2016 IEEE Conference on Computer Vision and Pattern Recognition

# Progressive Feature Matching with Alternate Descriptor Selection and Correspondence Enrichment

Yuan-Ting Hu Yen-Yu Lin

Academia Sinica, Taiwan

## Abstract

*We address two difficulties in establishing an accurate system for image matching. First, image matching relies on the descriptor for feature extraction, but the optimal descriptor often varies from image to image, or even patch to patch. Second, conventional matching approaches carry out geometric checking on a small set of correspondence candidates due to the concern of efficiency. It may result in restricted performance in recall. We aim at tackling the two issues by integrating adaptive descriptor selection and progressive candidate enrichment into image matching. We consider that the two integrated components are complementary: The high-quality matching yielded by adaptively selected descriptors helps in exploring more plausible candidates, while the enriched candidate set serves as a better reference for descriptor selection. It motivates us to formulate image matching as a joint optimization problem, in which adaptive descriptor selection and progressive correspondence enrichment are alternately conducted. Our approach is comprehensively evaluated and compared with the state-of-the-art approaches on two benchmarks. The promising results manifest its effectiveness.*

## 1. Introduction

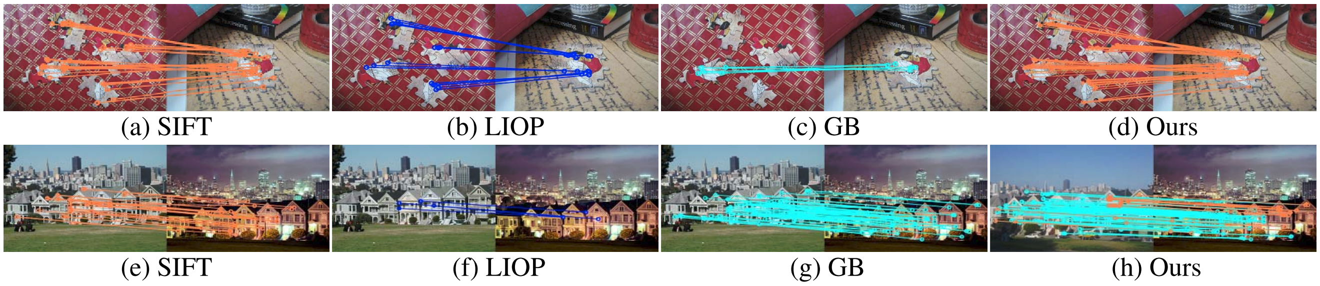
Image matching aims to seek the correspondences of common regions across images. It is an active and fundamental research topic in computer vision, since it has been an inherent part in a broad set of vision applications, such as panoramic stitching [6], common pattern discovery [27], object recognition [16, 28], image retrieval [31] and 3D reconstruction [1, 32]. A predominant paradigm of image feature matching, *e.g.*, [28] involves three steps: 1) detecting feature points and characterizing the detected points with a chosen descriptor, 2) establishing a reduced set of correspondence candidates, and 3) removing outliers from the reduced set by referring to both photometric and geometric consistency to get the final matching results.

Despite the popularity, there are still two main obstacles preventing us from getting satisfactory results for feature matching. First, most matching algorithms choose a specific descriptor. However, existing descriptors are designed with the trade-off between *distinctiveness* and *invariance*. The effectiveness of a descriptor depends on not only intra-image appearance but also inter-image variation. Thus, the optimal descriptor for matching is often *image-dependent* or even *region-dependent*. Most matching methods do not consider this issue. Second, geometric checking for outlier removal is widely adopted to enhance feature matching, but it is of a high computational complexity. The trade-off between *accuracy* and *efficiency* results in a compromising mechanism. Namely, geometric checking is applied only to a small, putative set of correspondence candidates. It degrades the performance in recall.

We aim at address the two aforementioned issues and design an algorithm that adaptively and efficiently selects good descriptors. We observe that descriptor selection and candidate enrichment are complementary to each other. High-quality matching results by adaptively selected descriptors reveal the transformations, which give a more authentic guidance on candidate enrichment. On the other hand, the enriched candidate set serves as a better reference for descriptor selection. This observation motivated us to cast image matching as a joint optimization problem in which descriptor selection and candidate enrichment are alternately carried out. Specifically, it is formulated as a labeling problem over a graph structure, and can be iteratively optimized by using only *graph cut* [5]. It turns out that our approach can produce matching results with high quality by leveraging multiple descriptors, and does not compromise in running time owing to dynamic candidate enrichment.

As an illustration, Figure 1 shows the matching results on two pairs of images, `jigsaws` and `painted ladies`, by using three different descriptors, *SIFT* [28], *LIOP* [37] and *GB* [3], and our approach. For each feature point that has corresponding point in the opposite image, we seek its match by the nearest neighbor search with one of the

1063-6919/16 $31.00 © 2016 IEEE
DOI 10.1109/CVPR.2016.44


(a) SIFT (b) LIOP (c) GB (d) Ours

(e) SIFT (f) LIOP (g) GB (h) Ours

Figure 1. Feature matching on two image pairs, (a) ∼ (d) `jigsaws` and (e) ∼ (h) `painted ladies`, by three different descriptors, SIFT [28], LIOP [37] and GB [3], and our approach. Only correct matches are drawn with colors corresponding to the used descriptors.

three descriptors, and draw the match if it is correct. SIFT shows good results owing to the highly textured patterns in `jigsaws`, while the shape-based descriptor GB gives better performance due to the strong coherence in shape in `painted ladies` and fails to match `jigsaws` due to the cluttered backgrounds and large view point change. This example points out that the optimal descriptor varies from image to image. In contrast, our method yields more correct matchings by adaptively selecting a descriptor for matching each point, like those shown in Figure 1(d) and 1(h).

To sum up, we integrate adaptive descriptor selection and progressive candidate enrichment into the process of image matching, and cast it as a graph optimization problem. The proposed approach is comprehensively evaluated on two benchmarks of image matching, including SNU dataset [12] and SYM dataset [18]. The results demonstrate that our approach can effectively recommend few but accurate correspondence candidates and select a proper descriptor for matching each feature point, thus resulting in a remarkable performance gain.

