Robust Feature Matching Using Spatial Clustering With Heavy Outliers

Xingyu Jiang, Jiayi Ma, Junjun Jiang, and Xiaojie Guo

***Abstract*— This paper focuses on removing mismatches from given putative feature matches created typically based on descriptor similarity. To achieve this goal, existing attempts usually involve estimating the image transformation under a geometrical constraint, where a pre-defined transformation model is demanded. This severely limits the applicability, as the transformation could vary with different data and is complex and hard to model in many real-world tasks. From a novel perspective, this paper casts the feature matching into a spatial clustering problem with outliers. The main idea is to adaptively cluster the putative matches into several motion consistent clusters together with an outlier/mismatch cluster. To implement the spatial clustering, we customize the classic density based spatial clustering method of applications with noise (*DBSCAN*) in the context of feature matching, which enables our approach to achieve quasi-linear time complexity. We also design an iterative clustering strategy to promote the matching performance in case of severely degraded data. Extensive experiments on several datasets involving different types of image transformations demonstrate the superiority of our approach over state-of-the-art alternatives. Our approach is also applied to near-duplicate image retrieval and co-segmentation and achieves promising performance.**

***Index Terms*— Feature matching, spatial clustering, DBSCAN, outlier, mismatch removal.**

I. Introduction

SEEKING reliable correspondences between two sets of image features is a fundamental problem in computer vision, and it has been a critical prerequisite in a wide spectrum of applications including 3D reconstruction, SLAM, image retrieval, image registration and fusion [1]–[7]. For example, in the system of structure-from-motion, the detailed quality of produced 3D points in structure from motion depends on the performance of corresponding matching [8].

The matching problem is typically solved in a two-step manner, *i.e.* first constructing a set of putative matches and then removing false matches from them. Very often, the putative set is formed by simply picking out point pairs with sufficiently similar feature descriptors (*e.g.*, scale invariant feature transform, SIFT [9]). However, the putative set includes, besides most of the true matches (*inliers*), a number of false matches (*outliers*), due to ambiguities of the local descriptors (particularly if the images suffer from low-quality, occlusion and repetitive patterns). Therefore, it is critical to design a robust approach to remove outliers for boosting the reliability of matches.

Existing methods usually address the outlier removal by imposing a geometric constraint, which restricts matches satisfying an underlying image transformation. In general, the transformation can vary with respect to different data. Thus, a pre-defined transformation model is often demanded, which can be either parametric (*e.g.*, affine, homography, epipolar geometry [10]) or non-parametric (*e.g.*, non-rigid [11]). However, this demand severely limits the applicability in many vision-based tasks such as deformable object recognition and dynamic scene matching, as the transformation models in these tasks are unknown beforehand. Moreover, the high computational complexity is another demerit of existing methods, especially when the image transformation is a complex non-rigid model, which is a further obstacle in real-time tasks.

To address the above issues, in this paper we propose a spatial clustering method aiming to exploit the motion consistency among the putative matches. This is based on the observation that the correct matches tend to have similar motion behavior, which could be clustered into several motion consistent groups, while the false matches tend to be randomly distributed across the image domain which can be labeled as outliers. To illustrate this idea, in Fig. 1 we present some typical image pairs and show their putative matches established by SIFT. From the results, we see that despite the different types of transformations in different scenes, the correct matches marked in different colors always tend to have coherent motions, with neighboring points sharing the similar motion.

Ideally, if the intrinsic number of clusters is given and feature correspondences are accurate, obtaining a reasonable result of clustering may be not difficult. However, on the one hand, we typically do not have the exact cluster amount at hand in practice; on the other hand, the correspondences often contain (many) false-positive pairs (*a.k.a.* outliers). These two issues significantly increase the difficulty of screening out the

Manuscript received February 3, 2019; revised July 2, 2019; accepted August 7, 2019. Date of publication August 26, 2019; date of current version October 9, 2019. This work was supported by the National Natural Science Foundation of China under Grant 61773295, Grant 61971165, and Grant 61772512. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jing-Ming Guo. *(Corresponding author: Jiayi Ma.)*

X. Jiang and J. Ma are with the Electronic Information School, Wuhan University, Wuhan 430072, China (e-mail: jiangx.y@whu.edu.cn; jyma2010@gmail.com).

J. Jiang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: junjun0595@163.com).

X. Guo is with the School of Computer Software, Tianjin University, Tianjin 300350, China (e-mail: xj.max.guo@gmail.com).

Digital Object Identifier 10.1109/TIP.2019.2934572

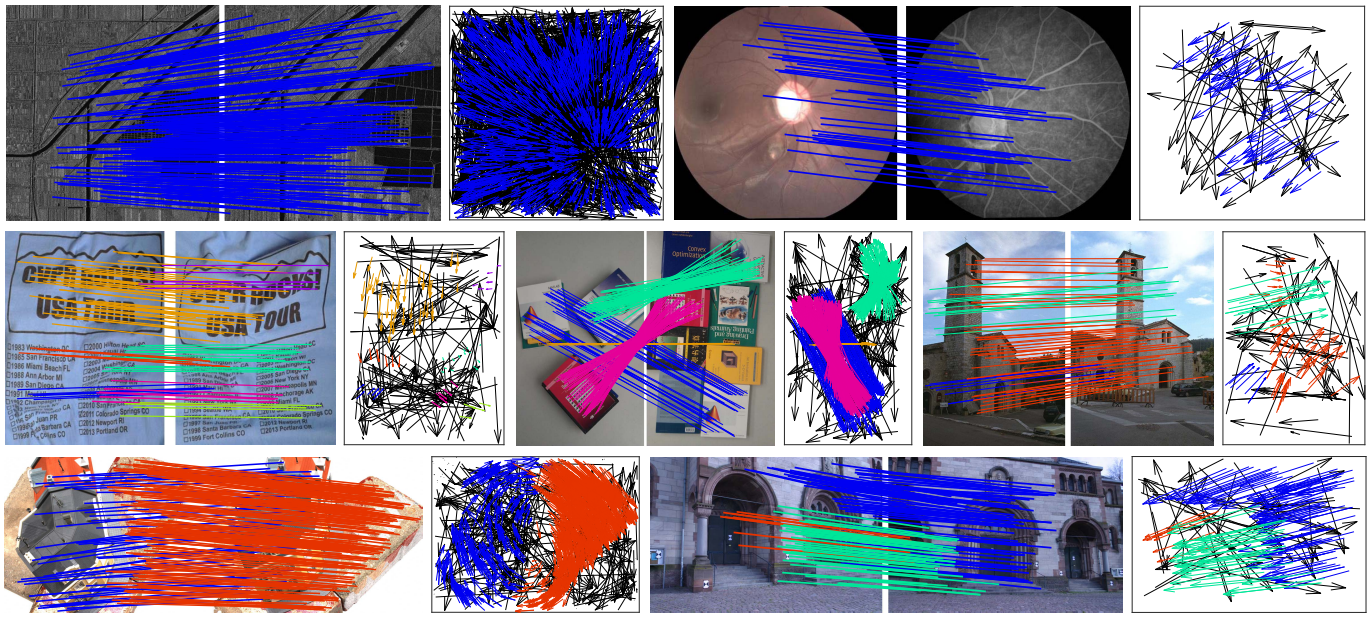


Fig. 1. Results of our proposed method on 7 typical image pairs. In each group of result, we show at most 200 feature matches in the left image pair for clarity; the head and tail of each arrow in the right motion field correspond to the positions of two corresponding feature points in the left image pair. Different colors denote different inlier clusters, and black indicates the outlier cluster.

outliers and clustering the inliers. *Thus, it is natural to ask that: can we automatically determine the number of clusters and eliminate the outliers simultaneously?* This paper tries to positively answer the above question. In particular, to capture the motion consistency, we cast the matching problem into spatial clustering with outliers, where the classic density-based spatial clustering of applications with noise (DBSCAN) [12] is customized to solve the problem. We also design an iterative clustering strategy to promote the matching performance when the putative matches suffer from a large number of outliers. In addition, our proposed method has quasi-linear complexity, and hence is beneficial to addressing large-scale or real-time matching problems.

