DAISY Filter Flow: A Generalized Discrete Approach to Dense Correspondences[*]

Hongsheng Yang[†], Wen-Yan Lin[*], and Jiangbo Lu[*]
[*]Advanced Digital Sciences Center, Singapore [†]University of North Carolina at Chapel Hill, USA

Abstract

Establishing dense correspondences reliably between a pair of images is an important vision task with many applications. Though significant advance has been made towards estimating dense stereo and optical flow fields for two images adjacent in viewpoint or in time, building reliable dense correspondence fields for two general images still remains largely unsolved. For instance, two given images sharing some content exhibit dramatic photometric and geometric variations, or they depict different 3D scenes of similar scene characteristics. Fundamental challenges to such an image or scene alignment task are often multifold, which render many existing techniques fall short of producing dense correspondences robustly and efficiently. This paper presents a novel approach called DAISY filter flow (DFF) to address this challenging task. Inspired by the recent PatchMatch Filter technique, we leverage and extend a few established methods: 1) DAISY descriptors, 2) filter-based efficient flow inference, and 3) the PatchMatch fast search. Coupling and optimizing these modules seamlessly with image segments as the bridge, the proposed DFF approach enables efficiently performing dense descriptor-based correspondence field estimation in a generalized high-dimensional label space, which is augmented by scales and rotations. Experiments on a variety of challenging scenes show that our DFF approach estimates spatially coherent yet discontinuity-preserving image alignment results both robustly and efficiently.

1. Introduction

Estimating a set of dense correspondences for two given images is an important cornerstone to a number of computer vision and graphics applications. Typically the resultant dense correspondence field is desired to be spatially coherent while preserving motion or structural discontinuities. Impressive advancements have been made in the past years for matching a pair of adjacent images either in time or in viewpoint, spawning several state-of-the-art techniques for optical flow [23] and stereo matching [19]. However, opposed to this comparatively restrictive setup, matching two general images that exhibit high variability in appearance is far more complicated [5, 12]. A variety of factors make this task very challenging, including significant photometric differences (e.g. exposure and tone variations) and non-rigid geometric transformation (e.g. scale and rotation changes), or even different scene contents between the two images.

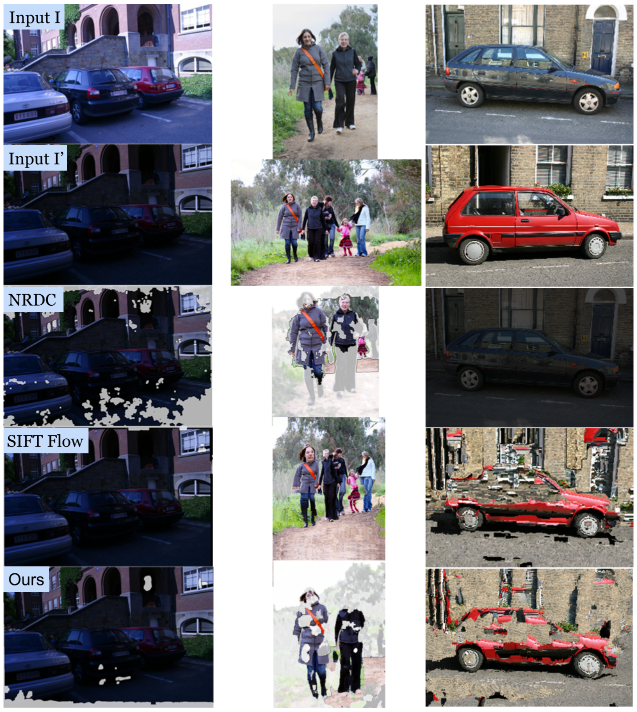

Figure 1. Some typical challenges of matching a pair of images. First two rows: two input images with significant photometric, geometric and scene content changes (from left to right). Third to fifth rows: image warping results from the second image to the first one using the estimated correspondences by NRDC [5], SIFT Flow [12], and our DFF method. DFF yields dense coherent matches consistently. NRDC gives *no* match for different scenes.

Noticeable previous attempts at estimating dense correspondences under difficult conditions are, for instance, NRDC [5] and SIFT Flow [12]. They deal with a pair of images either sharing some content but exhibiting dramatic photometric and geometric variation, or depicting different

[*]This study is supported by the HSSP research grant at the ADSC from Singapore's Agency for Science, Technology and Research (A*STAR). This work was mainly done when Hongsheng was working at ADSC. Corresponding author: jiangbo.lu@adsc.com.sg.

scenes of similar appearances, respectively. Albeit impressive matching results and applications are shown in these works, modeling photometric variations in NRDC [5] is not robust to more general cases when the appearance differences cannot be accounted for simply with parametric color transforms. Further, it also cannot handle matching across scenes. The SIFT Flow algorithm [12] uses fixed-scale SIFT descriptors [13] densely for the entire image lattice, so it cannot match the scenes consisting of non-rigid, spatially varying deformations (e.g. scale and rotation changes) well. Fig. 1 illustrates some typical cases where the existing methods fail to provide dense coherent matches.

Following the same spirit of the SIFT Flow in using densely computed descriptors, this paper presents a generalized image matching algorithm called *DAISY Filter Flow* (DFF). The DFF algorithm achieves much more robust performance in efficiently matching images of challenging non-rigid photometric and geometric variations, or across different scenes than the existing techniques [9, 5, 12, 6, 21]. Our approach is built upon a few established techniques but also extends them, which are 1) DAISY descriptors, 2) filter-based efficient flow field inference, and 3) the PatchMatch fast search. Inspired by the recent PatchMatch Filter (PMF) work [15], we cast dense correspondence estimation in a discrete labeling framework, and tightly integrate and optimize these selected modules with image segments as the bridge. Motivated by the known difficulty of pixel correlation-based methods for matching challenging images [20, 12] (see the PMF result in Fig. 8), this paper generalizes the PMF method [15] in two important ways: i) DAISY descriptors are employed and extended for general image matching; ii) to search across scales and rotations beyond just translations. As a result, our DFF algorithm, for the first time, allows performing spatially regularized, dense descriptor-based correspondence field estimation efficiently in a high-dimensional space. Being able to do so explains the key advantages of the DFF method in both matching robustness (see Fig. 1) and computational efficiency.