## 2. Related work

In this section, we briefly review a few relevant topics.

### 2.1. Image matching with geometric checking

Geometric checking addresses the ambiguity arisen in matching by detecting *outliers*, correspondences with inconsistent transformations here. *Voting-based* approaches, *e.g.*, [2, 7, 8, 17], are popular for their simplicity and efficiency. *RANSAC* [17] is a representative method of this class. It estimates the underlying transformation and removes outliers simultaneously. Chen *et al.* [7] and Avrithis and Tolias [2] performed *Hough voting* in the transformation space for identifying correct matches. *Clustering-based* approaches, such as [9, 40], can introduce extra constraints to aggregate consistent clusters of matches, and show their effectiveness in unconstrained matching cases. However, the values of parameters for clustering, such as the number of clusters and the thresholds for merging or splitting clusters, vary from case to case, and are difficult to set in advance.

*Graph-based* approaches are another popular branch of geometric checking. Methods, *e.g.*, [10, 14, 26, 41], model the coherence between potential matches by employing a graph structure and geometric checking is accomplished by graph partition. However, graph partition is an NP hard problem, and is usually solved with the continuous relaxation. Torresani *et al.* [34] efficiently solve the graph matching problem by breaking it into subproblems. Liu and Yan [27] handle multi-object matching via discovering strongly connected subgraphs. In the formulation of these methods, a vertex on the graph corresponds to a correspondence. Thus, the complexity increases when the number of correspondences becomes large. Our approach belongs to the graph-based branch. However, unlike most approaches, *e.g.*, [10, 14, 26, 27, 34], of this category, the vertices in our approach are associated with feature points instead of matches, so our approach scales better in a large set of match candidates. Furthermore, our approach explores the locality of spatial dependency. Thus, it can match multiple objects, and has a sparse graph and hence is more efficient.

### 2.2. Correspondence enrichment

Due to the high complexity, geometric checking is usually applied to a reduced set of match candidates, thus leading to low recall. Correspondence enrichment alleviates this unfavorable effect. Ferrari *et al.* [15] duplicated matches to the surrounding areas with similar appearance to expand the candidate set. Chen *et al.* [7] investigated *boundary preserving local regions* [21], and increased reliable matchings inside the regions. Cho and Lee [11] proposed a Bayesian framework for re-estimating a new reduced set, and improved graph matching results. Wang *et al.* [36] presented a progressive mode-seeking approach that efficiently explores the huge matching space through density sampling guided by a smaller, confident set. The common issue of approaches to correspondence enrichment is their sensitivity to the quality of the initial matches, because the enriched correspondences are biased towards the initial set. Our approach performs unsupervised descriptor selection, and can compile a better initial set. Besides, we recommend candidates that are either consistent with or complementary to the initial set, and enhance the diversity of the candidate set.

### 2.3. Multiple descriptor fusion

Different descriptors capture diverse visual evidences. Research efforts have been made on descriptor fusion for performance improvement. A number of studies such as [4, 30, 35, 39] have demonstrated the effectiveness of using multiple descriptors for image matching and classification. These approaches use fixed weights for descriptor fusion, and neglect that optimal features for image description are different from image to image. To address this issue, adaptive feature fusion has been carried out in recent studies such as [22, 24, 38, 20]. Xu *et al.* [38] fused gradient and color data models by an adaptive selection. Kim *et al.* [22] proposed a locally varying data term where multiple data models are merged based on their discriminant powers in the surrounding area. However, features in diverse descriptors are typically of different dimensions and with different scales of statistics. Fusion by directly combining the resulting features may be infeasible. Hu *et al.* [20] carried out descriptor selection in the homography space. However, their method trains one-class SVM by taking all correspondence candidates as input, and may be less efficient. Lempitsky *et al.* [23] and Hsu *et al.* [19] adaptively fuse multiple flow proposals and compile the final flow map. These methods are designed to optimize over *dense* flow proposals, and cannot be applied to *sparse* feature matching.

## 3. Problem statement

Given two images $I^P$ and $I^Q$ with detected feature points $U^P = \{u_i^P\}_{i=1}^{N^P}$ and $U^Q = \{u_i^Q\}_{i=1}^{N^Q}$, we aim at finding the corresponding point in $U^Q$ for each $u_i^P \in U^P$, if it exists. In this work, *Hessian-Affine* [29] detector is used for its efficiency and high repeatability. Thus, the support region of each feature point is an ellipse. Multiple descriptors are applied to each feature point $u_i \in U^P \cup U^Q$. The yielded feature vectors of $u_i$ are denoted by $\{\mathbf{x}_{i,m}\}_{m=1}^{M}$, where $M$ is the number of the adopted descriptors. For each $u_i^P \in U^P$, we compile the set of its most plausible $R$ matched points in $I^Q$, $\mathcal{C}_{i,m} = \{u_{i_{r,m}}^Q\}_{r=1}^{R}$, with descriptor $m$ and distance measure $\|\mathbf{x}_{i,m}^P - \mathbf{x}_{j,m}^Q\|$. After repeating this process for each feature point in image $I^P$, the set of correspondence candidates $\mathcal{C}$ is constructed:

$$\mathcal{C} = \bigcup_{i=1}^{N^P} \mathcal{C}_i, \text{ where } \mathcal{C}_i = \bigcup_{m=1}^{M} \mathcal{C}_{i,m}. \tag{1}$$

$\mathcal{C}$ contains at most $N^P \times R \times M$ correspondences after removing the duplicates. When $R$ is set to $N^Q$, $\mathcal{C}$ covers all the possible matches. However, it becomes too large to be efficiently dealt with, especially for geometric checking. In this work, we set $R = 1$. The resulting $\mathcal{C}$ acts as the initial match set, and will be gradually enriched.

## 4. The proposed approach

Our approach formulates the task of image matching as an energy minimization problem on a graph. In this section, we introduce the graph structure, energy function, optimization process and implementation details of our approach.