Our contributions in this paper include the following three aspects. (i) We propose a simple yet efficient method for robust feature matching using spatial clustering. It does not require a pre-defined transformation model as existing attempts do and can exploit multiple motion patterns in an image scene. (ii) We customize the classic DBSCAN to solve the matching problem, and design a general iterative clustering strategy which can promote the performance of DBSCAN-based methods in case of severely degraded data. (iii) We apply our feature matching method to two vision-based tasks, saying image retrieval and co-segmentation, and achieve satisfying performance.

II. Related Work

In this section, we first briefly introduce related works on feature matching and spatial clustering, and then introduce the DBSCAN algorithm that our work is based on in detail.

A. Feature Matching

Feature matching has been widely used in many fields including computer vision [3], medical imaging [13], remote sensing [14], robotics [15], [16], to name just a few. To remove outliers from putative sets, various techniques have been developed, which can be roughly categorized into three groups, *i.e.* resampling methods, non-parametric interpolation methods, and graph matching methods.

The resampling methods follow a hypothesis-and-verification strategy, the principle of which is to find the smallest possible outlier-free subset to estimate a pre-defined transformation model by resampling. The random sample consensus (RANSAC) algorithm [10] and analogous variants [17] are two classic schemes in resampling based methods. These approaches perform reasonably well when the geometric constraints are parametric. However, they have exposed their limitations when the geometric constraints are non-parametric. Furthermore, their performance sharply degenerates or even fails when the outliers in the putative set are dominant.

To mitigate the abovementioned issues, several non-parametric interpolation methods have been investigated, including identifying correspondence function (ICF) [18], and manifold regularization-based robust point matching (MR-RPM) [11]. The ICF seeks a correspondence function pair, mapping points in one image to their corresponding points in the other one. Then the outliers can be kicked out by checking the deviation in the estimated correspondence function aggressively. While the MR-RPM enforces the motion field to be smooth under manifold regularization and conquers the matching problem from a robust motion field interpolation perspective. However, the methods in this category are typically of cubic complexities, limiting their applicability to real-time tasks.

Graph matching is another alternative for solving the matching problem, with spectral matching [19], dual decomposition [20], mode-seeking [21], graph shift (GS) [22],

and multi-graph matching [23]–[25] as representatives. These methods usually formulate the feature matching as quadratic assignment problem to seek the maximum inlier set based on the affinity matrix. Graph matching provides considerable flexibility to the transformation model, but suffers from similar drawbacks of its non-polynomial-hard nature, which is not applicable to the large-scale vision tasks.

In the recent past, the feature matching has been addressed using piecewise-smoothness constraints, such as coherence based decision boundaries [2], grid-based motion statistics (GMS) [3], learning for mismatch removal [26], and learning a deep network to find good correspondences (LFGC) [5], which have achieved promising performance in terms of both accuracy and efficiency. In addition, several techniques have also been investigated to address specific matching problems, such as large scale changed image matching [27], 3D point cloud registration [28]–[30], as well as semantic region correspondences [31].

B. Spatial Clustering

Spatial clustering is the task of grouping a set of samples in a way that samples in the same cluster are more similar to each other than to those in other clusters [32]. The spatial clustering methods typically include connectivity-based method such as hierarchical clustering [33], centroid-based method such as K-means [34], and distribution-based method such as Gaussian mixture models clustering [35]. These methods, however, are not robust to noisy samples/outliers, *e.g.,* they do not set a special outlier cluster and the outliers are usually classified into the normal clusters which are most similar to them. In addition, these methods sometimes demand that the database should satisfy a specific distribution, and the computational complexity is also relatively high, limiting their capability to address the feature matching problem. For example, for the centroid-based methods, the samples are always assigned to the nearest center, leading to their failures on nonspherical or manifold clusters. While the accuracy of distribution-based methods usually depends on the capability of the trail probability to represent the data, and they are typically extremely slow especially when there are a large number of clusters [36].

One of the most representative spatial clustering algorithms with outliers is the DBSCAN [12]. It is robust to outliers and has relatively low complexity in contrast to many other clustering methods. Therefore, it is a good candidate to address mismatch removal in the feature matching problem.[1] *Nevertheless, there is a key drawback of DBSCAN, say the sensitivity to parameter settings, which will be problematic in addressing complex feature matching problems.* In addition, the clustering performance will be degraded if the outliers are dominated in the database, which often occurs in the feature matching problem. In this paper, we design an adaptive parameter estimation method and an iterative clustering strategy to address these challenges with DBSCAN as the basic model.

[1]Note that other density-based clustering methods such as density peak [36] and DBSCAN invariants [37] are also workable for robust feature matching. Here we choose the original DBSCAN due to its simplicity and generality and for the purpose of handling more general feature matching tasks.

C. DBSCAN

The basic idea of DBSCAN is that for each sample point of a cluster the neighborhood of a given radius (ε) should contain at least a minimum number of sample points ($MinPts$), where ε and $MinPts$ are two parameters. The DBSCAN is mainly realized by the following definitions.

Definition 1 (ε-Neighborhood of a Sample): The *ε-neighborhood* of a sample $\mathbf{p}$, denoted by $\mathcal{N}_\varepsilon(\mathbf{p})$, is defined as $\mathcal{N}_\varepsilon(\mathbf{p}) = \{\mathbf{q} \in D | d(\mathbf{p}, \mathbf{q}) \leq \varepsilon\}$, where D is a database of sample points, and d is a certain distance metric such as Euclidean distance.

Based on the above definition, if we require each inlier $\mathbf{p}$ to satisfy $|\mathcal{N}_\varepsilon(\mathbf{p})| \geq MinPts$, then the inliers located in the border of a cluster (*border inliers*) may be easily misjudged as outliers as they typically have fewer neighborhoods. To address this issue, DBSCAN defines the inliers satisfying $|\mathcal{N}_\varepsilon(\mathbf{p})| \geq MinPts$ as *core samples*, and recalls the border inliers by using the following additional definitions.

Definition 2 (Directly Density-Reachable): A sample $\mathbf{p}$ is *directly density-reachable* from a sample $\mathbf{q}$ *w.r.t.* ε and $MinPts$, if $\mathbf{p} \in \mathcal{N}_\varepsilon(\mathbf{q})$ and $\mathbf{q}$ is a core sample.

Definition 3 (Density-Reachable): A sample $\mathbf{p}$ is *density-reachable* from a sample $\mathbf{q}$ *w.r.t.* ε and $MinPts$, if there exists a chain of samples $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n$, where $\mathbf{p}_1 = \mathbf{q}$, $\mathbf{p}_n = \mathbf{p}$, such that $\mathbf{p}_{i+1}$ is directly density-reachable from $\mathbf{p}_i$.

Definition 4 (Density-Connected): A sample $\mathbf{p}$ is *density-connected* to a sample $\mathbf{q}$ *w.r.t.* ε and $MinPts$, if there is a sample $\mathbf{o}$ such that both $\mathbf{p}$ and $\mathbf{q}$ are density-reachable from $\mathbf{o}$ *w.r.t.* ε and $MinPts$.

The issue about border sample misjudgement could be well addressed through the definition of density-connected, and then a cluster can be intuitively defined as a set of density-connected samples *w.r.t.* density-reachability under parameters ε and $MinPts$.

Definition 5 (Cluster): A cluster C *w.r.t.* ε and $MinPts$ is a non-empty subset of D which satisfies the following conditions: (i) Maximality. $\forall \mathbf{p}$, $\mathbf{q}$: if $\mathbf{p} \in C$ and $\mathbf{q}$ is density-reachable from $\mathbf{p}$ *w.r.t.* ε and $MinPts$, then $\mathbf{q} \in C$. (ii) Connectivity. $\forall \mathbf{p}, \mathbf{q} \in C$: $\mathbf{p}$ is density-connected to $\mathbf{q}$ *w.r.t.* ε and $MinPts$.

Definition 6 (Outliers): Let $C_1, \cdots, C_k$ be the clusters. We define the $outliers$ as the set of samples in the database D not belonging to any cluster C_i, *i.e.* $outliers = \{\mathbf{p} \in D \mid \forall i : \mathbf{p} \notin C_i\}$.

The DBSCAN can discover a cluster by a two-step approach. First, given parameters ε and $MinPts$, it chooses an arbitrary sample from the database D satisfying the core sample condition. Then it retrieves all samples density-reachable from the core sample and determines all samples density-connected to each other as one cluster. Algorithm 1 simply summarizes the procedure, while for more details, please refer to the original paper [12].

