the number of pixels in I_1. Multiple descriptors are applied to better characterize each pixel in I_1 and I_2. A set of flow map proposals can then be generated by using any descriptor-based algorithm for image alignment, such as [3, 18, 24, 41]. We in this work use SIFT flow [24] for its stable and good performance. Suppose M descriptors are used. The generated flow proposals would be $\{W^m\}_{m=1}^M$, where $W^m = \{\mathbf{w}_i^m\}_{i=1}^N$ is the flow map produced by using SIFT flow with the substituted descriptor m. Our method yields the flow map W by referencing only the flow proposals $\{W^m\}_{m=1}^M$. Thus, it can conveniently work with heterogeneous descriptors without worrying about their diversities. Specifically, every pixel i can choose its flow vector from one of the proposals, *i.e.*, $\mathbf{w}_i \leftarrow \mathbf{w}_i^{\ell_i}$, where $\ell_i \in \{1, 2, ..., M\}$. Hence, it is formulated as a labelling problem by optimizing $L = \{\ell_i\}_{i=1}^N$ for producing the flow map $W = \{\mathbf{w}_i\}_{i=1}^N$.

As mentioned previously, we associate each pixel i with a learnable affine matrix $A_i \in \mathbb{R}^{3\times3}$, which specifies the flow within the neighborhood of that pixel. Let $\mathcal{N}_i$ denote the index set of the spatial neighbors of pixel i. A weight vector $\mathbf{e}_i = [e_{ij} | j \in \mathcal{N}_i]$ is maintained to define the neighborhood of pixel i. To sum up, our approach aligns images I_1 and I_2 by optimizing three variable sets: $L = \{\ell_i\}_{i=1}^N$, $A = \{A_i\}_{i=1}^N$, and $E = \{\mathbf{e}_i\}_{i=1}^N$.

3.2. Alignment objective

We incorporate the learning process of the pixel-specific affine matrix and neighborhood into image alignment, and cast it as the following constrained optimization problem:

$$\min_{E,A,L} \quad \sum_{i=1}^{N} J(\ell_i, A_i, \mathbf{e}_i) \tag{1}$$

$$\text{s.t.} \quad \mathbf{e}_i \succeq 0, \ \mathbf{e}_i^\top \mathbf{1} = 1, \ \text{for } i = 1, 2, ..., N, \tag{2}$$

where $\mathbf{1}$ is a column vector whose elements are one. The constraints in Eq. (2) ensure the non-negative and normalized neighborhood for each pixel i. $J(\mathbf{e}_i, A_i, \ell_i)$ is the en-

ergy function regarding pixel i, and is defined below:

$$J(\ell_i, A_i, \mathbf{e}_i) = \gamma\|\mathbf{p}'_i - A_i\mathbf{p}_i\|^2 + \sum_{j\in\mathcal{N}_i} e_{ij}\|\mathbf{p}'_j - A_i\mathbf{p}_j\|^2 + \alpha\sum_{j\in\mathcal{N}_i}(e_{ij}-s_{ij})^2 + \beta\sum_{j\in\mathcal{N}_i} e_{ij}[\ell_i \neq \ell_j], \quad (3)$$

where α, β, and γ are three non-negative constants. Their values are fixed for images in each used benchmark in the experiments. $\mathbf{p}'_i$ is the corresponding point of $\mathbf{p}_i$, *i.e.*, $\mathbf{p}'_i = \mathbf{p}_i + \mathbf{w}_i^{\ell_i}$. s_{ij} is a normalized similarity between A_i and A_j, and its value is set as

$$s_{ij} \leftarrow \frac{\exp(-d_{geo}(m_i, m_j)/\sigma)}{\sum_{j\in\mathcal{N}_i}\exp(-d_{geo}(m_i, m_j)/\sigma)}. \quad (4)$$

The precise definition of s_{ij} will be given later.

Three optimization variables present in the objective function pertaining to pixel i, including its proposal selector ℓ_i, affine matrix A_i, and neighborhood weight vector $\mathbf{e}_i$. Four terms in Eq. (3) are employed to enforce the compatibility among the three variables. Specifically, the first term measures the inconsistence between ℓ_i and A_i, the third term measures that between A_i and $\mathbf{e}_i$, while the fourth term measures that between ℓ_i and $\mathbf{e}_i$. The second term evaluates the joint inconsistence among the three variables. The justification for the four terms is given as follows.