### 4.1. Graph construction

We construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In $\mathcal{G}$, each vertex $v_i \in \mathcal{V}$ corresponds to feature point $u_i^P$ in image $I^P$, and the number of vertices $|\mathcal{V}|$ is $N^P$. The edge $e_{ij} \in \mathcal{E}$ is added to link $v_i$ and $v_j$ if $u_j^P$ is one of the spatially $k$ nearest neighbors of $u_i^P$. Each vertex $v_i$ is associated with a composite variable $\ell_i = [s_i, t_i]$, which represents that $u_i^P$ is matched to $u_{s_i}^Q$ by using selected descriptor $t_i$. Therefore, the domain of $\ell_i$ is $\mathcal{L} = \mathcal{S} \times \mathcal{T}$, where $\mathcal{S} = \{1, 2, ..., N^Q\}$ and $\mathcal{T} = \{1, 2, ..., M\}$.

By the constructed graph $\mathcal{G}$, the task of image feature matching in this work becomes a graph labeling problem. Specifically for matching images $I^P$ and $I^Q$, it is cast as seeking a plausible labeling $\boldsymbol{\ell} = [\ell_1 \ ... \ \ell_i \ ... \ \ell_{N^P}]$, which specifies which corresponding point in $I^Q$ is and which descriptor is selected for matching each feature point $u_i^P$ in $I^P$. For the ease of explanation, we similarly define $\mathbf{s}$ and $\mathbf{t}$ as $\mathbf{s} = [s_1 \ ... \ s_{N^P}]$ and $\mathbf{t} = [t_1 \ ... \ t_{N^P}]$, respectively.

### 4.2. Energy Function

For boosting the performance in both accuracy and efficiency, we incorporate adaptive descriptor selection, geometric checking, and correspondence enrichment into image matching. To that end, we seek a good labeling $\boldsymbol{\ell} = [\ell_1 \ ... \ \ell_{N^P}]$ by minimizing the following energy function

$$\begin{aligned} J(\boldsymbol{\ell}) = &\sum_{v_i \in \mathcal{V}} D_p^i(\ell_i, \mathcal{C}_i) + \lambda_1 \sum_{v_i \in \mathcal{V}} D_d^i(\ell_i) \\ &+ \lambda_2 \sum_{e_{ij} \in \mathcal{E}} V_g^{ij}(\ell_i, \ell_j) + \lambda_3 \sum_{e_{ij} \in \mathcal{E}} V_s^{ij}(\ell_i, \ell_j), \end{aligned} \tag{2}$$

where $\lambda_1$, $\lambda_2$, and $\lambda_3$ are three non-negative constants. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the constructed graph. $\mathcal{C}_i$ in Eq. (1) is set of the initial match candidates for feature point $u_i^P$, and will be gradually expanded during matching.

There are four terms introduced in the designed energy function Eq. (2), including two data terms, $D_p^i$ and $D_d^i$, and two pairwise terms, $V_g^{ij}$ and $V_d^{ij}$. For data terms, $D_p^i$ considers the *p*hotometric similarity of the matched points, while $D_d^i$ estimates the *d*iscriminant power of the selected descriptor in the surrounding region. For pairwise terms, $V_g^{ij}$ ensures the *g*eometric consistence between neighboring correspondences, while $V_s^{ij}$ enforces the *s*moothness of the selected descriptors. The definitions of the four terms and their justification are given in the following.

#### 4.2.1 Data term $D_p^i$

Set $\mathcal{C}_i$ covers the matched candidates of feature point $u_i^P$. The variable $\ell_i = [s_i, t_i]$ specifies the match $(u_i^P, u_{s_i}^Q)$ and the selected descriptor $t_i$ for $u_i^P$. We take the photometric consensus into account, and define data term $D_p$ as

$$D_p^i(\ell_i, \mathcal{C}_i) = \begin{cases} \infty, & \text{if } u_{s_i}^Q \notin \mathcal{C}_i, \\ \frac{f(i,s_i,t_i)}{\max(f(i,i^*,t_i),\epsilon)}, & \text{otherwise,} \end{cases} \tag{3}$$

where $\epsilon$ is a small constant used to avoid the problem of dividing by zero. The photometric dissimilarity function $f(a,b,c)$ between $u_a^P$ and $u_b^Q$ under descriptor $c$ is given as

$$f(a,b,c) = dist(\mathbf{x}_{a,c}^P, \mathbf{x}_{b,c}^Q), \tag{4}$$

where $dist(x_i, x_j)$ is the distance between $x_i$ and $x_j$ and we use Euclidean distance in all the experiments. $i^*$ is index of the most matched point of $u_i^P$, *i.e.*,

$$i^* = \arg\min_{j \in \mathcal{S}} f(i,j,t_i). \tag{5}$$

Data term $D_p^i$ in Eq. (3) excludes the matches that do not belong to the candidate set by setting their energy as infinity. For candidates, we consider *normalized* photometric dissimilarity via dividing by the distance to its the nearest neighbor. The normalized dissimilarity measure alleviates the variety of descriptors, and allows us to conduct cross-descriptor comparisons. In brief, the larger the value is, the less possible $u_i^P$ matches to $u_{s_i}^Q$ with descriptor $t_i$.

#### 4.2.2 Data term $D_d^i$

While $D_p^i$ measures the appearance dissimilarity between matched points with a descriptor, $D_d^i$ estimates the discriminant power of that descriptor at the feature point. Specifically, it is defined as

$$D_d^i(\ell_i) = \frac{f(i,s_i,t_i)}{\frac{1}{k}\sum_{e_{ij}\in\mathcal{E}} f(j,s_i,t_i)}, \tag{6}$$

where $k$ is the number of the spatial neighbors of $u_i^P$, and distance $f$ is given in Eq. (4).

The idea behind Eq. (6) is that descriptor $t_i$ is considered effective at $u_i^P$ if it is discriminant enough within the neighborhood of $u_i^P$. Otherwise, it introduces unfavorable ambiguity in matching, and is no longer an effective descriptor. Thus, this term penalizes descriptors that cannot distinguish the correspondence $(u_i^P, u_{s_i}^Q)$ from its neighboring correspondence $(u_j^P, u_{s_i}^Q)$.