III. Method

The first step of feature matching is to construct a set of putative matches by considering all possible matches between the given two feature point sets with those having distant

Algorithm 1 The *DBSCAN* Algorithm

Input: Observed database D, parameters ε, $MinPts$
Output: *ClusterID* of each sample in D
1 Initialize *ClusterID* with $\mathbf{0}$;
2 Calculate the distance matrix $\mathbb{D}$ of all samples from D;
3 Calculate $\mathcal{N}_\varepsilon$ of each sample based on Def. 1 with $\mathbb{D}$;
4 Select core samples by comparing $\mathcal{N}_\varepsilon$ and $MinPts$;
5 Identify density-connected samples using Defs. 2–4;
6 Label *ClusterID* of each sample based on Defs. 5, 6.

descriptors eliminated, for example, filtering out the matches between feature points whose SIFT descriptors are too dissimilar under the Euclidean distance. Then the problem boils down to removing false matches from the putative set. Fortunately, many well-designed image descriptors (e.g., SIFT [9]) can efficiently establish putative matches. Thus, in the following, we focus on outlier removal and, from a novel perspective, formulate it as a spatial clustering problem.

A. Problem Formulation

As aforementioned, two nearby feature points should have similar motion properties, such as direction and length of motion vector. Thus the motion field vectors induced by the putative matches usually consist of one or more motion consistent clusters together with an outlier one. In particular, the motion consistent constraint could involve rotation and scale change, and matches from the same class typically come from the same object or same depth of field in the image scene. Therefore, the outlier removal problem can be casted as a spatial clustering problem with outliers, which is essentially to address the following two problems:

- Characterize each putative match as a sample for clustering, *e.g.,* construct the properties of each putative match for similarity measurement, and design a distance metric specific for the matching problem.
- Design clustering rules in the context of feature matching, which should be adaptive and robust to parameter changes, to divide the putative matches into several motion consistent clusters and an outlier cluster.

Suppose we have obtained a set of N putative feature matches $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ from two given images, where $\mathbf{x}_i$ and $\mathbf{y}_i$ are two-dimensional vectors representing the spatial positions (*i.e.*, image coordinates) of the two corresponding feature points, respectively. Let $\mathbf{m}_i = \mathbf{y}_i - \mathbf{x}_i$ denote the motion vector of match $(\mathbf{x}_i, \mathbf{y}_i)$. We then convert the putative match set $\mathcal{S}$ into a noisy observation database D for spatial clustering using the following rule:

$$D = \{\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}, i = 1, 2, \cdots, N\}, \tag{1}$$

where $\mathbf{p}_i$ is a sample denoting the property of putative match $(\mathbf{x}_i, \mathbf{y}_i)$. To enhance the motion consistence, we design a weighted distance $d(\mathbf{p}_i, \mathbf{p}_j)$ as follows:

$$d(\mathbf{p}_i, \mathbf{p}_j) = \phi(\mathbf{x}_i, \mathbf{x}_j) + \phi(\mathbf{y}_i, \mathbf{y}_j) + w_{i,j} \cdot \phi(\mathbf{m}_i, \mathbf{m}_j), \tag{2}$$

with the weight parameter $w_{i,j}$ defined as

$$w_{i,j} = 1 + \gamma \cdot e^{-\min\{\phi(\mathbf{x}_i, \mathbf{x}_j), \phi(\mathbf{y}_i, \mathbf{y}_j)\}}, \tag{3}$$

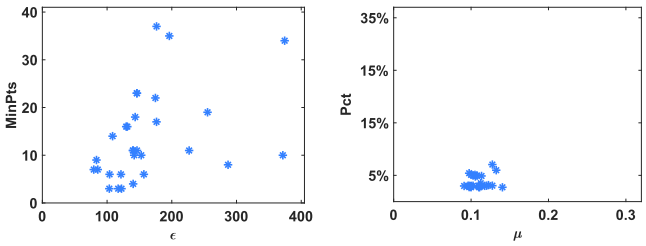


Fig. 2. Illustration of parameters distribution on 30 randomly chosen image pairs involving different types of transformations. Left: optimal parameter values of $(\varepsilon, MinPts)$; right: optimal parameter values of (μ, Pct). Each scatter point denotes the optimal parameter values on a certain image pair.

where $\phi(\cdot)$ denotes the distance measurement function, such as Euclidean distance and Gaussian kernel distance. In this paper, we adopt the Euclidean distance which works sufficiently well. Parameter γ is a positive number to enhance the motion consistence among neighboring feature points. Therefore, we can calculate the $N \times N$ distance matrix $\mathbb{D}$ accordingly with $\mathbb{D}_{i,j} = d(\mathbf{p}_i, \mathbf{p}_j)$, and the DBSCAN in Alg. 1 can be used to identify the outliers in the putative match set $\mathcal{S}$.

Note that compared with traditional matching methods, the proposed clustering-based strategy is more general. *In particular, unlike traditional methods that typically rely on pre-defined transformation models, our method can address image pairs undergoing any transformation models.*

B. Adaptive Parameter Estimation

After constructing the database D and defining the weighted distance d, the major problem of applying DBSCAN to mismatch removal is how to choose its parameters adaptively when handling complex matching problems. The left plot of Fig. 2 demonstrates the distributions of optimal parameter values for ε and $MinPts$ on 30 randomly chosen image pairs having different types of transformations, *e.g.* piecewise linear transformation, non-rigid deformation, wide baseline image pair, *etc*. The SIFT is adopted to extract putative matches, which are further converted into database samples according to Eq. (1). The matching performance is characterized by precision, recall and F-score, where the precision (P) is defined as the ratio of the identified inlier number and the preserved match number, the recall (R) is defined as the ratio of the identified inlier number and the whole inlier number, and the F-score is defined as the ratio of $2PR$ and $P + R$.

From the results, we see that the variances of the optimal parameter values are quite large, and hence using predefined fixed parameter values will be problematic to achieve accurate matching performance. Clearly, it is impossible to determine the parameters manually for each image pair when addressing real-world or large scale matching tasks such as image retrieval, SLAM and 3D reconstruction. Although the DBSCAN has developed a simple heuristic to determine the parameters based on searching the inflection point of the "thinnest" cluster in the database, it typically fails if there are more than one inflection points or outliers are dominated in the database D which frequently happens in the feature matching problem. Therefore, it is significant to develop a method for adaptive parameter estimation.

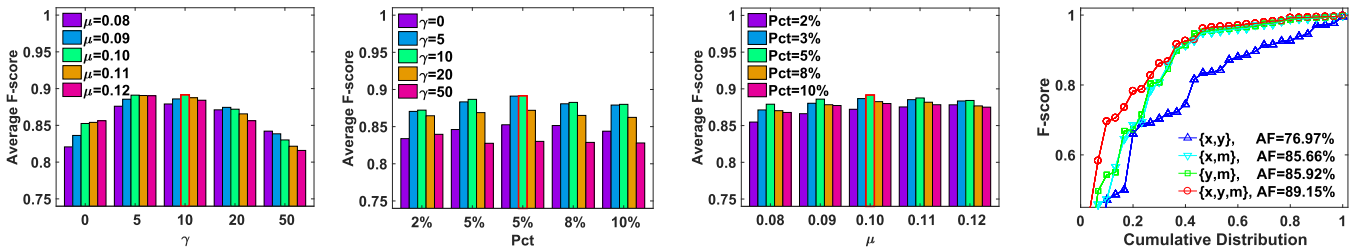


Fig. 3. F-scores with different parameter settings and different definitions of $\mathbf{p}$ on the 30 image pairs in Fig. 2. For the first three plots, we respectively fix Pct, μ and γ, and change the other two to find the optimal settings. The last plot provides the F-score curves for different definitions of $\mathbf{p}$, *i.e.*, $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}})^{\mathrm{T}}$, $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$, $\mathbf{p}_i = (\mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$, and $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$. A point on the curve in the last plot with coordinate (x, y) denotes that there are $100 \times x$ percent of image pairs which have F-score no more than y.