1.1. Related Work and Key Design Factors

Below we review the related prior work and motivate a few key design factors for dense correspondence estimation.

Dense descriptors. Several patch-based descriptors have been developed such as SIFT [13] and SURF [3], which find many applications due to their robustness to perspective and illumination changes. When it comes to dense pixel-wise matching, computational complexity becomes a critical design factor, in addition to the transform-invariance power sought for. Tola *et al.* [20] proposed a local region descriptor called DAISY, which is very efficient to compute densely thanks to the constructional scheme of reusing shared histograms across pixels. They showed that DAISY outperforms SIFT in wide-baseline stereo matching, while running about 60x faster [20]. However, DAISY's current design [20] deals only with rigid camera motions and assumes the two given images are calibrated. To tackle more general image matching tasks, our DFF approach generalizes the standard DAISY descriptor, avoiding unreliable region scale and rotation decision for DAISY descriptors deterministically from a single image.

Transform invariance. As photometric (e.g. illumination and tone) and geometric differences (e.g. scales and rotations) often exist between a pair of general images, image alignment or dense correspondence algorithms need to be robust against these transformations. As discussed earlier, NRDC [5] does not work well for complicated appearance variations and also is not capable of matching across different scenes. SIFT Flow [12] works fine for matching different scenes with sufficiently similar characteristics, but is not robust to large changes in scale and orientation. Showing that scale selection is difficult and unreliable for the majority of pixels, Hassner *et al.* [6] extracted a set of SIFTs for each pixel at multiple scales, and then used a subspace-mapped descriptor representation to match scenes/objects in different scales. However, the improved matching quality comes at a significant computational price. The Scale-Invariant Descriptors (SID) [9] is shown in [6] to be less capable of matching across difference scenes. Based on large patches (hence sensitive to occlusions) and many convolution layers, SID requires a considerable amount of memory even for a small-sized image. Recently, a deformable spatial pyramid matching (DSP) method [8] is proposed. Though efficient, DSP is not rotation-invariant and cannot handle image pairs with challenging object pose or viewpoint changes nor complex geometric variations. Its regular spatial partitioning of the image grid tends to impose improper regularization for differently moving objects, incurring strong 4×4 blocking artifacts in the estimated flows.

The proposed DFF approach shares some similar ideas with [6, 10] in that we do not fix but rather allow "enumerating" a range of admissible scales [6, 10] and rotations [10] to describe features around each pixel. However, our method is able to achieve excellent computational efficiency, even when fine-grained enumerations are chosen adaptively for each pixel in an extended search space.

Efficient inference and label search. As pointed out in [12], the bottleneck of SIFT Flow lies in the large search window size, so computational efficiency is considered as an important direction for improvement. In fact, SIFT features by design support varying descriptors in scales and rotations, but simply enlarging the descriptor representation space will only cause a huge computational load further to SIFT feature computation and belief propagation optimization in SIFT Flow [12]. As a result, SIFT Flow [12] as well as the original DAISY-based wide-baseline stereo [20] used a fixed scale globally for all region descriptors. Recently, the generalized PatchMatch technique [2] has shown its

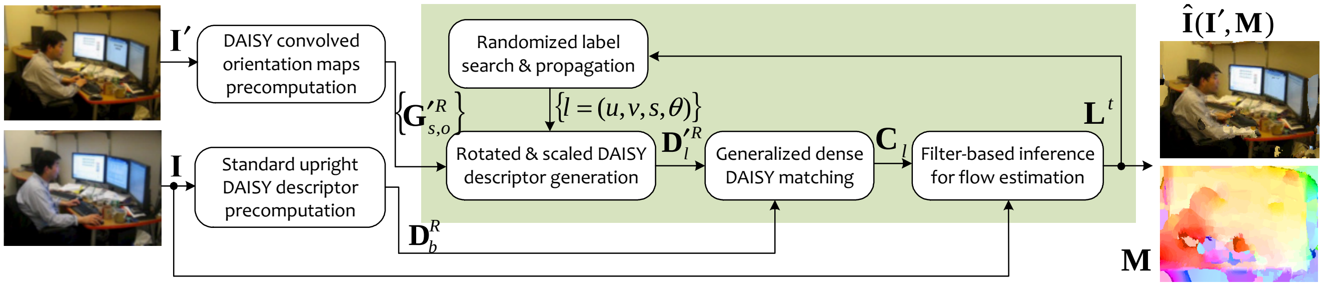
$\mathbf{I}'$
$\mathbf{I}$
DAISY convolved orientation maps precomputation
Standard upright DAISY descriptor precomputation
Randomized label search & propagation
$\{\mathbf{G}'^R_{s,o}\}$
$\{l=(u,v,s,\theta)\}$
Rotated & scaled DAISY descriptor generation
$\mathbf{D}'^R_l$
Generalized dense DAISY matching
$\mathbf{C}_l$
Filter-based inference for flow estimation
$\mathbf{L}^t$
$\mathbf{D}^R_b$
$\hat{\mathbf{I}}(\mathbf{I}',\mathbf{M})$
$\mathbf{M}$

Figure 2. DAISY filter flow algorithm pipeline. The algorithm is comprised of two phases: DAISY precomputation and online matching.

powerful strength in quickly finding a dense nearest neighbor field (NNF) across scales and rotations beyond just translations. However, the NNF algorithm enforces no spatial smoothness constraint, so the NNF results are largely incorrect and do not reveal true motion flows. In fact, a Markov Random Field (MRF)-based inference formulation is good at enforcing spatial regularization, but global energy minimization methods involved become very slow for the large label space. Inspired by the recent success in speeding up the MRF inference with filter-based methods [18, 22], we explore in this paper a synegetic combination of the filter-based inference with the randomized PatchMatch search in a high-dimensional label space. The proposed DFF framework resembles that of the PatchMatch Filter [15], but generalizes it for efficient, dense descriptor-based general image matching in a high-dimensional space.