#### 4.2.3 Pairwise term $V_g^{ij}$

For two neighboring points $u_i^P$ and $u_j^P$, *i.e.*, $e_{ij} \in \mathcal{E}$, their correspondences $c_i = (u_i^P, u_{s_i}^Q)$ and $c_j = (u_j^P, u_{s_j}^Q)$ are obtained by referring to $\ell_i$ and $\ell_j$, respectively. Since the support region of each feature point is an ellipse in this work, the affine transformation with 6 degrees of freedom, $T_i$, can be inferred for correspondence $c_i$, and reveals the geometric evidence around $u_i^P$. Similarly, $T_j$ is inferred for $c_j$.

Pairwise term $V_g^{ij}$ is developed upon the observation that nearby feature points usually reside in the same object, and hence undergo similar transformations. We hence prefer a geometrically smooth matching field by defining $V_g^{ij}$ as

$$V_g^{ij}(\ell_i, \ell_j) = 1 - \exp\left(-f_{rep}(c_i, c_j, T_i, T_j)/\sigma\right), \tag{7}$$

where $\sigma$ is a positive constant, and $f_{rep}$ is the *reprojection error* [13], which measures the geometric inconsistency between two correspondences $c_i$ and $c_j$ with respectively associated affine transformations $T_j$ and $T_i$, *i.e.*,

$$f_{rep}(c_i, c_j, T_i, T_j) = (d_{c_i|T_j} + d_{c_j|T_i})/2, \tag{8}$$

where

$$d_{c_i|T_j} = (\|T_j(u_i^P) - u_{s_i}^Q\| + \|T_j^{-1}(u_{s_i}^Q) - u_i^P\|)/2, \tag{9}$$

and $d_{c_j|T_i}$ is similarly defined.

#### 4.2.4 Pairwise term $V_s^{ij}$

The effective descriptors for two neighboring feature points on the same object are usually the same due to repeatedly appeared patterns within the object. Take Figure 1 for an example. SIFT descriptor is consistently better in the whole image `jigsaws`, and GB descriptor is consistently better for the second image `painted ladies`. This observation can serve as a good prior for descriptor selection, and alleviates the effect caused by noises. To this end, the pairwise term $V_s^{ij}$ is given by

$$V_s^{ij}(\ell_i, \ell_j) = \begin{cases} 0, & \text{if } t_i = t_j, \\ 1, & \text{otherwise,} \end{cases}. \tag{10}$$

It encourages the smoothness of the selected descriptors.

### 4.3. Optimization

Directly solving Eq. (2) to optimize the labeling $\boldsymbol{\ell}$ is feasible by using existing solvers, such as graph cut. Nevertheless, we want to further speed up the optimization, and carry out progressive enrichment of match candidates. To this end, we divide labeling $\boldsymbol{\ell}$ into two parts, $\mathbf{s}$ and $\mathbf{t}$. An iterative, alternate strategy is adopted to optimize $\mathbf{s}$ and $\mathbf{t}$ and enrich $\mathcal{C} = \{\mathcal{C}_i\}_{i=1}^{N^P}$ in Eq. (1). At each iteration, one of the three variables is optimized or enriched while keeping the others fixed, and then their roles are switched sequentially. Iterations are repeated until convergence or the maximum number of iterations is reached.

#### 4.3.1 On optimizing $\mathbf{t}$

The pairwise term $V_g^{ij}$ in Eq. (7) is irrelevant to $\mathbf{t}$. By fixing $\mathbf{s}$ and $\mathcal{C}$, the optimization problem in Eq. (2) becomes

$$\begin{aligned} J(\mathbf{t}) = {} & \sum_{v_i\in\mathcal{V}} D_p^i([s_i,t_i],\mathcal{C}_i) + \lambda_1 \sum_{v_i\in\mathcal{V}} D_d^i([s_i,t_i]) \\ & + \lambda_3 \sum_{e_{ij}\in\mathcal{E}} V_s^{ij}([s_i,t_i],[s_j,t_j]). \end{aligned} \tag{11}$$

The first two terms in the right-hand side of Eq. (11) jointly yield the new data term, in which the cost of assigning $t_i$ to vertex $v_i$ is computable. Thus, we efficiently optimize $\mathbf{t}$ in Eq. (11) by using graph cut [5].

#### 4.3.2 On optimizing $\mathbf{s}$

By fixing $\mathbf{t}$ and $\mathcal{C}$, the optimization problem in Eq. (2) is similarly reduced to

$$\begin{aligned} J(\mathbf{s}) = {} & \sum_{v_i\in\mathcal{V}} D_p^i([s_i,t_i],\mathcal{C}_i) + \lambda_1 \sum_{v_i\in\mathcal{V}} D_d^i([s_i,t_i]) \\ & + \lambda_2 \sum_{e_{ij}\in\mathcal{E}} V_g^{ij}([s_i,t_i],[s_j,t_j]), \end{aligned} \tag{12}$$

and can also be solved via graph cut [5].

#### 4.3.3 On enriching $\mathcal{C}$

For each feature point $u_i^P$, the candidate set $\mathcal{C}_i$ is gradually enriched to improve recall. We have tried several ways for enriching $\mathcal{C}_i$, and found that two diverse ways jointly work best in our implementation. Namely, we seek two of the $k$ nearest neighbors of $u_i^P$ as references, and each reference recommends $u_i^P$ an additional match. The first reference, denoted by $u_{i_1^*}^P$, is the most *concerted* neighbor, *i.e.*, the one that contributes the least energy in Eq. (2). It is most likely to be matched correctly. The other, denoted by $u_{i_2^*}^P$, is the one with the strongest geometrically inconsistency with $u_i^P$. It brings the complementary information. Specifically, the two references are defined as

$$\begin{aligned} i_1^* = {} & \operatorname*{arg\,min}_{\{j|e_{ij}\in\mathcal{E}\}} D_p^j(\ell_j,\mathcal{C}_j) + \lambda_1 D_d^j(\ell_j) \\ & + \sum_{\{n|e_{jn}\in\mathcal{E}\}} \big(\lambda_2 V_g^{jn}(\ell_j,\ell_n) + \lambda_3 V_s^{jn}(\ell_j,\ell_n)\big), \end{aligned} \tag{13}$$

and

$$i_2^* = \operatorname*{arg\,max}_{\{j|e_{ij}\in\mathcal{E}\}} V_g^{ij}(\ell_i,\ell_j). \tag{14}$$