The first term $\|\mathbf{p}'_i - A_i\mathbf{p}_i\|^2$ in Eq. (3) reveals the consensus between proposal selector ℓ_i and affine matrix A_i. While the former gives the correspondence of pixel i by $\mathbf{p}'_i = \mathbf{p}_i + \mathbf{w}_i^{\ell_i}$, the latter specifies the local flow field centered on pixel i, *e.g.*, $A_i\mathbf{p}_i$ at pixel i. The smaller the first term, the more consistent the two variables are.

The second term $\sum_{j\in\mathcal{N}_i} e_{ij}\|\mathbf{p}'_j - A_i\mathbf{p}_j\|^2$ in Eq. (3) can be regarded as a generalization of the first term. It considers the compatibility between affine matrix A_i and proposal selectors $\{\ell_j\}_{j\in\mathcal{N}_i}$ within the ith pixel's neighborhood, which is parametrized by $\mathbf{e}_i$.

The third term $\sum_{j\in\mathcal{N}_i}(e_{ij}-s_{ij})^2$ in Eq. (3) evaluates the consistency between the neighborhood and the affine transformation of pixel i. The design of this term is inspired by the observation that nearby pixels within the same object tend to undergo similar transformations in correspondence. We leverage this property to derive object-aware neighborhood. Specifically, e_{ij} indicates the contribution of pixel j to the neighborhood of pixel i. On the other hand, s_{ij} measures the normalized similarity between affine matrices A_i and A_j. In Eq. (4), $m_i = (\mathbf{p}_i, A_i\mathbf{p}_i, A_i)$ is the specified correspondence of pixel i by A_i, and m_j is similarly defined. We adopt the *reprojection error* [9], d_{geo}, to measure the distance between two affine matrices. σ is set as the average distance from A_i to $\{A_j\}_{j\in\mathcal{N}_i}$. The reprojection error is defined as

$$d_{geo}(m_i, m_j) = \frac{1}{2}(d_{geo}(m_i|m_j) + d_{geo}(m_j|m_i)), \quad (5)$$

where

$$d_{geo}(m_i|m_j) = \frac{1}{2}(\|A_i\mathbf{p}_i - A_j\mathbf{p}_i\| + \|\mathbf{p}_i - A_j^{-1}A_i\mathbf{p}_i\|), \quad (6)$$

and $d_{geo}(m_j|m_i)$ is symmetrically defined. It can be observed that $\mathbf{s}_i = [s_{ij}|j \in \mathcal{N}_i]$ is a distribution. Thus, optimizing this term leads to the coherence between $\mathbf{s}_i$ and $\mathbf{e}_i$.

The last term $\sum_{j\in\mathcal{N}_i} e_{ij}[\ell_i \neq \ell_j]$ in Eq. (3) preserves the consistence between the proposal selectors of neighboring pixels. Based on the fact that the characteristics of pixels are spatially correlative, the optimal descriptors for feature extraction tend to be consistent along the spatial domain. This energy function encourages the spatial smoothness of the adopted descriptors, since each flow proposal corresponds to a specific descriptor in this work.

It is worth pointing out that our formulation in Eq. (1) can collaborate with multiple heterogeneous descriptors without accessing their respective feature vectors and distances. This property prevents our approach from suffering from the diversities among these descriptors, such as dimensions, statistics scales, similarity measures. It also distinguishes this work from previous approaches relevant to descriptor fusion, such as [19, 21, 40].

3.3. Optimization procedure

Since direct optimization to Eq. (1) is difficult, we instead adopt an iterative, alternating strategy to optimize L, A, and E. At each iteration, one of the three variables is optimized while keeping the others fixed, and then their roles are switched sequentially. Iterations are repeated until convergence or a maximum number of iterations is reached.

On optimizing L. While fixing A and E, the optimization problem in Eq. (1) is reduced to

$$\min_L \sum_{i=1}^{N}\Big(\gamma\|\mathbf{p}_i + \mathbf{w}_i^{\ell_i} - A_i\mathbf{p}_i\|^2 + \sum_{j\in\mathcal{N}_i} e_{ij}\|\mathbf{p}_j + \mathbf{w}_j^{\ell_j} - A_i\mathbf{p}_j\|^2 + \beta\sum_{j\in\mathcal{N}_i} e_{ij}[\ell_i \neq \ell_j]\Big). \quad (7)$$

With optimization variables $L = \{\ell_i \in \{1, 2, ..., M\}\}_{i=1}^N$, Eq. (7) is a *labeling problem*. We efficiently solve it by using *graph cut* [6].