To address the above mentioned issue, we first introduce the concept of *K-dist* and give its definition as follows:

Definition 7 (K-Dist of a Sample): For any positive integer K, the *K-dist* of a sample $\mathbf{p}$, denoted as *K-dist*($\mathbf{p}$), is defined as $d(\mathbf{p}, \mathbf{q})$ between $\mathbf{p}$ and sample $\mathbf{q} \in D$ such that: (i) for at least K samples $\mathbf{o} \in D \setminus \mathbf{p}$, it holds that $d(\mathbf{p}, \mathbf{o}) \leq d(\mathbf{p}, \mathbf{q})$; (ii) for at most $K-1$ samples $\mathbf{o} \in D \setminus \mathbf{p}$, it holds that $d(\mathbf{p}, \mathbf{o}) < d(\mathbf{p}, \mathbf{q})$.

The optimal values of radius ε and minimum sample number $MinPts$ depend on the density of database D, which changes with different image scenes. From a new perspective, we determine the core samples by replacing the constraint $|\mathcal{N}_\varepsilon(\mathbf{p})| \geq MinPts$ with *K-dist*($\mathbf{p}$) $\leq \varepsilon$, where the positive integer K plays the same role of parameter $MinPts$. Without loss of generality, we assume that the optimal K is determined by N with a percentage Pct, *i.e.,* $K = [N \cdot Pct]$, where $[\cdot]$ represents the rounding operation. In addition, we bound its values between B_L and B_U to ensure robustness. Therefore, we determine K as follows:

$$K = \max\{\min\{[N \cdot Pct], B_U\}, B_L\}. \tag{4}$$

The parameter K is used to constrain the scale of cluster. For example, a cluster should contain at least K samples within the radius ε, *i.e.*, one core sample. In this paper, we empirically set $B_L = 3$ and $B_U = 30$, which works sufficiently well for addressing the feature matching problem.

Next, we focus on the adaptive estimation of the other parameter ε. For each database D, we estimate K using Eq. (4) and calculate the *K-dist* of each sample to produce a set $\mathbf{d}_K$:

$$\mathbf{d}_K = \{K\text{-}dist(\mathbf{p}) | \mathbf{p} \in D\}. \tag{5}$$

Clearly, ε should belong to $[\min(\mathbf{d}_K), \max(\mathbf{d}_K)]$. Without loss of generality, we make an assumption that the optimal ε is determined by $\mathbf{d}_K$ with a parameter $\mu \in [0, 1]$ as follows:

$$\varepsilon = \mu \cdot \big(\max\{\mathbf{d}_K\} - \min\{\mathbf{d}_K\}\big) + \min\{\mathbf{d}_K\}. \tag{6}$$

Therefore, the estimation of parameters $MinPts$ and ε is converted to determining parameters Pct and μ. In fact, for the feature matching problem, the parameters Pct and μ have global optima that are robust to different image scenes, as shown in the right plot of Fig. 2. To further investigate the best parameter settings of Pct, μ as well as the motion consistency weight γ, we test the average F-scores with different parameter settings on the 30 image pairs. The results are reported in Fig. 3, we respectively fix one parameter of $\{Pct, \mu, \gamma\}$ as its "optimal" setting and change the other two to find the optimal settings. As can be seen from the results, $Pct = 5\%$, $\mu = 0.1$ and $\gamma = 10$ achieve the best average F-score, which are considered as the default optimal parameter settings throughout this paper. Clearly, a positive value of γ but no more than 20 can significantly enhance the matching performance compared with $\gamma = 0$.

Note that the definition $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$ in Eq. (1) bears some redundancy as the motion vector $\mathbf{m}_i$ is defined as $\mathbf{y}_i - \mathbf{x}_i$. This indicates a manifold structure in the data space used for clustering. However, such definition is beneficial for promoting the clustering performance in the presence of outliers. To validate this idea, we provide a quantitative comparison for the mismatch removal performance on the 30 image pairs in Fig. 2. To this end, we construct the match sample $\mathbf{p}_i$ with $(\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}})^{\mathrm{T}}$, $(\mathbf{x}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$, $(\mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$, and $(\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$, respectively. Their corresponding F-score curves are given in the last plot of Fig. 3, where the average F-scores are 76.97%, 85.66%, 85.92% and 89.15%, respectively. Clearly, using $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}})^{\mathrm{T}}$ has the worst performance, using $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$ and $\mathbf{p}_i = (\mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$ have the similar performance, while using $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$ has the best performance. We give an explanation as follows. Firstly, the spatial positions of feature points $\mathbf{x}_i$ and $\mathbf{y}_i$ typically have high correlation. Using the motion vector can remove such correlation, and hence can increase the separability of the data. Therefore, using $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$ or $\mathbf{p}_i = (\mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$ achieves better performance than using $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}})^{\mathrm{T}}$. Secondly, it is straightforward that using $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$ and $\mathbf{p}_i = (\mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$ achieve the similar performance, as they present similar structures in the data space. Thirdly, by using the definition of $\mathbf{p}_i = (\mathbf{x}_i^{\mathrm{T}}, \mathbf{y}_i^{\mathrm{T}}, \mathbf{m}_i^{\mathrm{T}})^{\mathrm{T}}$, the input data are put into a higher dimensional space, which in general can further increase the separability of the data. Therefore, it can achieve the best performance. This is somewhat similar to the property that a nonlinear separable space could become linear separable by increasing the dimension of the input data (typically have a manifold structure).

C. Iterative Clustering

In the DBSCAN algorithm, a critical step is to use the ε-neighborhood $\mathcal{N}_\varepsilon$ (or *K-dist*) to identify core samples and retrieve border inliers. As can be seen from Def. 1, $\mathcal{N}_\varepsilon$ is defined on the whole database D which also involves noisy samples/outliers. *Typically, the noisy samples may lead to misjudgement and this problem will be magnified when D is*

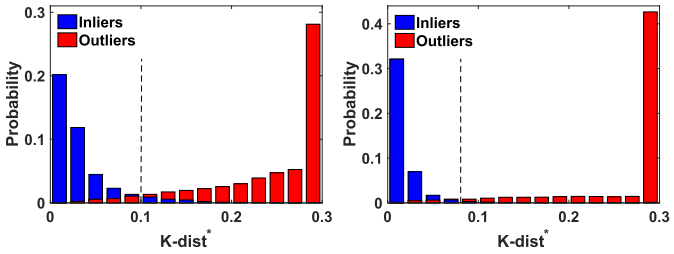


Fig. 4. Distribution of *K-dist** of putative matches from the 30 image pairs in Fig. 2. Left: using putative set D to construct neighborhood; right: using putative inlier set to construct neighborhood. For each bin, we overlap the inlier and outlier probabilities, where the one with smaller probability is shown in the outer layer.

contaminated by a large number of outliers which often occurs in feature matching. Therefore, it is desirable to define $\mathcal{N}_\varepsilon$ on the inlier set $\mathcal{I}$ as follows:

$$\mathcal{N}_\varepsilon(\mathbf{p}) = \{\mathbf{q} \in \mathcal{I} | d(\mathbf{p}, \mathbf{q}) \leq \varepsilon\}. \tag{7}$$

However, the inlier set $\mathcal{I}$ is to be solved in our problem and unknown in advance. To solve this dilemma, in this paper we propose a simple yet effective iterative clustering strategy. In particular, we first construct $\mathcal{N}_\varepsilon$ based on the whole database D and obtain the clustering result. Then we extract the inliers according to the clustering result and use it as an approximation to $\mathcal{I}$ and construct $\mathcal{N}_\varepsilon$ in Eq. (7) for the next round of clustering. This procedure can proceed until convergence. In our experiments we found that two iterations are sufficient to produce satisfying results in the context of feature matching, and hence we adopt two iterations as the default setting for efficiency. By using this iterative clustering strategy, the calculation of $\mathcal{N}_\varepsilon$ for an inlier sample will be less influenced by the outliers, which is beneficial to mismatch removal, especially when the input data are severely degraded.