We apply the affine transformations of the two reference neighbors to $u_i^P$, seek the closest feature points in image $I^Q$, and add the sought feature points to $\mathcal{C}_i$.

```
Algorithm 1: The proposed approach
Input: Two sets of detected features $U^P$ and $U^Q$,
       Max iteration $T$;
Output: The labeling $\boldsymbol{\ell} = (\mathbf{s}, \mathbf{t})$;
Initialize $\boldsymbol{\ell}$; (Section 4.4);
Construct the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; (Section 4.1) ;
Iteration $\longleftarrow 1$ ;
while Iteration $< T$ && not converge do
    if Iteration $! = 1$ then
        $\forall v_i \in \mathcal{V}$, enrich $\mathcal{C}_i$. (Section 4.3.3);
    Optimize $\mathbf{t}$ by solving Eq. (11). (Section 4.3.1);
    Optimize $\mathbf{s}$ by solving Eq. (12). (Section 4.3.2);
    Iteration $\longleftarrow$ Iteration $+ 1$;
```

## 4.4. Implementation details

We initialize the label $\ell_i = (s_i, t_i)$ for each vertex $v_i$ by calculating the ratio between the distance to its nearest neighbor and to its 2nd nearest neighbor by each descriptor, and get $M$ ratios. Then we initialize $t_i$ as the descriptor with the smallest ratio, and $s_i$ as the nearest neighbor measured by descriptor $t_i$.

There are five parameters in our approach, including neighborhood size $k$ in Section 4.1, leading coefficients $\lambda_1$, $\lambda_2$, and $\lambda_3$ in Eq. (2), and hyperparameter $\sigma$ in Eq. (7). They are tuned and fixed for each adopted dataset. To conclude this section, we summarize our approach in Algorithm 1.

Using multiple descriptors helps improve the performance, but extracting multiple descriptors increases the computational cost. There exists a trade-off between matching accuracy and efficiency while our approach offers a flexible and practical framework for both single-descriptor and multiple-descriptor matching.

## 5. Experimental results

A comprehensive study of our approach is presented in the section. First, we introduce the experimental setup and the evaluation metrics. Then, three sets of experiments are conducted, including the comparisons with the state-of-the-art approaches on two benchmarks, the effect of enrichment, and the advantages of employing multiple descriptors.

### 5.1. Experimental setup

SNU dataset [12] and SYM dataset [18] are used in our experiments. They consist of 6 and 46 pairs of images to be matched, respectively. Various challenges such as multiple object matching with cluttered backgrounds in SNU dataset and variations in lighting conditions (day/night), ages (old/nowadays scene) and rendering styles (photograph/drawing) in SYM dataset make the two datasets a

Table 1. Performance in mAP (%) on SNU dataset [12].

| Descriptor | SIFT | LIOP | DAISY | RI | GB | ALL |
|---|---|---|---|---|---|---|
| SM [25] | 55.30 | 38.74 | 46.71 | 34.57 | 12.13 | 55.72 |
| ACC [9] | 60.28 | 29.83 | 36.49 | 15.10 | 8.88 | 59.59 |
| HV [7] | 60.12 | 43.97 | 50.06 | 37.14 | 12.08 | 71.14 |
| PGM [11] | 65.66 | 48.33 | 59.37 | 39.19 | 12.71 | 67.55 |
| CONCAT. | | | | | | 39.70 |
| Ours (w/o enrich.) | 69.70 | 48.52 | 55.47 | 23.53 | 9.84 | 76.75 |
| Ours (w/ enrich.) | 72.29 | 58.31 | 64.12 | 38.66 | 12.59 | **81.81** |

Table 2. Performance in mAP (%) on SYM dataset [18].

| Descriptor | SIFT | LIOP | DAISY | RI | GB | ALL |
|---|---|---|---|---|---|---|
| SM [25] | 18.92 | 16.79 | 22.72 | 7.99 | 32.57 | 40.22 |
| ACC [9] | 26.74 | 19.49 | 29.97 | 10.70 | 29.28 | 42.36 |
| HV [7] | 22.21 | 18.69 | 26.92 | 11.88 | 38.28 | 43.79 |
| PGM [11] | 29.46 | 22.36 | 35.01 | 15.49 | 47.53 | 48.01 |
| CONCAT. | | | | | | 13.36 |
| Ours (w/o enrich.) | 26.53 | 20.93 | 30.30 | 11.67 | 41.70 | 46.67 |
| Ours (w/ enrich.) | 30.04 | 27.27 | 32.63 | 15.84 | 41.00 | **49.35** |

good test bed for performance evaluation. We use Hessian-affine detector [29] for its high repeatability and we adopt five complementary descriptors for capturing diverse visual cues, including SIFT [28], LIOP [37], DAISY [33], raw intensities (RI) and geometric blur (GB) [3]. The RI descriptor extracts the grey-level pixel intensities of the feature regions in a raster scan order. Four state-of-the-art approaches are adopted for comparison, including SM [25] (a graph-based method), ACC [9] (a clustering-based approach), HV [7] (a voting-based method) and PGM [11] (a correspondence recommendation framework). The four compared approaches are designed to work with a single descriptor. We can extend them for handling multiple descriptors by concatenating all the initial candidates produced by the five descriptors. For fair comparison, the reprojection error is used as the dissimilarity measure between correspondences. When using a single descriptor, we set $R = 5$ for all approaches, including ours. $R$ is set as 1 in the cases where multiple descriptors are adopted. We also implement the method in [30], CONCAT., which concatenates multiple descriptors to match images. In our implementation, we concatenate all the five adopted descriptors, and apply the method in [7] for geometric verification.