2. DAISY Filter Flow System Overview

Given a pair of images $\mathbf{I}$ and $\mathbf{I}'$, the goal of dense correspondence estimation is to generate a spatially coherent, discontinuity-preserving motion field $\mathbf{M} = \{m(p) = (u(p), v(p))\}$ for each pixel $p = (x_p, y_p) \in \mathbf{I}$. This task is also known as image alignment or registration. The estimated correspondence field $\mathbf{M}$ can be used e.g. for photo enhancement through the reconstructed image $\hat{\mathbf{I}}(\mathbf{I}', \mathbf{M})$ warped from $\mathbf{I}'$ [5], or for transferring the warped scene labels from a database image $\mathbf{I}'$ to parse $\mathbf{I}$ [11].

Motivated by the recent major advance in fast dense nearest neighbor field search [2] and filter-based methods for efficient inference [18], we take a discrete optimization approach and cast the dense descriptor-based correspondence estimation in a high-dimensional label space. More specifically, our DFF algorithm aims to infer a spatially coherent labeling field $\mathbf{L} = \{l(p) = (u(p), v(p), s(p), \theta(p))\}$ and assign an optimal label $l(p)$ to each pixel $p \in \mathbf{I}$. The locally varying random variables for the scale $s(p)$ and orientation $\theta(p)$ are newly introduced, which are associated with each region descriptor centered at pixel p to deal with the high appearance variability between the two images. This design counteracts the negative effects of deciding an invariant feature scale and orientation rigidly for the dense image lattice [12], as such a decision is unreliable for a majority of inconspicuous pixels in an image [6]. We choose to integrate and extend the DAISY descriptor [20] in the proposed framework, because of its favorable performance in computational efficiency and robustness. The DAISY descriptor is densely computed to describe regions around each pixel, similar to the dense SIFT feature in SIFT Flow [12].

The pipeline of the proposed DFF algorithm is shown in Fig. 2. At a high level, the entire system can be partitioned into two phases: i) precomputing the standard upright DAISY descriptors for $\mathbf{I}$ and the *convolved orientation maps* [20] for $\mathbf{I}'$ (Sect. 3.1), and ii) iterative evaluation of matching costs of hypothetical labels (Sect. 3.2) and filter-based flow inference (Sect. 3.3), which are efficiently performed for a significantly reduced subset of plausible labels generated by a randomized label search and propagation scheme (Sect. 3.4). All these algorithmic modules are carefully designed and seamlessly integrated together.

3. The DAISY Filter Flow Algorithm

3.1. The DAISY Descriptor and Precomputation

As introduced earlier, dense DAISY descriptors are efficient to compute because histograms computed for one region can be reused for all neighboring pixels, and also the DAISY computation pipeline enables a very efficient memory access pattern [20]. Thanks to this computational advantage and also our fast inference and label search methods to be detailed later, we need not adopt a coarse-to-fine (C2F) matching scheme used in SIFT Flow [12] to reduce the complexity stress. Instead, we perform DAISY descriptor computation and matching for the full image grid. This effectively addresses the fundamental limitation of a conventional C2F framework that does not handle motion details or large displacements of small objects well [23].

Fig. 3(a) shows the standard upright DAISY descriptor $\mathbf{D}^R_b(p)$ centered on pixel p. R denotes the distance from the center pixel p to the outermost sampling grid points [20]. Given a discretized transformation label $l = (u, v, s, \theta)$, the generalized DAISY descriptor, scaled by s, rotated by θ and centered on the translated pixel $p' = p + (u, v)$, is denoted by $\mathbf{D}^R_l(p) \equiv \mathbf{D}^R_{s,\theta}(p')$ as shown in Fig. 3(b). $\mathbf{D}^R_b(p)$ is a special case of $\mathbf{D}^R_l(p)$, when $l = (0, 0, 1, 0)$.

As will be discussed in Sect. 3.2, generalized DAISY descriptors need to be generated for any given hypothetical label l on the fly for the target image $\mathbf{I}'$. Therefore, it

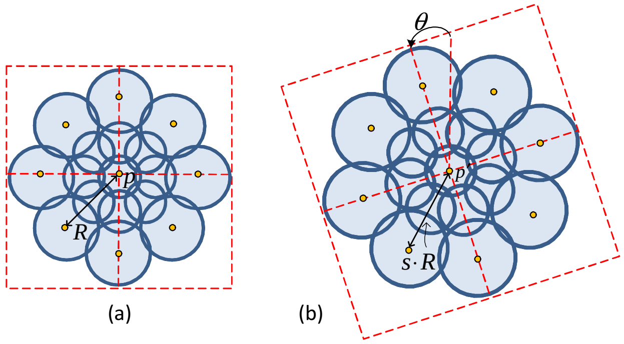

Figure 3. (a) The standard upright DAISY descriptor $\mathbf{D}_b^R(p)$ centered at pixel p [20]. (b) A generalized DAISY descriptor $\mathbf{D}_{s,\theta}^R(p')$ scaled by s, rotated by θ and centered at a translated location p'.