To validate this idea, we use the aforementioned 30 image pairs and for each image pair we calculate $\mathbf{d}_K$ using Eq. (5), which is subsequently normalized to [0, 1] as:

$$\mathbf{d}_K^* = \frac{\mathbf{d}_K - \min\{\mathbf{d}_K\}}{\max\{\mathbf{d}_K\} - \min\{\mathbf{d}_K\}}. \tag{8}$$

Then we obtain a normalized *K-dist* for each putative match, denoted as *K-dist**. The statistical results of *K-dist** on all the 30 image pairs are reported in the left plot of Fig. 4. We see that the inliers tend to have small values of *K-dist** while the outliers tend to have large values of *K-dist**, and they can be roughly separated with a proper threshold.[2] With a threshold 0.1, we can obtain an average F-score about 0.892, and hence the putative inliers are able to achieve a good approximation to the ground truth inlier set $\mathcal{I}$. By using our iterative clustering strategy, in the second iteration, we adopt the neighborhood definition in Eq. (7), which is equivalent to calculating the *K-dist* in Def. 7 by requiring $\mathbf{q} \in \mathcal{I}$ rather than $\mathbf{q} \in D$. In this case, the margin between inliers and outliers is distinctly enlarged, as shown in the right plot of Fig. 4. With a threshold 0.08, the average F-score can be significantly promoted from 0.892 to 0.931.

[2]In fact, the optimal threshold corresponds to the value of μ.

Algorithm 2 The *RFM-SCAN* Algorithm

Input: Putative set $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, parameter γ, Pct, μ
Output: Inlier set $\mathcal{I}$
1 Construct database D based on $\mathcal{S}$ using Eq. (1) ;
2 Calculate *K-dist* of each sample in D using Def. 7 ;
3 Determine ε and $MinPts$ by Pct and μ based on *K-dist* ;
4 Run *DBSCAN* in Alg. 1 and obtain a putative inlier set $\mathcal{I}_0$;
5 Update ε and $MinPts$ using $\mathcal{I}_0$ to construct neighborhood;
6 Run *DBSCAN* in Alg. 1 and obtain the inlier set $\mathcal{I}$.

As the proposed robust feature matching is based on spatial clustering algorithm with noisy samples, we name it as *RFM-SCAN* and summarize the whole procedure in Alg. 2.

D. Computational Complexity

Our *RFM-SCAN* involves three major steps including the *K-dist* calculation, adaptive parameter estimation and outlier removal with DBSCAN. For the *K-dist* calculation, it requires to search the K-th nearest neighbor for each sample in D, and the time complexity is close to $O(N \log N)$ by using K-D tree [38]. For the adaptive parameter estimation, searching the minimum and maximum in $\mathbf{d}_K$ costs $\log N$ complexity. For DBSCAN, it first requires to construct the ε-neighborhood $\mathcal{N}_\varepsilon$ for each sample and determine the core points, which has time complexity $O(N \log N)$. Then it retrieves the border inliers and labels the clusterID of each sample from $\mathcal{N}_\varepsilon$, which has time complexity $O(\sum_{i=1}^N |\mathcal{N}_\varepsilon(\mathbf{p}_i)|)$, and can be approximately written as $O(KN)$. Therefore, the total time complexity is about $O(N(K + \log N))$. The space complexity of our *RFM-SCAN* is about $O(KN)$ due to the memory requirements for storing the neighborhood $\mathcal{N}_\varepsilon$. Generally, K is a constant and $K \ll N$, thus the time and space complexities of our method can be simply written as $O(N \log N)$ and $O(N)$, respectively. This is significant for addressing large-scale problems or real-time tasks.

IV. Experiment Results

In this section, we test the performance of our RFM-SCAN on general feature matching and apply it to two vision-based tasks, *i.e.* image retrieval and co-segmentation. We implement our algorithm with MATLAB code. The experiments are conducted on a desktop with 4.0 GHz Intel Core i7-6700K CPU and 8GB memory.

A. Results on Feature Matching

1) Qualitative Illustration: Figure 1 presents some intuitive results on the matching performance of our RFM-SCAN. The seven image pairs undergo different types of image transformations including affine (1st), non-rigid (2nd, 3rd and 4th), and epipolar geometry (5th, 6th and 7th). The initial inlier ratios in the seven testing pairs are about 43.09%, 49.50%, 43.81%, 75.74%, 56.35%, 78.49% and 68.48%, respectively. The ground truth is established by manually checking each putative match in each image pair, and we make the benchmark before conducting experiments to ensure its objectivity. By using our RFM-SCAN to remove false matches, it can obtain

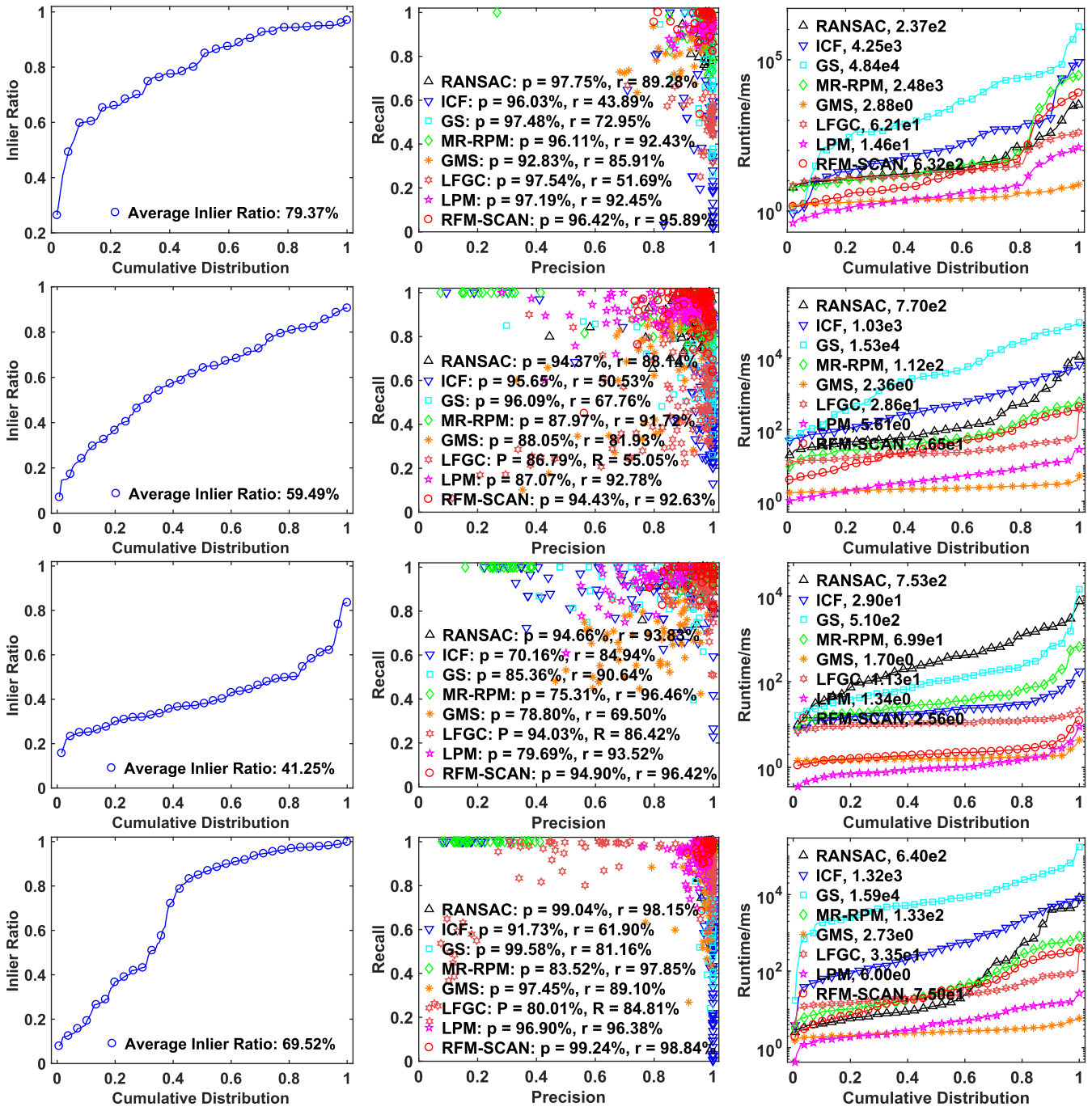


Fig. 5. Quantitative comparison on four datasets. From top to bottom: *Daisy*, *DTU*, *Retina* and *RS*. From left to right: cumulative distribution of inlier ratio in the putative sets, precision-recall statistics, and cumulative distribution of runtime. A point on the curve in the first and third columns with coordinate (x, y) denotes that there are $100 \times x$ percent of image pairs which have inlier ratio or runtime no more than y.

precision-recall pairs $(97.16\%, 100.0\%)$, $(100.0\%, 100.0\%)$, $(98.94\%, 93.94\%)$, $(90.61\%, 99.12\%)$, $(95.95\%, 100.0\%)$, $(97.28\%, 99.81\%)$, and $(99.21\%, 100.0\%)$, respectively. From the results, we see that the number of clusters in each image pair is automatically determined based on the motion consistency of correct matches (this can be clearly seen in the 4th *Book* pair), and very few putative matches are misjudged on all testing pairs. Therefore, our RFM-SCAN has strong generalization ability to handle different types of transformations.