## 5.2. Evaluation metrics

The performance of a matching algorithm is presented in the forms of *precision* and *recall* jointly. The two measures are defined as

$$\text{PRECISION} = \frac{n\text{TP}}{n\text{TP} + n\text{FP}} \quad \text{and} \quad \text{RECALL} = \frac{n\text{TP}}{n\text{P}}, \qquad (15)$$

where $n\text{TP}$ and $n\text{FP}$ is the returned numbers of correspondences which are correctly and wrongly identified by a matching method, respectively, and $n\text{TP} + n\text{FP}$ is the number of total returned correspondences. $n\text{P}$ is the number of points in $I^P$ whose corresponding points exist.

For each matching approach, all the detected correspondences are sorted by its own criterion, such as the density values in HV and the contributed energy values in our approach. Specifically, the energy value for correspondence $c_i = (u_i^P, u_{\ell_i}^Q)$ is defined as

$$D_p^i(\ell_i, \mathcal{C}_i) + \lambda_1 D_d^i(\ell_i) + \sum_{\{n|e_{in} \in \mathcal{E}\}} \Big(\lambda_2 V_g^{in}(\ell_i, \ell_n) + \lambda_3 V_s^{in}(\ell_i, \ell_n)\Big). \qquad (16)$$

By sampling on the sorted lists, we get a set of precisions and recalls and present them with a "1−precision" vs. "recall" curve (PR curve) on an image pair, or *mean average precision* (mAP) and *mean accuracy* (mAccu) on a dataset. The mAP is the mean of the average precision of each image pair in a dataset, while the average precision is calculated as the mean of precisions with different numbers of returned correspondences. Besides, we set the constraint that every matching algorithm can detect at most one corresponding point for each $u_i^P$, so the number of returned correspondences, namely $n\text{TP} + n\text{FP}$, is at most $N^P$. We then term the recall when we set $n\text{TP} + n\text{FP}$ to its largest value as *accuracy* and mean accuracy (mAccu) is the average accuracy of a dataset.

## 5.3. Comparisons on two benchmark datasets

In this set of experiments, we compare our approach with the state-of-the-art approaches on SNU and SYM datasets. The five descriptors, SIFT, LIOP, DAISY, RI, and GB are used in the experiments. Each approach, including SM, ACC, HV, PGM, and ours, is applied to work with the five descriptors individually and jointly. The performance in mAP is reported in Table 1 and Table 2. The best performance by using a single descriptor is underlined and the one with overall best performance is given in bold. Our method with multiple descriptors get 81.81% and 49.35%, the highest performance, on SNU dataset and SYM dataset, respectively. We would like to mention that the performances of directly solving Eq. (2) by graph cut are 81.61% and 48.34%, respectively, which are only slightly different from the performance by the alternate solution. Note that we use the real detector, Hessian-affine detector, so the performance cannot be directly compared to the one reported in [18] with the grid detector, which is a synthetic detector.

As reported in Table 1, SIFT consistently outperforms the other four descriptors with each baselines on SNU dataset. On the contrary, GB gives the best performance on SYM dataset in Table 2. The goodness of descriptors varies from dataset to dataset. Our approach allows cross descriptor verification and adaptive descriptor selection. The results show that our approach effectively fuses the five descriptors and gets superior performance compared to the state-of-the-art approaches on the two datasets

Table 3. Performance of our approach with and without descriptor selection and enrichment, respectively, in mAP (%).

| | SNU | | SYM | |
|---|---|---|---|---|
| | w/o selec. | w/ selec. | w/o selec. | w/ selec. |
| w/o enrich. | 73.16 | 76.75 | 42.99 | 46.67 |
| w/ enrich. | 74.22 | 81.81 | 44.52 | 49.35 |

even when they use multiple descriptors. While these approaches carry out geometric checking, our approach further considers intra-descriptor photometric similarity and the discriminability of each descriptor. The method CONCAT. doesn't work well because simple concatenation of multiple descriptors neglects that the effectiveness of descriptor is image-dependent. It is notable that our approach with a single descriptor remarkably outperforms the baselines on SNU dataset, and is still comparable to PGM, which can recommend plausible correspondences, on SYM dataset with GB. It is because our approach finds the locally smooth correspondences. The local property works well with multiple object matching on SNU dataset, but it is less favorable on SYM dataset, in which the image pair undergoes a global transformation.

In order to evaluate the component of descriptor selection in our approach, we turn off this component by setting $\lambda_1$ and $\lambda_3$ in Eq. (2) to zero and optimizing only label $s_i$ described in Section 4.1. Table 3 compares the performance of our approach and this variant. The results with descriptor selection are much better than those without it, indicating the advantage of using descriptor selection in our approach.

### 5.4. Effect of enrichment

To show the effect of enrichment, we present two sets of experiments where our method is applied with all the five descriptors under two settings: with enrichment and without enrichment. First, we show the results with and without enrichment with different values of parameter $R$. $R$ controls the size of $\mathcal{C}$ in Eq. (1). The results on SNU dataset and SYM dataset are shown in Figure 2 with $x$ axis representing the value of $R$ and $y$ axis standing for mAccu. mAccu measures the portion of correctly matched interest points. It reveals whether we find additional good matches or not in the step of enrichment. With enrichment, we see highly boosted performance. In the second experiment, we compare the results with and without enrichment at each optimization iteration. The results are shown in Figure 3, where $x$ axis represents the iteration number and $y$ axis represents mAccu. The enrichment happens after the first iteration, and we can see the significant performance gain by enrichment on the two datasets. In addition, our approach almost converges after two iterations. Thus, we only alternate the three-step optimization twice for time efficiency.

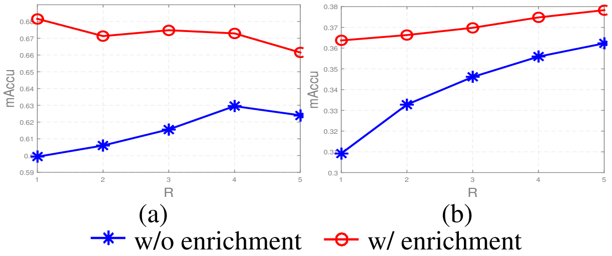


Figure 2. Performance in mAccu with different values of $R$ of our approach on (a) SNU dataset and (b) SYM dataset.