On optimizing A. By fixing L and E, the optimization problem in Eq. (1) becomes

$$\min_A \sum_{i=1}^{N} V(A_i), \text{ where} \quad (8)$$

$$V(A_i) = \sum_{j\in\mathcal{N}_i} e_{ij}\|\mathbf{p}'_j - A_i\mathbf{p}_j\|^2 + \gamma\|\mathbf{p}'_i - A_i\mathbf{p}_i\|^2. \quad (9)$$

There is no dependence between A_i and A_j for $i \neq j$ in Eq. (8). This property greatly reduces the optimization time,

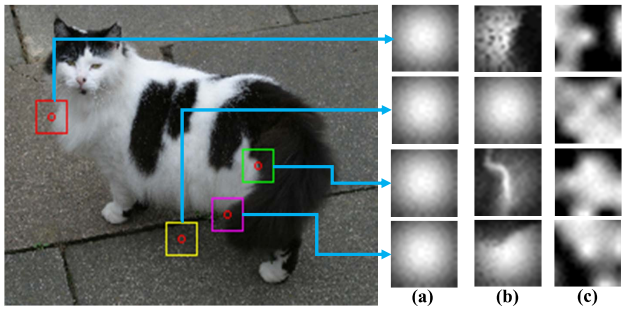

Figure 2. Comparison of three different weighting schemes. (a) Gaussian weight. (b) Bilateral weight. (c) The proposed method.

since the optimal A_i can be independently obtained by solving $V(A_i)$ in Eq. (9). $V(A_i)$ is a *weighted least square problem*, which is convex and has a closed-form solution. By setting the derivative of $V(A_i)$ with respect to A_i to zero, the optimal A_i is obtained by

$$A_i = \Big(\sum_{j\in\mathcal{N}_i^+} e_{ij}\mathbf{p}_j^{'}\mathbf{p}_j^\top\Big)\Big(\sum_{j\in\mathcal{N}_i^+} e_{ij}\mathbf{p}_j\mathbf{p}_j^\top\Big)^{-1}, \tag{10}$$

where $\mathcal{N}_i^+ = \mathcal{N}_i \cup i$ and $e_{ii} = \gamma$.

On optimizing E. With the fixed L and A, we also find the independence between $\mathbf{e}_i$ and $\mathbf{e}_j$ for $i \neq j$ in optimization. Each $\mathbf{e}_i$ can then be independently solved. Specifically, the reduced optimization problem with respect to $\mathbf{e}_i$ is given below

$$\begin{aligned} &\min_{\mathbf{e}_i}\ \alpha\|\mathbf{e}_i - \mathbf{s}_i\|^2 + \mathbf{e}_i^\top \mathbf{r}_i \\ &\text{s.t.}\ \ \mathbf{e}_i \succeq 0,\ \mathbf{e}_i^\top \mathbf{1} = 1, \\ &\text{where}\ \ \mathbf{s}_i = [s_{ij}|j\in\mathcal{N}_i]^\top, \\ &\qquad \mathbf{r}_i = [\|\mathbf{p}_j' - A_i\mathbf{p}_j\|^2 + \beta[\ell_i \neq \ell_j]|j\in\mathcal{N}_i]^\top. \end{aligned} \tag{11}$$

The constrained optimization problem in Eq. (11) is a *convex quadratic programming problem*. We efficiently solve it by using package *CVXGEN* [27].

3.4. Implementation details

In this work, four descriptors are adopted, including SIFT [25], GB [5], DAISY [34] and LIOP [38], and total four flow proposals are generated by SIFT flow and its variants. We set the radius of the neighborhood of each pixel as 10 pixels. The maximal number of iterations T is set as 30 in the experiments, though our approach converges within 10 iterations in most cases. In the alternating optimization, initializing two of the three sets of the optimization variables is required. We choose neighborhood weights $E = \{\mathbf{e}_i\}$ and proposal selectors $L = \{\ell_i\}$ in this work. Each neighborhood weight vector $\mathbf{e}_i$ is initialized by using Gaussian filter. For proposal selector ℓ_i, we compute the descriptor dissimilarity between pixel i and its nearest neighbor in the other image, and the dissimilarity between pixel i and the mapped pixel under each descriptor m and the corresponding flow proposal W^m. We set ℓ_i as that with the largest dissimilarity ratio.