Note that it may be not straightforward to understand why our RFM-SCAN is robust to image rotation and scale change. We give an explanation as follows. Our method is based on the observation that the correct matches tend to have similar motion behavior (consistency). In fact, such motion consistency is defined in a local small region, *i.e.*, ϵ-neighborhood with $MinPts$ elements. That is to say, if the motion vectors in such a local small region are similar, then they are considered to be consistent. By transmitting such motion consistency throughout the whole putative match set, we finally cluster the putative set into several motion consistent groups. Clearly, for rotation and scaling, the differences between the motion vectors associated with correct matches in a small region

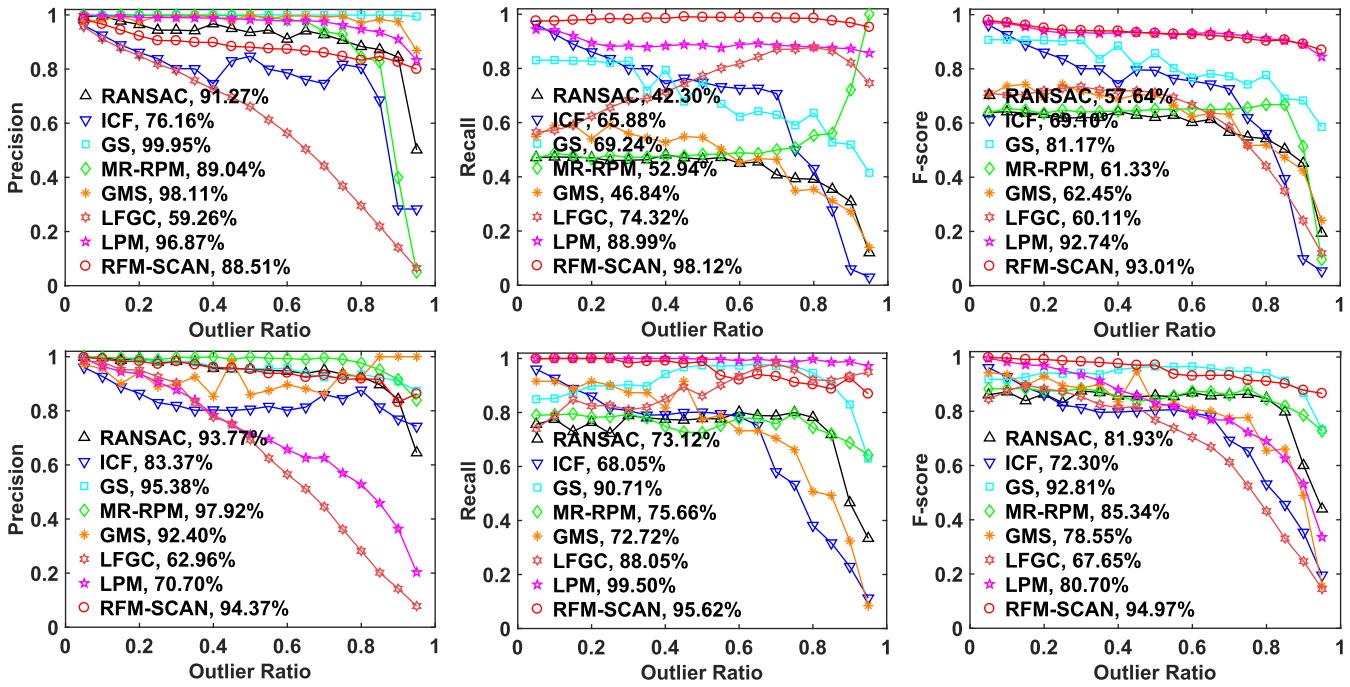


Fig. 6. Robustness test of different methods on two typical pairs with relatively complex transformations in Fig. 1, such as *Book* (top) and *Church* (bottom). From left to right: the average precision, recall and F-score at different outlier ratios over 20 trials.

are small. At least, the motion consistency of inliers is clearly stronger than that of outliers. Therefore, considering the transitivity of the motion consistency, we can identify the correct matches, even in case of large rotation or scaling, as illustrated in the 1st and 6th examples in Fig. 1.

2) Quantitative Comparison: Next, we provide quantitative evaluation of our RFM-SCAN with comparison to seven classic and state-of-the-art feature matching methods including RANSAC [10], ICF [18], GS [22], MR-RPM [11], GMS [3], LPM [1], and LFGC [5]. All the seven methods are implemented based on the publicly available codes, and we have tried our best to tune their parameters. The experiments are conducted on four datasets including *DAISY*, *DTU*, *Retina* and *RS*. In particular, *DAISY* [39] consists of several wide baseline pairs and two short sequences with ground truth depth maps; from which we create 52 image pairs in total for evaluation. *DTU* [40] contains many different scenes with ground truth camera positions; from which we choose two scenes (*i.e.*, *Frustum* and *House*) and create 131 image pairs with large viewpoint changes for evaluation. *Retina* [1] is a medical dataset consisting of 65 retinal image pairs undergoing non-rigid transformations. *RS* [1] is a remote sensing dataset consisting of 156 image pairs including color-infrared, *SAR* and panchromatic photographs. For the first two datasets, the ground truth feature matches are established based on the ground truth information supplied by the datasets. For the other two datasets, the ground truth matches are established with a benchmark as aforementioned.

The initial inlier ratio, precision, recall and runtime statistics on the four datasets are summarized in Fig. 5. We see that the initial inlier ratios, especially in the *Retina* dataset, are quite low, making the feature matching task challenging. The average numbers of putative matches in the four datasets are

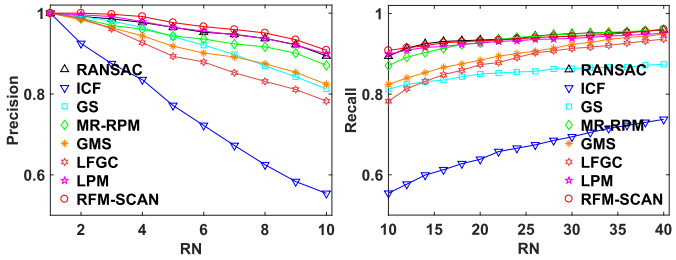


Fig. 7. Precision (left) and recall (right) with respect to RN, *i.e.*, the required number of images to be retrieved for a given image.

about 1475.60, 545.99, 69.03 and 445.34, respectively. For the precision-recall statistics, each scattered dot represents a precision-recall pair on an image pair. From the results, we see that RANSAC can produce satisfying results on all the four datasets. This is because we have used enough sampling times to obtain an outlier-free subset for transformation estimation even in case of low initial inlier ratio. ICF and GS usually have high precision or recall, but not simultaneously. MR-RPM works well on most image pairs, but may fail in case of low initial inlier ratio. GMS does not achieve satisfying performance, because we feed the same input as the other methods into GMS. We note that GMS was originally designed with a very large number of low-quality matches instead. LFGC typically achieves high precision but low recall. This is due to that its main goal is to identify good matches and accurately recover the transformation matrix between two point sets, which may falsely remove a set of unstable true matches, leading to a low recall. In addition, our testing data such as RS and Retina involving low-overlapped areas or non-rigid deformations are different from the training data of LFGC typically suffer from large scale or viewpoint changes, and