is desired if this descriptor generation process can be made highly efficient. This design concern pinpoints another important reason for employing DAISY features in our proposed framework: all the *convolved orientation maps* [20], denoted as $\{\mathbf{G}'^R_{s,o}\}$ in Fig. 2, can be precomputed for $\mathbf{I}'$ and reused with a very small overhead to generate $\mathbf{D}'^R_l(p)$ or equivalently $\mathbf{D}'^R_{s,\theta}(p')$ during the runtime. Generating $\mathbf{D}'^R_{s,\theta}(p')$ from $\{\mathbf{G}'^R_{s,o}\}$ amounts to simply rotating the sampling grid of concentric circles by θ (e.g. orange dots in Fig. 3(b)), and then shifting circularly the concatenated histogram bins. We refer the readers to [20] for specific details of DAISY descriptors. In this paper, we set the number of convolved orientation layers to two, namely using two rings of concentric circles around the center pixel p. This choice limits the negative effects of occlusion when compared to larger regions. Our DAISY feature precomputation strategy for a pair of given images $\mathbf{I}$ and $\mathbf{I}'$ is summarized as follows. We precompute and store the standard upright DAISY descriptors $\mathbf{D}_b^R$ densely for $\mathbf{I}$. The convolved orientation maps $\{\mathbf{G}'^R_{s,o}\}$ for $\mathbf{I}'$ are precomputed and stored for a discrete set of predefined scale coefficients E with $s \in E$, and also a set of quantized rotations $\Theta = \{\theta_o, o \in [1, H]\}$ for which the image gradient norms are computed.

3.2. Generalized Dense DAISY Matching

Unlike the optical flow techniques which typically build on the brightness constancy assumption between corresponding pixels, the proposed DFF method uses the densely computed feature descriptor distance to evaluate the matching evidence as SIFT Flow [12]. However, our DFF method overcomes the limitations of SIFT Flow that uses scale-fixed upright regions to compute SIFT features, and it can reliably estimate non-rigid motion with pronounced scale and rotation changes at local region or pixel levels.

Given a hypothetical label $l = (u, v, s, \theta)$, the generalized DAISY matching cost $\mathbf{C}_l(p)$ between a pixel $p \in \mathbf{I}$ and its correspondence candidate $p'(= p + (u, v)) \in \mathbf{I}'$ is computed as the truncated L1 distance between two descriptors:

$$\mathbf{C}_l(p) = \min\left(\left\|\mathbf{D}_b^R(p) - \mathbf{D}'^R_{s,\theta}(p')\right\|_1, t\right) . \quad (1)$$

The truncation threshold t is used to account for matching outliers and occlusions. Similarly with the strategies in [2, 5], we compare an upright unscaled patch descriptor $\mathbf{D}_b^R(p)$ for pixel $p \in \mathbf{I}$, with a patch descriptor $\mathbf{D}'^R_{s,\theta}(p')$ that is scaled by s and rotated by θ around pixel $p' \in \mathbf{I}'$. Thanks to the precomputed convolved orientation maps $\{\mathbf{G}'^R_{s,o}\}$ for the image $\mathbf{I}'$, $\mathbf{D}'^R_{s,\theta}(p')$ can be quickly generated. With two readily available vectors, i.e. $\mathbf{D}_b^R(p)$ and $\mathbf{D}'^R_{s,\theta}(p')$, $\mathbf{C}_l(p)$ for a random label l can be efficiently computed recursively.

3.3. Filter-Based Inference for Flow Estimation

Inspired by the recent filter-based methods as a fast alternative to solve multi-labeling problems in computer vision [18, 22], we take a filtering-based approach here. This design choice is contrasted with the belief propagation optimization method in SIFT Flow [12], which becomes very slow for the large label space or high image resolutions [18].

As the first attempt at enforcing the spatial smoothness on the descriptor-based raw label cost, the proposed DFF method applies an edge-preserving filtering approach for efficient label inference as follows. First, given the raw matching cost $\mathbf{C}_l(p)$ evaluated for the pixel p and the label l in (1), the filtered matching cost $\bar{\mathbf{C}}_l(p)$ is computed as:

$$\bar{\mathbf{C}}_l(p) = \sum_{q \in W^r(p)} \lambda_{q,p}(\mathbf{I})\mathbf{C}_l(q) , \quad (2)$$

where $W^r(p)$ is the local aggregation window centered at pixel p. The filter kernel radius is denoted by r. A variety of fast edge-aware filters [17, 7, 14] can be used to calculate the contribution $\lambda_{q,p}(\mathbf{I})$ of a support pixel q adaptively. They commonly utilize the input image $\mathbf{I}$ to guide the filtering process and generate a spatially smooth yet discontinuity-preserving filtered result. Such a result resembles that of applying a global message passing algorithm to a MRF formulation, but it is obtained much faster and avoids a C2F scheme. We choose the linear-time CLMF-0 filtering technique [14] here for its favorable filtering quality and speed trade-off, but other filters can also be adopted.

With the aggregated matching cost $\bar{\mathbf{C}}_l(p)$, the optimal label l_p for each pixel p is progressively updated with a Winner-Takes-All (WTA) scheme:

$$l_p = \text{arg min}_{l \in \mathcal{L}} \bar{\mathbf{C}}_l(p) , \quad (3)$$

where $\mathcal{L}$ denotes the four-dimensional label space of $\{(u, v, s, \theta)\}$. Occlusion detection via cross checking [18, 15] and label post-refinement can be optionally applied.

3.4. Randomized Label Search with Regularization

Though filter-based alternatives provide very competitive runtime over global optimization based methods [18], exhaustively evaluating the raw and aggregated costs $\mathbf{C}_l(p)$ and $\bar{\mathbf{C}}_l(p)$ as in (1,2) for every single label $l \in \mathcal{L}$ is still prohibitively time-consuming. The reason is that the complexity scales linearly with this high-dimensional label space

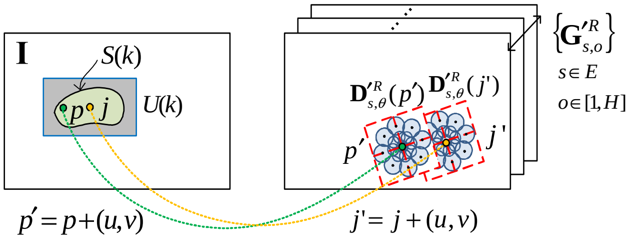
I
S(k)
p
j
U(k)
$\mathbf{D}'^{R}_{s,\theta}(p')$
$\mathbf{D}'^{R}_{s,\theta}(j')$
p'
j'
$\{\mathbf{G}'^{R}_{s,o}\}$
$s \in E$
$o \in [1, H]$
$p' = p + (u, v)$
$j' = j + (u, v)$