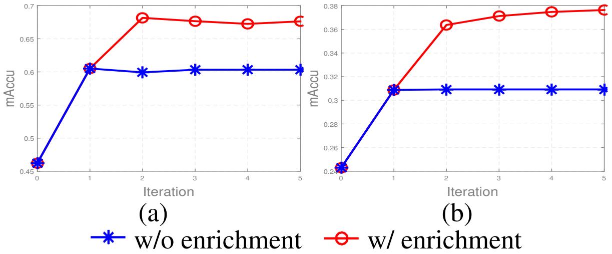


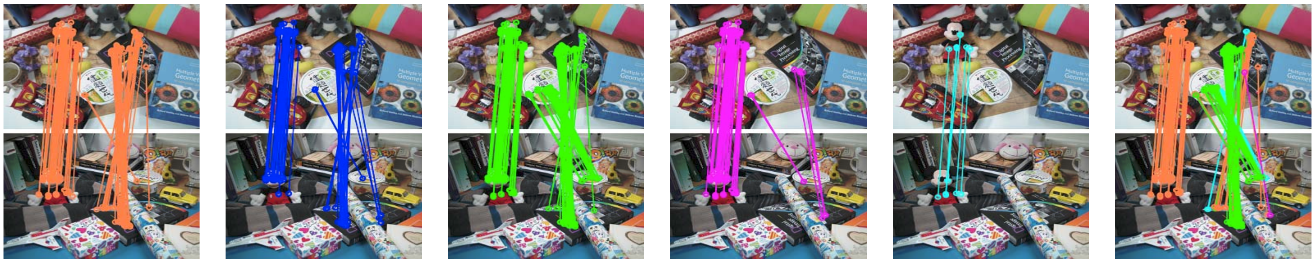
Figure 3. Performance in mAccu along the iterative optimization procedure on (a) SNU dataset and (b) SYM dataset.

### 5.5. Advantages of using multiple descriptors

We investigate into the performance gain from employing multiple descriptors comparing to a single descriptor by applying our approach. The visualization of the matching results with different descriptors on `mickeys` of SNU dataset and `bdom` of SYM dataset are shown in Figure 4 and Figure 5 respectively. Only correct correspondences are drawn in specific colors with respect to the adopted descriptors (SIFT in orange, LIOP in blue, DAISY in green, RI in magenta and GB in cyan). In Figure 4, there are three common objects. The overall result by DAISY is the best among the five descriptors, but it seems that if we use SIFT, we can have even better results for matching some parts of the image, *i.e.* the Mickey doll. The performance of SIFT on matching the cup of instant noodles is poor while DAISY in this case finds the most correspondences on it. Without making tradeoff of picking a descriptor, our method can adaptively select a good descriptor for matching each point, just as shown in Figure 4(f) where our approach unsupervisedly picks SIFT for matching the Mickey doll and DAISY for matching the cup noodles. We can have similar observation in Figure 5 where our approach selects SIFT for matching the dome and mostly GB for the other regions. The results in Figure 4 and Figure 5 show the advantage of applying our approach to multiple, complementary descriptors.

### 5.6. Comparisons of running time performance

We compare our approach with the baselines in terms of running time. All the experiments are conducted on a PC equipped with Intel $i7$-4770 CPU and 16GB memory. Our approach is implemented in `Matlab` and with the graph cut



(a) SIFT (219/516)

(b) LIOP (133/516)

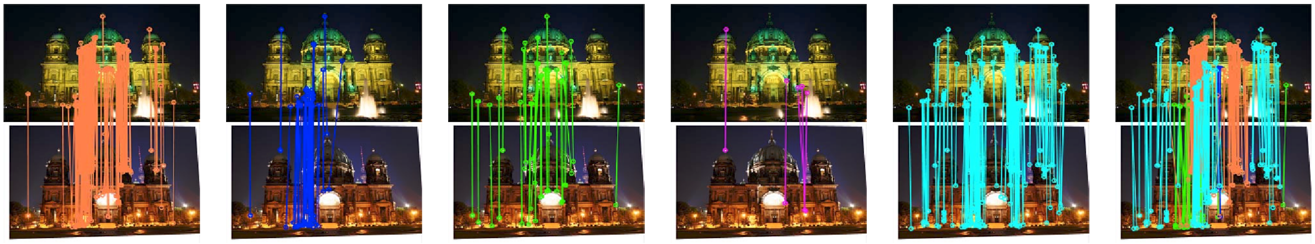
(c) DAISY (270/516)

(d) RI (130/516)

(e) GB (14/516)

(f) Ours (310/516)

Figure 4. Matching results of our approach by using the five descriptors individually (a)-(e) and jointly (f) on image pair, `mickeys` of SNU dataset. Correct matchings are drawn with the colors specifying the adopted or selected descriptors. The last row shows ( `nTP` / `nP` ).



(a) SIFT (288/1189)

(b) LIOP (82/1189)

(c) DAISY (66/1189)

(d) RI (10/1189)

(e) GB (278/1189)

(f) Ours (407/1189)

Figure 5. Matching results of our approach by using the five descriptors individually (a)-(e) and jointly (f) on image pair, `bdom` of SYM dataset. Correct matchings are drawn with the colors specifying the adopted or selected descriptors. The last row shows ( `nTP` / `nP` ).