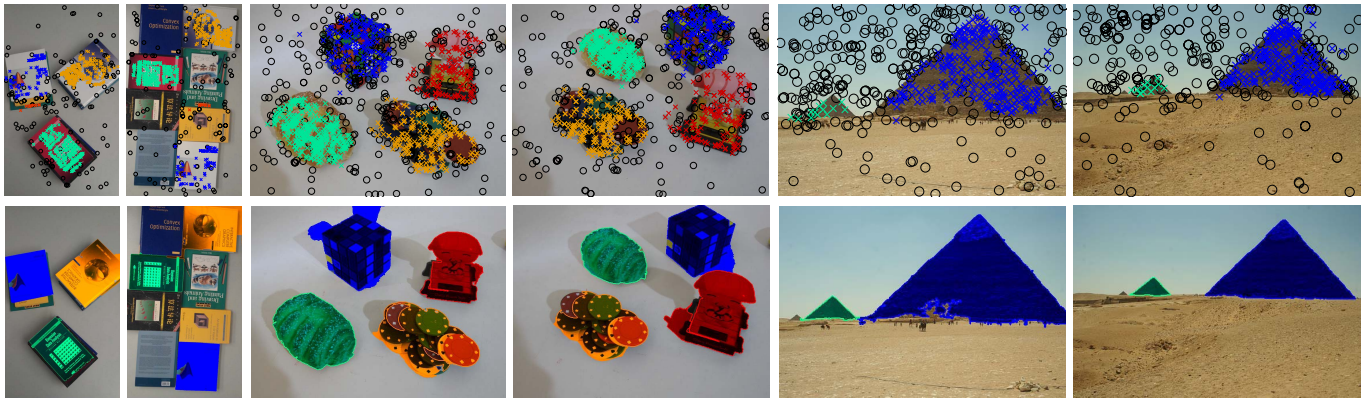


Fig. 8. Qualitative illustration of co-segmentation results of our RFM-SCAN on *Book* (*left*) in Fig. 1, *CBTC* (*middle*) taken from *AdelaideRMF* [41] and *Pyramid* (*right*) taken from *iCoseg* [42]. For each group of results, the top two images show the inlier clusters denoted by colored '×' and outliers denoted by 'o', while the corresponding co-segmentation results masked in different colors are demonstrated in the bottom two images.

the LFGC requires additional ground truth camera intrinsics as input for data normalization, which are not available in our testing data [26]. LPM has high recall, but its precision is badly degraded in case of low initial inlier ratio either. In contrast, our RFM-SCAN has the best precision-recall trade-off, where the scattered points are almost concentrated on the upper right corner. *Compared with LPM etc., our method can besides picking out true-positives from outliers, distinguish the matches automatically with respect to the geometrical clusters.* In addition, our RFM-SCAN is also quite efficient which only falls behind GMS, LFGC and LPM, and it typically requires only dozens of milliseconds to fulfill the mismatch removal task.

3) Robustness Test: As our iterative clustering strategy can promote the matching performance, in this section we test the robustness of our RFM-SCAN in case of extremely large outlier ratio and compare it to the aforementioned seven state-of-the-art methods. To this end, we choose two image pairs in Fig. 1 with relatively complex image transformations for evaluation, such as *Book* (4th) and *Church* (5th). The original numbers of inliers in these three image pairs are 565 and 71, respectively. For each image pair, we randomly remove or add outliers, so that the outlier ratio varies from 0.05 to 0.95 at an interval of 0.05. We then repeat the experiments 20 times and the average precision, recall and F-score at each outlier ratio are used to characterize the performance.

The statistical results are reported in Fig. 6. From the results, we see that the performance of all the eight methods degrades with the increase of outlier ratio. our RFM-SCAN has the best precision-recall trade-off, as our method has the best F-score curves in the third column, which is a comprehensive evaluation between precision and recall. *The F-scores of our RFM-SCAN are larger than* 0.85 *even that there are* 95% *outliers in the putative sets.* This demonstrates that the proposed method is robust and can handle a large number of outliers.

Note that the precision of our RFM-SCAN drops faster than the recall. The reason is that as the outlier ratio increases, a small part of outliers may have weak motion consistency and then form one or more false inlier clusters, leading to a decrease in precision. In contrast, the inliers in general always have motion consistency that are seldom affected by outliers, and hence it can achieve a large recall even in case of a large outlier ratio.

B. Applications

1) Near-Duplicate Image Retrieval: In this section, we apply our RFM-SCAN to solving vision-based tasks. First, we consider the near-duplicate image retrieval task. Given a query image, the goal is to retrieve the images of the same object or scene from a large database and return a ranked list. We choose the *California-ND* dataset [43] for evaluation. All the categories with 10 or more images are enlisted, and for each category 10 images are randomly selected for quantitative evaluation, resulting in $14,280$ image pairs in total. The matching algorithms on all the $14,280$ image pairs are executed and the number of preserved matches is employed to measure the similarity between two images. A ranked list for each given image according to its similarities with every other image in the dataset is returned. The performance is characterized by precision and recall based on the ranked lists. The precision is valid for $RN \leq 10$ and the recall is valid for $RN \geq 10$ with RN denoting the number of retrieved images.

The statistical results of eight methods are reported in Fig. 7. Our RFM-SCAN overall has the best performance, followed by LPM and RANSAC. In particular, for $RN \leq 10$, our method consistently has the best precision; while for $RN \geq 10$, RANSAC performs slightly better than our RFM-SCAN. The average retrieved correct image numbers of RANSAC, ICF, GS, MR-RPM, GMS, LFGC, LPM and our RFM-SCAN for $RN = 10$ are approximately 8.94, 5.54, 8.13, 8.71, 8.24, 7.83, 8.99, and 9.08, respectively.

2) Co-Segmentation: With the clusters generated by RFM-SCAN, it is easy to apply them to the co-segmentation task, which aims to segment common foreground objects from image pairs simultaneously by referring the jointly information of multiple images [44]. To achieve this goal, we first generate an initial segmentation result for each image using an existing super-pixel segmentation method such as linear spectral clustering [45]. We then consider the super-pixels containing no or few matches as background, and vice versa. For the

TABLE I
AVERAGE PRECISION (%) AND STANDARD DEVIATION (±%) OF CO-SEGMENTATION ON *iCoseg* DATASET. BOLD INDICATES BETTER RESULT

	CS-GMS	ICS-SCF	ICS-GLE	HOCS	RFM-SCAN
Pyramid	96.65 (1.84)	87.41 (10.45)	95.72 (3.52)	83.21 (20.96)	**97.33** (3.90)
Goose	88.94 (8.95)	87.43 (10.28)	96.21 (5.89)	96.39 (2.81)	**98.34** (1.46)
Helicopter	96.68 (4.05)	95.27 (6.96)	97.95 (2.43)	95.75 (5.33)	**97.96** (1.82)
Hot Balloon	96.85 (6.21)	96.41 (4.15)	97.71 (3.76)	85.40 (24.64)	**98.68** (1.92)
Sta-of-Lib	94.11 (3.49)	94.66 (2.05)	94.64 (5.37)	97.98 (2.18)	**98.12** (2.31)

foreground, as RFM-SCAN can generate a number of clusters, it can be divided into different objects automatically according to different clusters. Some typical results are reported in Fig. 8. In particular, there are three, four and two objects respectively which are masked with different colors. From the results, we see that the multiple objects can be identified as different clusters and hence are almost perfectly co-segmented.

To conduct a quantitative evaluation on this task, we choose the *iCoseg* [42] as our test dataset. It contains 38 groups and 643 images in total, and the ground truth segmentation masks are also provided along with the dataset. We select five representative groups including 57 images for evaluation, as shown in Table I. The segmentation precision, which is defined as the ratio of the correct segmented pixel number and total pixel number, is adopted to characterize the performance. We calculate the average precision and standard deviation to validate the final co-segmentation performance. Four state-of-art methods such as CS-GMS [46], ICS-SCF [47], ICS-GLE [48] and HOCS [49] are used for comparison. The statistical results are reported in Table I. Clearly, our RFM-SCAN is able to consistently achieve the best average precision.

V. CONCLUSION

In this paper, we have proposed a spatial clustering-based approach called *RFM-SCAN* for robust feature matching. It is able to adaptively cluster a set of putative matches into several inlier groups with motion consistency together with an outlier one in linearithmic time complexity. The major parameters are estimated adaptively and the number of clusters are determined automatically. The qualitative and quantitative results on general feature matching as well as two vision-based tasks have demonstrated the superiority of our strategy over the state-of-the-art methods.