Figure 4. Segment-based collaborative label search and hypothetical DAISY descriptors built from $\{\mathbf{G}'^{R}_{s,o}\}$. See the text for details.

size, i.e., $|\mathcal{L}| = L^u * L^v * |E| * H$, where $(L^u, L^v, |E|, H)$ denote the discrete search states for each dimension of (u, v, s, θ), respectively. As motivated earlier, the generalized PatchMatch machinery [2] can perform the nearest neighbor search across translations, rotations, and scales highly efficiently with a complexity of $O(\log|\mathcal{L}|)$. However, without enforcing any smoothness constraints, the PatchMatch algorithm generates very noisy and incorrect motion for most parts of an image [2, 23]. We therefore are interested in developing an approach to synegetically combine and take advantage of the two techniques, i.e., leveraging the label cost filtering to implicitly but efficiently enforce the (local) spatial smoothness prior, and the fast randomized PatchMatch search in a high-dimensional label space.

To this end, we follow the key idea of PatchMatch Filter [15] to use segments or superpixels [1] as the bridge to connect the two techniques developed separately for different purposes. The rationale is that segments group similar pixels into spatially compact atomic regions, so they not only reduce the image representation complexity, but also offer good potential to exploit the spatial regularization and computational redundancy at the segment level. In line with this reasoning, the input image $\mathbf{I}$ is first partitioned into a set of disjoint K segments $\mathbf{I} = \{S(k), k = 1, 2, ..., K\}$, and an adjacency graph $\mathcal{G}$ is built with segments $\{S(k)\}$ as the graph nodes. As the key processing routine for the proposed DFF algorithm, the segment-based label search and matching cost computation for a group of pixels are performed collaboratively as shown in Fig. 4 and explained below.

Given a hypothetical label $l = (u, v, s, \theta)$, take example for two pixels $p, j \in S(k)$, their corresponding candidate DAISY descriptors $\mathbf{D}'^{R}_{s,\theta}(p')$ and $\mathbf{D}'^{R}_{s,\theta}(j')$ are retrieved and built from the prestored maps $\{\mathbf{G}'^{R}_{s,o}\}$ based on the identical similarity transformation. Next, the raw and aggregated costs $\mathbf{C}_l(p, j)$ and $\bar{\mathbf{C}}_l(p, j)$ are evaluated as (1,2). To allow for full-kernel filtering for segment boundary pixels $\partial S(k)$ and also more regular data prefetch and storage, we expand the minimum rectangle enclosing $S(k)$ by r pixels outwards and denote the slightly enlarged rectangular region as $U(k)$. The raw cost computation is applied to all pixels in $U(k)$.

Now we present a segment-based PatchMatch algorithm for an augmented four-dimensional label space, generalizing the recent PatchMatch Filter [15]. The basic workflow is close to that of the generalized PatchMatch method [2], i.e., two sets of label candidates from the *propagation* and *random search* steps are evaluated for each graph node in scan order iteratively. The search process stops when the maximum iteration number is reached or until convergence. The major difference between the algorithm here and the generalized PatchMatch method [2] is caused by the graph structures, as the segment-based adjacency graph $\mathcal{G}$ usually contains a variable number of neighbors for each node/segment $S(k)$. The proposed DFF approach is summarized in Algorithm 1. Some major algorithmic steps differing from the generalized PatchMatch algorithm [2] are explained as follows. In Step **3**, each of the spatially neighboring nodes/segments adjacent to segment $S(k)$ selects a "good" label giving rise to a low DAISY matching distance, and propagates it to $S(k)$. These labels collectively make a propagated hypothetical label set L^N for $S(k)$. Though more elaborate schemes to select "good" labels can be designed, we find that it works fine by randomly sampling a pixel belonging to each neighboring segment, and then

Algorithm 1: DAISY filter flow estimation process

```
Input: (1) The precomputed standard DAISY descriptors
       $\mathbf{D}^R_b$ for image $\mathbf{I}$. (2) The precomputed DAISY
       convolved orientation maps $\{\mathbf{G}'^{R}_{s,o}\}$ for image $\mathbf{I}'$.
Discrete label space: $\mathcal{L} = [u_1,\ u_2] \times [v_1,\ v_2] \times E \times \Theta$.
Output: The estimated pixel-wise label map
        $\mathbf{L} = \{l(p) = (u(p), v(p), s(p), \theta(p))\}$.
/* Initialization */
1: Partition $\mathbf{I}$ into a set of disjoint $K$ segments
$\mathbf{I} = \{S(k), k = 1, 2, ..., K\}$ and build adjacency graph $\mathcal{G}$.
2: Assign an initial label $l^{\mathbf{0}} = (0, 0, 1, 0)$ to each segment
$S(k)$. For each pixel $p \in S(k)$, set $l_p = l^{\mathbf{0}}$.
/* Iterative label search and optimization */
repeat
    for $k = 1 : K$ do
        3: Propagate a set of labels $L^N$ randomly sampled
        from neighboring segments to the segment $S(k)$.
        for $l \in L^N$ do
            4: Evaluate the raw DAISY matching cost
            $\mathbf{C}_l(q)$ for each pixel $q \in U(k)$ with Eq. (1).
            5: Compute the aggregated cost $\bar{\mathbf{C}}_l(p)$ for each
            pixel $p \in S(k)$ with Eq. (2).
            if $\bar{\mathbf{C}}_l(p) < \bar{\mathbf{C}}_{l_p}(p), \forall p \in S(k)$ then
                6: $l_p \longleftarrow l$.
        7: Decide for $S(k)$ a representative label $l^*_k$ and
        generate a set of random labels $L^E$ around $l^*_k$. The
        $(u, v, s, \theta)$ components of random labels are
        generated as [2].
        8: Perform random label candidates evaluation and
        update by following Step 4–6 for $l \in L^E$.
until convergence or the maximum iteration number.
```