Algorithm 1 Optimization Procedure

Input: Flow candidates, $\{W^m\}_{m=1}^M$; Max iteration, T;
  Initialize E and L;
  for $i \leftarrow 1, 2, \ldots, T$ **do**
    $A \leftarrow \{A_i^*\}$, where A_i^* is optimized via Eq. (10);
    Update $\{s_{ij}\}$ in Eq. (4);
    $E \leftarrow \{\mathbf{e}_i^*\}$, where $\mathbf{e}_i^*$ is optimized via Eq. (11);
    $L \leftarrow L^*$, where L^* is optimized via Eq. (7);
  end for
  Refine labels L via Eq. (12);
Output: Flow field, $W = \{\mathbf{w}_i\}$ where $\mathbf{w}_i \leftarrow \mathbf{w}_i^{\ell_i}$;

In the proposed method, the object-aware neighborhoods are gradually revealed through the iterative procedure. The image alignment result is accordingly improved owing to the high-quality neighborhoods. Figure 2 gives an example for comparing the proposed method to Gaussian and bilateral weights. For the pixel whose weights shown in the first row of the figure, it is a background pixel near the boundary of the object and the background. Gaussian weight cannot get it right since it is isotropic and content-independent. For the background pixel at the second row, it is on a flat area and all weighting schemes work well. For the third row, the pixel is on the object but near appearance discontinuity. Bilateral weight fails because it solely relies on appearance. Finally, the pixel at the last row locates on a junction of three regions: bright object, dark object and dark background. Gaussian cannot distinguish the object and the background well while bilateral weight classifies the dark object and the dark background together because of similar appearance. The proposed scheme properly weights these parts through the iterative process.

Optimizing Eq. (1) for all pixels is computationally very expensive. Hence, we apply our approach to an evenly sampled grid with grid lines spaced every 5 pixels. We obtain an intermediate flow map $\tilde{W} = \{\tilde{\mathbf{w}}_i\}_{i=1}^N$ and the full weight matrix $\tilde{E}$ using bicubic interpolation on flows and weights of the sampled grid. The final flow map is obtained by solving the following labelling problem:

$$\min_L \sum_i^N \left[\|\mathbf{w}_i^{\ell_i} - \tilde{\mathbf{w}}_i\|^2 + \sum_{j\in\mathcal{N}_i} \tilde{e}_{ij}\left(\alpha^{'}[\ell_i \neq \ell_j] + \beta^{'}\|\mathbf{w}_i^{\ell_i} - \mathbf{w}_j^{\ell_j}\|^2\right)\right], \tag{12}$$

in which the data term encourages the flow resemble the intermediate flow while the smoothness term measures inconsistency of labels and flows for neighboring pixels. $\alpha^{'}$ and $\beta^{'}$ are two non-negative constants.

Algorithm 1 summarizes the optimization procedure. The whole algorithm is implemented in MATLAB on a mod-

Table 1. Quantitative comparisons of the proposed method with competing methods. Ours + G./B. denotes the proposed method with a fixed weight map using Gaussian/Bilateral weights.

Method	Descriptor	Dataset			
		VGG	MSRC	Caltech	LMO
SF [24]	SIFT	45.02	41.99	46.08	68.86
	GB	42.71	38.33	50.66	67.80
	DAISY	44.97	41.22	46.68	68.78
	LIOP	47.76	34.87	41.92	66.24
	Con.	49.47	42.20	48.77	63.29
	Avg.	41.18	41.90	49.48	70.27
	FF [21]	47.76	40.40	47.79	69.03
GPM [3]	SIFT	43.99	32.58	37.41	57.44
	GB	41.95	31.41	38.37	57.49
	DAISY	48.44	37.92	39.56	61.05
	LIOP	42.22	25.15	29.44	52.41
	Con.	50.42	38.30	41.47	61.96
	Avg.	39.86	32.04	34.34	56.77
	FF [21]	42.22	34.02	37.15	59.73
DFF [41]	DAISY (R + S)	51.09	40.37	50.95	69.93
Ours + G.	All	53.70	44.67	52.72	71.07
Ours + B.	All	53.64	44.55	53.13	71.02
Ours	All	**54.31**	**46.22**	**54.87**	**72.42**

ern PC with Intel Core i7 3.4GHz CPU. The running time for an image of size 256×256 is about 15 seconds for iterative optimization and less than 1 seconds for the final flow refinement. In addition, the pre-processing steps for generating each feature map and proposal take around 2 and 12 seconds, respectively. Figure 3 plots the convergence of the method and it usually convergences in few steps, around 10 steps in the figure.