Table 4. Running time on SNU dataset [12] and SYM dataset [18].

| Dataset | # of points | SM | ACC | HV | PGM | Ours |
|---|---|---|---|---|---|---|
| SNU | 1100±200 | 3.26 s | 452.44 s | 3.02 s | 82.21 s | 1.93 s |
| SYM | 1200±600 | 4.67 s | 1034.9 s | 4.36 s | 85.22 s | 3.95 s |

solver [5] in `C++`. Because convergence is reached within 2 iterations in most cases, we run our algorithm for 2 iterations on all the experiments. When our approach collaborates with $M = 5$ descriptors on SNU dataset [12], in which each image in average has 1,100 detected feature points, its running time per iteration is around $1.33$ seconds ($0.20$ seconds for optimizing $t$, $0.39$ seconds for optimizing $s$ and $0.73$ seconds for enrichment). If we directly solve $t$ and $s$ in Eq. (2), the running time per iteration is around $2.85$ seconds ($2.08$ seconds for optimizing $t$ and $s$, and $0.74$ seconds for enrichment). Solving Eq. (2) alternately is about $4$ times faster than solving it directly when optimizing $t$ and $s$, but both ways get similar performance. In addition, we compare our running time with other baselines and report the results in Table 4 on SNU dataset and SYM dataset. We implement SM and HV, and use the released implementations by [9] and [11] for ACC and PGM, respectively. As shown in Table 4, our method is much faster than ACC and PGM while comparable to SM and HV. The results demonstrate that our approach is superior to the four state-of-the-art approaches in both accuracy and efficiency.

More results and the code will be available at https://sites.google.com/site/yuantinghu/publications/featmat.

## 6. Conclusions

We have introduced an image matching approach with the capability of point-specific descriptor selection, cross descriptor geometric checking, and progressive correspondence enrichment. It is formulated as an optimization problem on a graph, and can be effectively solved by using the graph-cut algorithm. Through the iterative optimization process, the plausible match candidates are gradually revealed by taking their consistence with the nearby correspondences into account, while more and more correct correspondences are detected with the aid of enriched candidates. The proposed approach has been comprehensively evaluated on two benchmark datasets. Experimental results show that it outperforms the state-of-the-art methods in both the aspects of accuracy and efficiency. In the future, we would like to apply the proposed approach to vision applications, such as scene parsing and co-segmentation, where high-quality and dense matching is crucial. It would also be interesting to investigate how our approach to sparse matching collaborates with approaches to dense matching, such as image alignment and optical flow.

**Acknowledgement.** This work was supported in part by grants MOST 103-2221-E-001-026-MY2 and MOST104-2628-E-001-001-MY2.

## References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. Seitz, and R. Szeliski. Building Rome in a day. *Commun. ACM*, 2011. 1

[2] Y. Avrithis and G. Tolias. Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval. *IJCV*, 2014. 2

[3] A. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, 2001. 1, 2, 6

[4] A. Bosch, A. Zisserman, and X. Muñoz. Representing shape with a spatial pyramid kernel. In *Proc. ACM Conf. Image and Video Retrieval*, 2007. 3

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001. 1, 5, 8

[6] M. Brown and D. Lowe. Recognising panoramas. In *ICCV*, 2003. 1

[7] H.-Y. Chen, Y.-Y. Lin, and B.-Y. Chen. Robust feature matching with alternate Hough and inverted Hough transforms. In *CVPR*, 2013. 2, 6

[8] T.-J. Chin, J. Yu, and D. Suter. Accelerated hypothesis generation for multistructure data via preference analysis. *PAMI*, 2012. 2

[9] M. Cho, J. Lee, and K.-M. Lee. Feature correspondence and deformable object matching via agglomerative correspondence clustering. In *ICCV*, 2009. 2, 6, 8

[10] M. Cho, J. Lee, and K.-M. Lee. Reweighted random walks for graph matching. In *ECCV*, 2010. 2

[11] M. Cho and K.-M. Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. In *CVPR*, 2012. 2, 6, 8

[12] M. Cho, Y.-M. Shin, and K.-M. Lee. Co-recognition of image pairs by data-driven monte carlo image exploration. In *ECCV*, 2008. 2, 5, 6, 8

[13] O. Choi and I. Kweon. Robust feature point matching by preserving local geometric consistency. *CVIU*, 2009. 4

[14] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *NIPS*, 2006. 2

[15] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *ECCV*, 2004. 2

[16] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation from single or multiple model views. *IJCV*, 2006. 1

[17] M. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981. 2

[18] D. Hauagge and N. Snavely. Image matching using local symmetry features. In *CVPR*, 2012. 2, 5, 6, 8

[19] K.-J. Hsu, Y.-Y. Lin, and Y.-Y. Chuang. Robust image alignment with multiple feature descriptors and matching-guided neighborhoods. In *CVPR*, 2015. 3

[20] Y.-T. Hu, Y.-Y. Lin, H.-Y. Chen, K.-J. Hsu, and B.-Y. Chen. Matching images with multiple descriptors: An unsupervised approach for locally adaptive descriptor selection. *TIP*, 2015. 3

[21] J. Kim and K. Grauman. Boundary preserving dense local regions. 2011. 2

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

[23] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, 2008. 3

[24] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *PAMI*, 2010. 3

[25] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005. 6

[26] M. Leordeanu, R. Sukthankar, and M. Hebert. Unsupervised learning for graph matching. *IJCV*, 2012. 2

[27] H. Liu and S. Yan. Common visual pattern discovery via spatially coherent correspondences. In *CVPR*, 2010. 1, 2

[28] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2, 6

[29] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 2004. 3, 6

[30] E. Mortensen, H. Deng, and L. Shapiro. A SIFT descriptor with global context. In *CVPR*, 2005. 3, 6

[31] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *PAMI*, 1997. 1

[32] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Trans. on Graphics*, 2006. 1

[33] E. Tola, V. Lepetit, and P. Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *PAMI*, 2010. 6

[34] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, 2008. 2

[35] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007. 3

[36] C. Wang, L. Wang, and L. Liu. Progressive mode-seeking on graphs for sparse feature matching. In *ECCV*, 2014. 2

[37] Z. Wang, B. Fan, and F. Wu. Local intensity order pattern for feature description. In *ICCV*, 2011. 1, 2, 6

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

[39] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 2007. 3

[40] W. Zhang, X. Wang, D. Zhao, and X. Tang. Graph degree linkage: Agglomerative clustering on a directed graph. In *ECCV*, 2012. 2

[41] F. Zhou and F. Torre. Deformable graph matching. In *CVPR*, 2013. 2