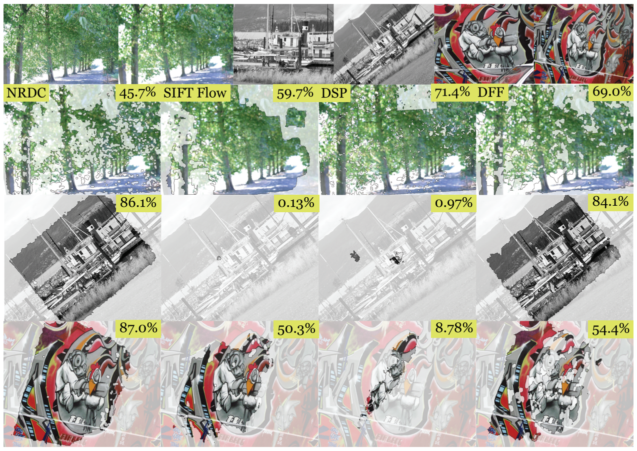

Figure 5. Comparison of NRDC [5], SIFT Flow [12], DSP [8], and our DFF method (from left to right) on test images [16] featuring large changes in sharpness, planar scale and rotation, and viewpoint. Only correct matches are highlighted with the percentages.

propagating its best label visited so far to $S(k)$. In Step **7**, there exists another major difference from [2] in the random search phase, because pixels are the basic units in [2] while here the operations need to be performed for a segment. This means a "representative" label l_k^* for the pixels covered in the segment $S(k)$ shall be decided first, before applying the random label sampling around the picked l_k^*. Again, more sophisticated methods can be developed to recommend the "representative" label l_k^*, e.g. by evaluating the consensus level of a label candidate among the pixels in $S(k)$, but we simply draw a pixel from the segment $S(k)$ and assign its current best label to l_k^*. Using this simple scheme also attributes to the homogeneous region delineation power of image segmentation methods. Provided with l_k^*, we follow the random search procedure in [2], and sample around l_k^* using windows of exponentially decreasing sizes for four random variables (u, v, s, θ). As [15], we choose SLIC superpixels [1] for the favorable performance.

4. Experiments

The DFF algorithm was implemented based on the publicly available DAISY code. The DAISY feature related parameters were typically set as: $R = 8$ (increased to 16 for large images), $s \in E = \{0.5, 1.0, 1.5, 2.0, 2.5\}$, and $H = 7$, namely $\theta \in \Theta = \{15 * o, o \in [-3, 3]\}$ (a search range similar to [5]). Unlike the conventional optical flow methods [18, 15], the search ranges for motion vectors (u, v) here were set the same as the image size to capture possibly large location changes of objects across images/scenes. The truncation parameter t in (1) was empirically set to 10. The filtering kernel radius r was set to 9. The SLIC segment number K increases sublinearly with the image size, e.g., $K = 500$ for 640×480 images, which is not so critical. Our runtime was measured on a 2.9GHz Intel Core i5 CPU with 8GB memory, using a single core.

We evaluate our method over a range of dense matching tasks and use the existing datasets and metrics. Like [5, 6, 21], we focus on challenging pair-wise matching cases.

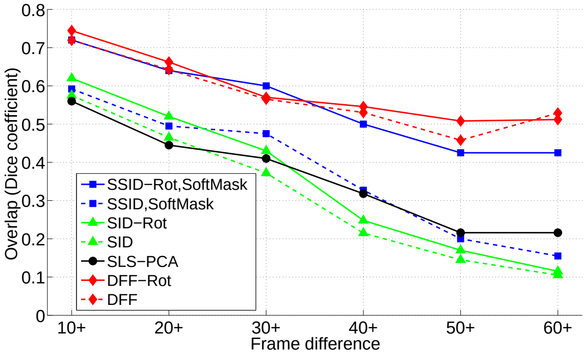

Figure 6. Quantitative overlap results on the Moseg dataset [4]. Average results are reported for different "n+" cases, i.e., using all the test pairs of n or more frames away from the source image $\mathbf{I}$. SIFT Flow and DSP with inferior results are not shown here.

Results on the dataset of Mikolajczyk *et al.* **[16].** We first compare our DFF method with NRDC [5], SIFT Flow [12] and DSP [8] on the standard dataset [16]. We adopt the same evaluation method in [12, 5], i.e., the estimated correspondences that fall within 15 pixels from the ground truth location in the image $\mathbf{I}'$ are considered correct. Fig. 5 shows the general performance on three kinds of test cases: sharpness difference, planar scale and rotation changes, and viewpoint difference. Though our method is designed more for robustly matching images with significant non-rigid motions or of different scene contents, DFF shows its consistent and versatile performance on this dataset. It outperforms NRDC [5] on the sharpness changes, and SIFT Flow [12] and DSP [8] on the geometric changes.

Results on the Moseg dataset [4]. Following the recent segmentation-aware SID (SSID) work [21], we test our approach on this dataset that contains 31 challenging outdoor image pairs with large-displacement, multi-layered motion. Based on the evaluation protocol in [21], we use the estimated flow to warp the segmentation mask from the image $\mathbf{I}'$ to the source image $\mathbf{I}$, and measure the overlap with the ground truth using the Dice coefficient[1]. Fig. 6 shows the overlap results obtained by state-of-the-art methods using SIFT Flow [12] for flow estimation: SID [9], SLS [6] and SSID [21]. SSID uses soft segmentation over SID to suppress the information likely coming from different objects. As in [21], the postfix "-Rot" indicates the rotation-invariant capability is turned *off* in the methods, since foreground objects do not contain many rotations. Both our DFF-Rot and DFF methods consistently outperform the state-of-the-art methods, and they perform also better than the closest competitor *SSID-Rot, SoftMask* particularly for challenging pairs of large frame displacements. Though *SSID-Rot, SoftMask* clearly outperforms *SSID, SoftMask* for this specific

[1] As [21], it is computed as $2 * |A \cap B| / (|A| + |B|)$ for two maps A, B.