4. Experiments

This section describes the datasets and metrics for evaluation, the competing methods and the experimental results.

4.1. Datasets and metrics

We use existing datasets and metrics to evaluate the proposed method on several image alignment tasks. Four datasets are used: VGG dataset [28], MSRC dataset [30, 32], Caltech dataset [18] and LMO dataset [18, 23]. These datasets can be categorized into three groups by tasks.

Dense correspondence. In this task, we find dense pixel correspondences between two images. For this task, we use VGG dataset. It has eight image sets, each of which contains images with five different degrees of a specific variation. There are five types of variations: viewpoint changes, scale changes, image blur, JPEG compression, illumination changes. The dataset provides ground truth correspondences for evaluation. For evaluation of the task, we use the ratio of correct matches, which fall within T pixels from the ground truth correspondences in location. We set T as 0.005 of the image dimension, corresponding to roughly $3 \sim 5$ pixels for these images.

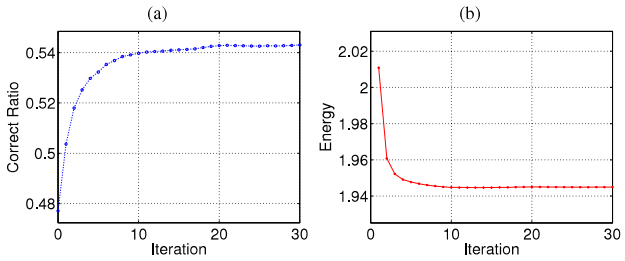

Figure 3. (a) Correct ratio and (b) Energy along with the number of iterations for the VGG dataset. They show that the convergence is fast and steady.

Object matching. The task locates the foreground objects of some object classes. Caltech and MSRC are two popular datasets for the task with large intra-class variations. With a similar setting to previous work [18], we selected 20 and 14 object classes from Caltech and MSRC respectively. For each class, we randomly pick 10 pairs of images and use image alignment to transfer foreground labels from one image to the other. The intersection over union (IoU) metric [11] is used to evaluate the results as it allows us to isolate the matching quality of the foreground objects from the irrelevant backgrounds [18].

Scene matching. While the object matching task is only concerned with foreground/background segmentation, scene matching task annotates each pixel with one of multiple class labels. LabelMe Outdoor (LMO) dataset with 33 class labels is used for the task. We use a similar setting to Kim *et al.* [18] by randomly splitting the test and exemplar images in half ($1,344$ images each). For each test image, we first find its closest exemplar image in GIST space. Then, we use image alignment to find pixel matching and transfer labels from the selected exemplar to the test image accordingly. Accuracy is used as the metric for the task. While measuring accuracy, we only consider the matchable pixels that belong to the classes common to both images.

4.2. Competing methods

We categorize the competing methods into two group: single-feature methods and feature-fusion methods. We

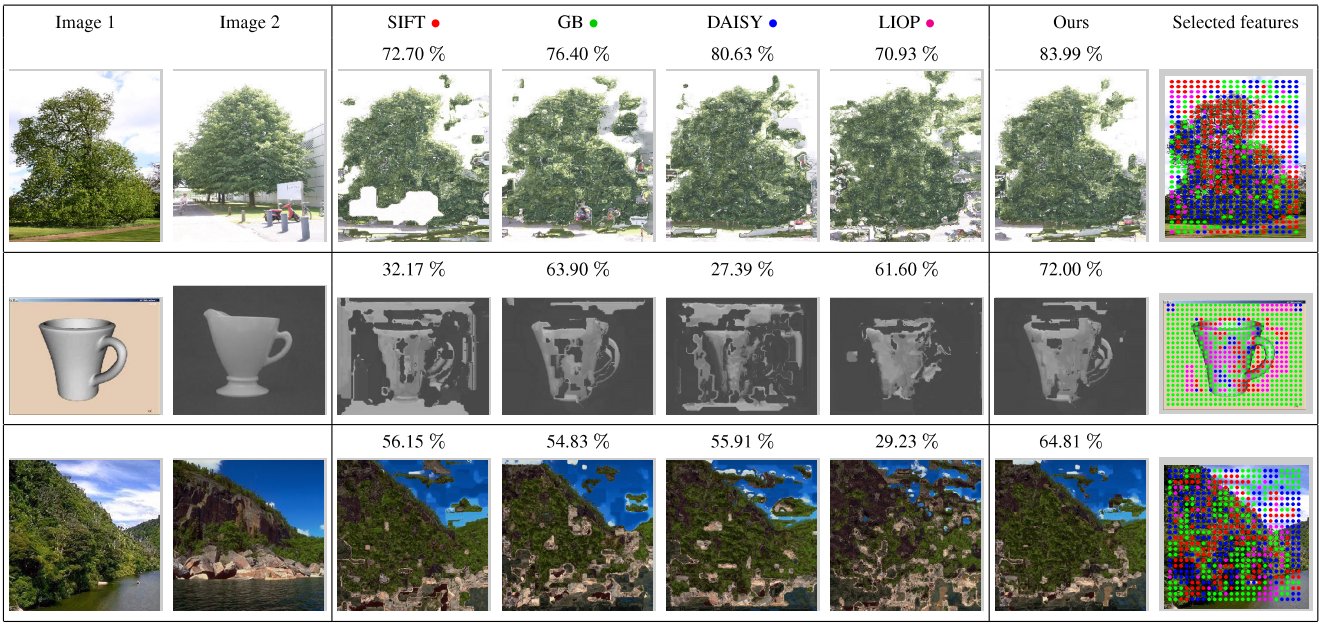

Figure 4. Example image alignment results using the proposed method and SIFT-flow variants with four different feature descriptors. The first one example is from MSRC; the next one from Caltech and the last one from LMO. On the top of each result, we show its numerical score (IoU for MSRC/Caltech and accuracy for LMO). It is clear that features perform quite differently for different examples. By integrating features, the proposed method outperforms any individual one. The last column visualizes the features selected by the proposed method across these images (red for SIFT, green for GB, blue for DAISY and magenta for LIOP).

compare the proposed method to three state-of-the-art image alignment methods: SIFT flow (SF) [24], Generalized PatchMatch (GPM) [3], and DAISY filter flow (DFF) [41]. SF and GPM are single-feature methods. DFF can be regarded as a feature-fusion method because it compares DAISY features across different scales and orientations. For each single-feature methods (SF and GPM), we have four variants with four different features, SIFT, DAISY, GB and LIOP. It gives us a total of 8 single-feature methods. We have also extended SF and GPM to utilize multiple features in three different ways: concatenated feature (Con.), average flow (Avg.) and FusionFlow (FF) [21]. For Con., the four types of features are concatenated as a single feature. It is then fed into a single-feature method for obtaining the flow map. For Avg., we average the four flow maps obtained from four variants of a single-feature method. Finally, given the four flow maps from different features as candidates, FusionFlow [21] fuses them into a single one by computing minimum cuts. This way, we have 6 feature-fusion methods derived from SF and GPM. Therefore, there are a total of 15 competing methods.

4.3. Results

Table 1 summarizes the mean scores of the proposed method and all competing methods. The numeric scores in the table represent the correct ratios, the IoU scores and the accuracy scores for the task of dense correspondence (VGG), object matching (MSRC/Caltech) and scene matching (LMO), respectively. For single-feature methods, we can observe that there is no single best feature for all datasets. By leveraging strengths of different features, the proposed method outperforms all single-feature methods by a margin. Interestingly, although with the advantage of having more features, feature-fusion methods do not always outperform single-feature ones. It shows that fusion of features is not a trivial process. Comparing features in the feature domains could lead to bias so that some features are more dominant. The proposed method compares feature more fairly in a common domain and outperforms all competing methods as shown in Table 1.

We also compare different weighting schemes. We replace the object-aware weights in the proposed method with Gaussian weights and bilateral weights, denoted as Ours+G. and Ours+B. in Table 1. The experiment shows that the object-aware neighborhoods does improve the performance.

Figure 4 presents a visual comparison of the proposed method with several SIFT-flow variants with different features (SIFT, GB, DAISY and LIOP) on image alignment. These examples are from MSRC (the first row), Caltech (the next one) and LMO (the last one). On the top of each result, we show its numerical score (IoU for the first two examples and accuracy for the last one). The last column of Figure 4 visualizes the selected features by the proposed method across images. The red, green, blue and magenta colors denote the SIFT, GB, DAISY and LIOP features respectively.