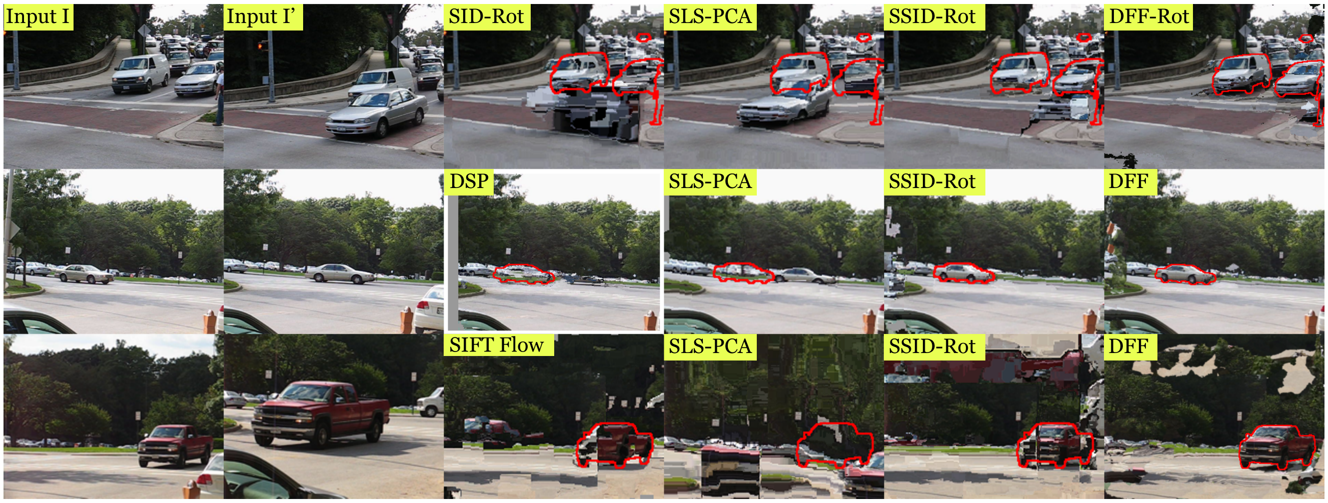

Figure 7. Visual results of warping the image $\mathbf{I}'$ to $\mathbf{I}$ on the Moseg pairs [4] of large displacement and scale changes. As [21], we overlay the ground truth segmentation masks of the image $\mathbf{I}$ onto the warped images in red, facilitating the object-mask alignment inspection.

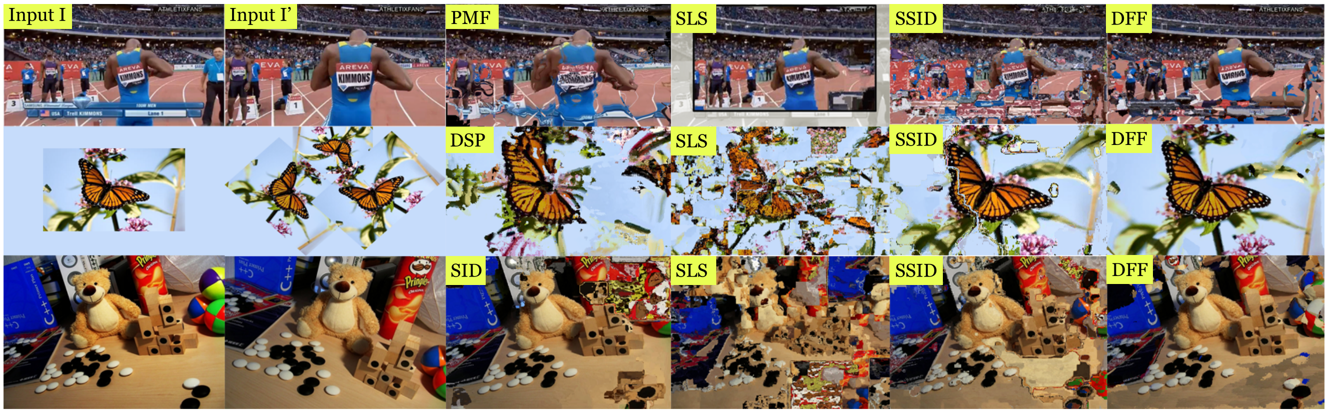

Figure 8. Image alignment results for the images of similar scene content but with significant geometric variations. Each algorithm warps the image $\mathbf{I}'$ onto the image $\mathbf{I}$ based on its estimated flow. The first test image pair was also used in [6], and SLS [6] cropped the area of high confidence matches to show. The last test image pair depicting a wide-baseline stereo case was used in [20]. Unlike [20], the epipolar geometry between the images is not assumed to be known here for a general image alignment task. (All figures are best viewed in color.)

dataset, we find it gives much worse results for challenging test images in Fig. 8. In addition, reliable unsupervised soft image segmentation may still remain as a challenge. Visual results of representative leading methods are given in Fig. 7.

Visual comparison on other challenging pairs. Now we present a visual comparison of our DFF method with other competing methods on the image pairs with significant non-rigid motion or geometric variations (Fig. 8), as well as images of different scenes in different scales or orientations (Fig. 9). Fig. 8 shows that DFF produces much more accurate dense warping results than PMF [15], SLS [6], and DSP [8]. Our enlarged label search space allows dense DAISY descriptors to choose the best scale and orientation parameters at a pixel level for robust matching. In contrast, SLS (and also DSP) is not rotation-invariant by design, and fails to do a good job for the last two cases. With the built-in scale and rotation invariance, SSID [21] and its base SID [9] achieve quite competitive results. However, they tend to generate more gross warping artifacts (also seen in Fig. 7) likely due to the C2F regularization scheme of SIFT Flow [12]. Fig. 9 shows the warping results for different scenes by transferring the pixel colors of the estimated correspondences in $\mathbf{I}'$ to reconstruct $\mathbf{I}$. A good result shall have the appearance of $\mathbf{I}'$ in the scale, pose and scene structure of the image $\mathbf{I}$ [12, 6]. Both of the test scenes confirm that our DFF method gives more coherent alignment results with the image structures of $\mathbf{I}$ much better reconstructed.