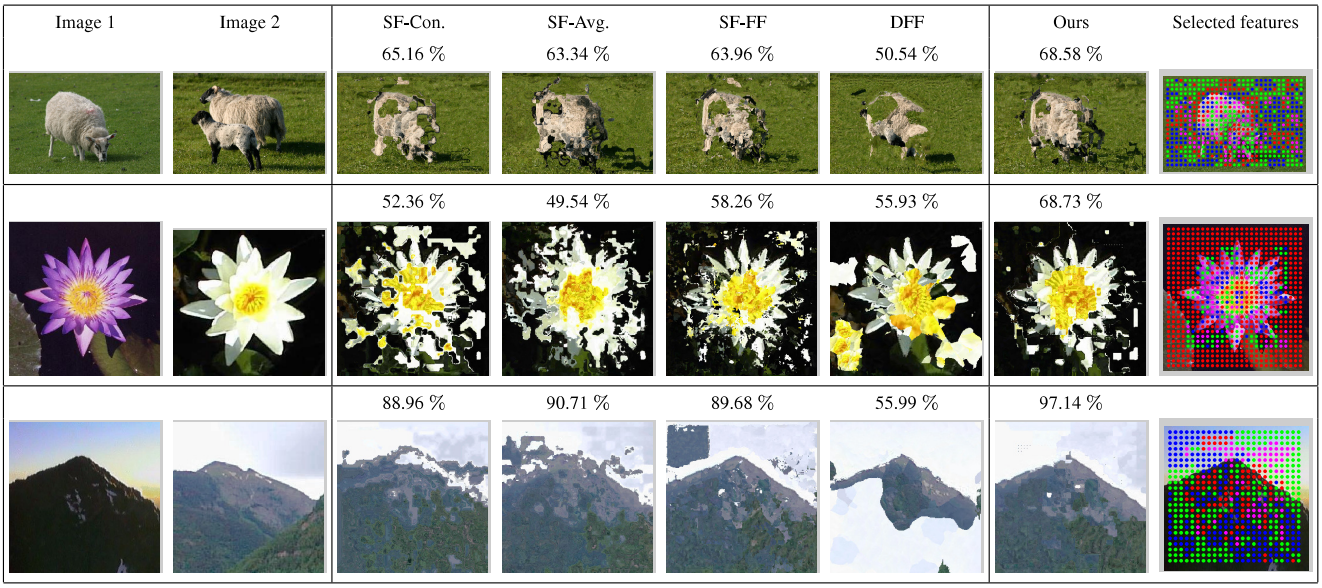

Figure 5. Example image alignment results using the proposed method and some feature fusion methods. The first one is from MSRC, and the second and third examples come from Caltech and LMO, respectively. On the top of each result, we show its numerical score (IoU for MSRC/Caltech and accuracy for LMO). The proposed method outperforms all of them, showing that the proposed scheme for feature and neighborhood selection is more effective. The last column visualizes the features selected by the proposed method across these images (red for SIFT, green for GB, blue for DAISY and magenta for LIOP).

It is evident from the figure that these features are complementary and there is no single feature suitable for all visual variations. For example, in the example of the first row, the trees exhibit view variations in two images. DAISY is known for viewpoint invariance and thus it performs better than other features individually. For the cups in the second row, the geometric shape is more distinctive and the shape-based descriptor, GB, performs the best. The example in the last row has multiple classes and each has its own specific feature. Thus, three features, SIFT, GB and DAISY, have similar performance. Our method can leverage the strengths of individual features by feature selection. It selects the DAISY feature for more parts of the tree in the first example, the GB feature for most parts around the silhouette of the cup in the second example.

Figure 5 shows several image alignment results with the proposed method and several feature fusion methods, SIFT-flow with concatenated features, average flow, FusionFlow and DAISY filter flow on several challenging examples including multi-objects (the first row), dramatic color difference (the second row) and textureless regions (the third row). The proposed method outperforms other methods quantitatively and visually in most examples by leveraging multiple features..

Limitations: Our approach acts as an extra layer to fuse multiple flow proposals. It will fail in the cases where none of them is good enough. However, the proposed formulation for proposal fusion is general in the sense that it makes no assumption about how these proposals are yielded. Thus, the aforementioned issue could be alleviated by using complementary proposals with different combinations of descriptors and image alignment algorithms, such as SIFT-flow [23], DSP [18], PatchMatch [2, 3] or DFF [41].

5. Conclusions

We have introduced an image alignment method with the capability of pixel-specific feature selection and neighborhood construction. The affine matrix learned within a local neighborhood is adopted as a common domain for feature selection. At the same time, a matching-guided object-aware neighborhood is estimated through the estimated affine matrix and flows. By improving the neighborhood, we have a better estimate of the affine matrix. On the other hand, a more accurate affine matrix leads to better neighborhood construction. Experiments show that the proposed method outperforms the state-of-the-art methods. In the future, we would like to overcome the limitations and integrate sparse matching into our method. In addition, it would also be interesting to investigate how the proposed idea can be extended to other domains such as scene parsing and image classification.

Acknowledgments. This work was supported in part by grant NSC 101-2628-E-002-031-MY3 and MOST 103-2221-E-001-026-MY2.

References

[1] A. E. Abdel-Hakim and A. A. Farag. CSIFT: A SIFT descriptor with color invariant characteristics. In *CVPR*, 2006. 2

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. In *SIGGRAPH*, 2009. 2, 8

[3] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. In *ECCV*, 2010. 1, 2, 3, 6, 7, 8

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 2002. 2

[5] A. C. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, 2001. 1, 2, 5

[6] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *TPAMI*, 2001. 4

[7] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *TPAMI*, 2011. 3

[8] H.-Y. Chen, Y.-Y. Lin, and B.-Y. Chen. Robust feature matching with alternate hough and inverted hough transforms. In *CVPR*, 2013. 2

[9] M. Cho, J. Lee, and K. M. Lee. Unsupervised matching of deformable objects by agglomerative correspondence clustering. In *ICCV*, 2009. 2, 4

[10] T. Deselaers and V. Ferrari. Global and efficient self-similarity for object classification and detection. In *CVPR*, 2010. 2

[11] M. Everingham, L. J. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 6

[12] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. In *SIGGRAPH*, 2011. 1, 2

[13] T. Hassner, V. Mayzels, and L. Zelnik-Manor. On sifts and their scales. In *CVPR*, 2012. 2

[14] D. Hauagge and N. Snavely. Image matching using local symmetry features. In *CVPR*, 2012. 2

[15] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodestic support weights. In *ICIP*, 2009. 3

[16] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *TPAMI*, 2013. 2, 3

[17] K. Karsch, C. Liu, and S. B. Kang. DepthTransfer: Depth extraction from video using non-parametric sampling. *TPAMI*, 2014. 1

[18] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013. 2, 3, 6, 8

[19] T. Kim, H. Lee, and K. Lee. Optical flow via locally adaptive fusion of complementary data costs. In *ICCV*, 2013. 3, 4

[20] I. Kokkinos and A. Yuille. Scale invariance without scale selection. In *CVPR*, 2008. 2

[21] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-Continuous optimization for optical flow estimation. In *CVPR*, 2008. 3, 4, 6, 7

[22] W.-Y. Lin, M.-M. Cheng, J. Lu, H. Yang, M. N. Doo, and P. Torr. Bilateral functions for global motion modeling. In *ECCV*, 2014. 3

[23] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *TPAMI*, 2011. 1, 2, 6, 8

[24] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *TPAMI*, 2011. 1, 2, 3, 6, 7

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2, 5

[26] J. Lu, K. Shi, D. Min, L. Lin, and M. N. Do. Cross-based local multipoint filtering. In *CVPR*, 2012. 3

[27] J. Mattingley and S. Boyd. CVXGEN: A code generator for embedded convex optimization. *Optimization and Engineering*, 2012. 5

[28] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 2005. 2, 6

[29] C. Rother, V. Kolmogorov, and A. Blake. GrabCut - Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 3

[30] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013. 1, 2, 6

[31] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007. 2

[32] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 2009. 2, 6

[33] M. Stoll, S. Volz, and A. Bruhn. Adaptive integration of feature matches into variational optical flow methods. In *ACCV*, 2012. 3

[34] E. Tola, V. Lepetit, and P. Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *TPAMI*, 2010. 2, 5

[35] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, 1998. 1, 3

[36] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer. Dense segmentation-aware descriptors. In *CVPR*, 2013. 3

[37] J. van de Weijer, T. Gevers, and A. D. Bagdanov. Boosting color saliency in image feature detection. *TPAMI*, 2006. 2

[38] Z. Wang, B. Fan, and F. Wu. Local intensity order pattern for feature description. In *ICCV*, 2011. 1, 2, 5

[39] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *ICCV*, 2013. 3

[40] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *TPAMI*, 2012. 3, 4

[41] H. Yang, W.-Y. Lin, and J. Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *CVPR*, 2014. 2, 3, 6, 7, 8

[42] Q. Yang. A non-local cost aggregation method for stereo matching. In *CVPR*, 2012. 3

[43] K.-J. Yoon and I. Kweon. Adaptive support-weight approach for correspondence search. *TPAMI*, 2006. 3