Complexity. For all the test cases, our DFF algorithm based on randomized search has often converged after 12–25 iterations. For 320×240 images, the average runtime of DFF is 20–38 seconds, compared to state-of-the-art methods: 235 seconds for SSID (“SoftMask” embedding) [21], 204 seconds for SID [9], and ~ 60 minutes for SLS [6] measured on our PC. Further, SID and SSID are memory-

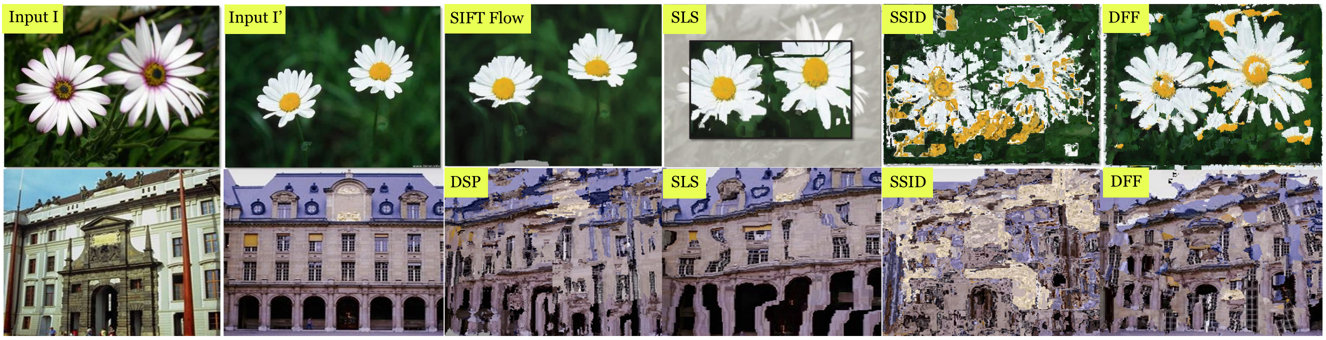

Figure 9. Scene alignment results for the images from different scenes also with drastic appearance differences.

demanding, and require ∼6GB memory for 320×240 images (our 8GB memory insufficient for 640×480 images). Though DFF conducts dense search in an enlarged label space, three reasons make it computationally appealing: 1) the small DAISY descriptor size of 136, compared to 3328 for SID and 8256 for SLS (528 for its PCA variant) [21], 2) precomputed DAISY information allowing for fast online computation (The DAISY related data precomputation consumes only about 6% of the overall complexity and 70 MB memory), and 3) efficient filter-based inference integrated with a generalized PatchMatch label search scheme.

5. Conclusion

We presented a dense DAISY descriptors-based image and scene matching framework. The proposed DFF method demonstrated its robustness in establishing dense correspondences between challenging image pairs in presence of significant variance in geometric and photometric transformation (e.g. scale, rotation, wide baseline, large and nonrigid motions, illumination changes, image quality) and also across different scene contents. Though a considerably enlarged label space is searched, our DFF method achieves clear runtime and memory advantages while assuring image matching quality. We will make our code publicly available.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. PAMI*, 34(11), 2012. 5, 6

[2] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *Proc. of ECCV*, 2010. 2, 3, 4, 5, 6

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Speeded-up robust features. In *Proc. of ECCV*, 2006. 2

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 6, 7

[5] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. In *SIGGRAGH*, 2011. 1, 2, 3, 4, 6

[6] T. Hassner, V. Mayzels, and L. Zelnik-Manor. On SIFTs and their scales. In *Proc. of CVPR*, 2012. 2, 3, 6, 7

[7] K. He, J. Sun, and X. Tang. Guided image filtering. In *Proc. of ECCV*, 2010. 4

[8] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013. 2, 6, 7

[9] I. Kokkinos and A. Yuille. Scale invariance without scale selection. In *CVPR*, 2008. 2, 6, 7

[10] W.-Y. Lin, L. Liu, Y. Matsushita, K.-L. Low, and S. Liu. Aligning images in the wild. In *CVPR*, 2012. 2

[11] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE Trans. PAMI*, 2011. 3

[12] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. PAMI*, 33(5), 2011. 1, 2, 3, 4, 6, 7

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2

[14] J. Lu, K. Shi, D. Min, L. Lin, and M. N. Do. Cross-based local multipoint filtering. In *CVPR*, 2012. 4

[15] J. Lu, H. Yang, D. Min, and M. N. Do. Patchmatch filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *CVPR*, 2013. 2, 3, 4, 5, 6, 7

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, Nov. 2005. 6

[17] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. Bilateral filtering: Theory and applications. *Foundations and Trends in Comp. Graphics and Vision*, 4(1):1–73, 2008. 4

[18] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, 2011. 3, 4, 6

[19] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal of Computer Vision*, 47:7–42, 2002. 1

[20] E. Tola, V. Lepetit, and P. Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. PAMI*, 32(5):815–830, 2010. 2, 3, 4, 7

[21] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. M. Noguer. Dense segmentation-aware descriptors. In *CVPR*, 2013. 2, 6, 7, 8

[22] V. Vineet, J. Warrell, and P. H. S. Torr. Filter-based mean-field inference for random fields with high-order terms and product label-spaces. In *Proc. of ECCV*, 2012. 3, 4

[23] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Trans. PAMI*, 2012. 1, 3, 5