REFERENCES

[1] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, "Locality preserving matching," *Int. J. Comput. Vis.*, vol. 127, no. 5, pp. 512–531, 2019.

[2] W.-Y. Lin *et al.*, "CODE: Coherence based decision boundaries for feature correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 34–47, Jan. 2018.

[3] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2828–2837.

[4] Y. Lin, Z. Lin, and H. Zha, "The shape interaction matrix-based affine invariant mismatch removal for partial-duplicate image search," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 561–573, Feb. 2017.

[5] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2666–2674.

[6] Z. Min, J. Wang, and M. Q.-H. Meng, "Robust generalized point cloud registration with orientational data based on expectation maximization," *IEEE Trans. Autom. Sci. Eng.*, to be published. doi: 10.1109/TASE.2019.2914306.

[7] Z. Min, J. Wang, and M. Q.-H. Meng, "Joint rigid registration of multiple generalized point sets with hybrid mixture models," *IEEE Trans. Autom. Sci. Eng.*, to be published. doi: 10.1109/TASE.2019.2906391.

[8] M. Daoudi, A. Srivastava, and R. Veltkamp, *3D Face Modeling, Analysis and Recognition*. Hoboken, NJ, USA: Wiley, 2013.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[10] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[11] J. Ma, J. Wu, J. Zhao, J. Jiang, H. Zhou, and Q. Z. Sheng, "Nonrigid point set registration with robust transformation learning under manifold regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published. doi: 10.1109/TNNLS.2018.2872528.

[12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96. 1996, pp. 226–231.

[13] A. Ghaffari and E. Fatemizadeh, "Image registration based on low rank matrix: Rank-regularized SSD," *IEEE Trans. Med. Imag.*, vol. 37, no. 1, pp. 138–150, Jan. 2018.

[14] J. Li, Q. Hu, M. Ai, and R. Zhong, "Robust feature matching via support-line voting and affine-invariant ratios," *ISPRS J. Photogramm. Remote Sens.*, vol. 132, pp. 61–76, Oct. 2017.

[15] M. Liu, C. Pradalier, and R. Siegwart, "Visual homing from scale with an uncalibrated omnidirectional camera," *IEEE Trans. Robot.*, vol. 29, no. 6, pp. 1353–1365, Dec. 2013.

[16] J. Zhao and J. Ma, "Visual homing by robust interpolation for sparse motion flow," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, pp. 1282–1288.

[17] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 138–156, 2000.

[18] X. Li and Z. Hu, "Rejecting mismatches by correspondence function," *Int. J. Comput. Vis.*, vol. 89, no. 1, pp. 1–17, 2010.

[19] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 1482–1489.

[20] L. Torresani, V. Kolmogorov, and C. Rother, "Feature correspondence via graph matching: Models and global optimization," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 596–609.

[21] M. Cho and K. M. Lee, "Mode-seeking on graphs via random walks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 606–613.

[22] H. Liu and S. Yan, "Common visual pattern discovery via spatially coherent correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1609–1616.

[23] J. Yan, M. Cho, H. Zha, X. Yang, and S. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1228–1242, Jun. 2016.

[24] J. Yan, J. Wang, H. Zha, X. Yang, and S. Chu, "Consistency-driven alternating optimization for multigraph matching: A unified approach," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 994–1009, Mar. 2015.

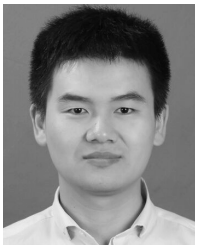
[25] J. Yan, C. Li, Y. Li, and G. Cao, "Adaptive discrete hypergraph matching," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 765–779, Feb. 2018.

[26] J. Ma, X. Jiang, J. Jiang, J. Zhao, and X. Guo, "LMR: Learning a two-class classifier for mismatch removal," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4045–4059, Aug. 2019.

[27] L. Zhou, S. Zhu, T. Shen, J. Wang, T. Fang, and L. Quan, "Progressive large scale-invariant image matching in scale space," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2362–2371.
[28] Á. P. Bustos and T.-J. Chin, "Guaranteed outlier removal for point cloud registration with correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2868–2882, Dec. 2018.
[29] H. Deng, T. Birdal, and S. Ilic, "PPFNet: Global context aware local features for robust 3D point matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 195–205.
[30] J. Vongkulbhisal, B. I. Ugalde, F. De la Torre, and J. P. Costeira, "Inverse composition discriminative optimization for point cloud registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2993–3001.
[31] P. Speciale, D. P. Paudel, M. R. Oswald, H. Riemenschneider, L. Van Gool, and M. Pollefeys, "Consensus maximization for semantic region correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7317–7326.
[32] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.
[33] L. Rokach and O. Maimon, "Clustering methods," in *Data Mining and Knowledge Discovery Handbook*. Boston, MA, USA: Springer, 2005, pp. 321–352.
[34] J. A. Hartigan and M. A. Wong, "A K-means clustering algorithm," *Appl. Stat.*, vol. 28, no. 1, pp. 100–108, 1979.
[35] J. D. Banfield and A. E. Raftery, "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, vol. 49, no. 3, pp. 803–821, 1993.
[36] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[37] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2000, vol. 29, no. 2, pp. 93–104.
[38] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.
[39] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.
[40] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 153–168, 2016.
[41] H. S. Wong, T.-J. Chin, J. Yu, and D. Suter, "Dynamic and hierarchical multi-structure geometric model fitting," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1044–1051.
[42] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "iCoseg: Interactive co-segmentation with intelligent scribble guidance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3169–3176.
[43] A. Jinda-Apiraksa, V. Vonikakis, and S. Winkler, "California-ND: An annotated dataset for near-duplicate detection in personal photo collections," in *Proc. QoMEX*, Jul. 2013, pp. 142–147.
[44] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching—Incorporating a global constraint into MRFs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 993–1000.
[45] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1356–1363.
[46] K. R. Jerripothula, J. Cai, F. Meng, and J. Yuan, "Automatic image co-segmentation using geometric mean saliency," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 3277–3281.
[47] K. R. Jerripothula, J. Cai, and J. Yuan, "Image co-segmentation via saliency co-fusion," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1896–1909, Sep. 2016.
[48] X. Dong, J. Shen, L. Shao, and M.-H. Yang, "Interactive cosegmentation using global and local energy optimization," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3966–3977, Nov. 2015.
[49] W. Wang and J. Shen, "Higher-order image co-segmentation," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1011–1021, Jun. 2016.

Xingyu Jiang received the B.E. degree from the Department of Mechanical and Electronic Engineering, Huazhong Agricultural University, Wuhan, China, in 2017. He is currently pursuing the Ph.D. degree with the Electronic Information School, Wuhan University. His research interests include computer vision, machine learning, and pattern recognition.

Jiayi Ma received the B.S. degree in information and computing science and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2014, respectively. From 2012 to 2013, he was an Exchange Student with the Department of Statistics, University of California at Los Angeles, Los Angeles, CA, USA. He is currently a Professor with the Electronic Information School, Wuhan University, Wuhan. He has authored or coauthored over 110 refereed journal and conference papers, including the IEEE TPAMI/TIP/TSP/TNNLS/TIE/TGRS/TCYB/TMM/TCSVT, IJCV, CVPR, ICCV, IJCAI, AAAI, ICRA, IROS, and ACM MM. His current research interests include the areas of computer vision, machine learning, and pattern recognition. He was a recipient of the Natural Science Award of Hubei Province (first class) as the first author, the Chinese Association for Artificial Intelligence (CAAI) Excellent Doctoral Dissertation Award (a total of eight winners in China), and the Chinese Association of Automation (CAA) Excellent Doctoral Dissertation Award (a total of ten winners in China). He is an Editorial Board Member of *Information Fusion* and *Neurocomputing*, and a Guest Editor of *Remote Sensing*.



Junjun Jiang received the B.S. degree from the Department of Mathematics, Huaqiao University, Quanzhou, China, in 2009, and the Ph.D. degree from the School of Computer, Wuhan University, Wuhan, China, in 2014. From 2015 to 2018, he was an Associate Professor with the China University of Geosciences, Wuhan. Since 2016, he has been a Project Researcher with the National Institute of Informatics, Tokyo, Japan. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. His research interests include image processing and computer vision. He was a recipient of the Finalist of the World's FIRST 10K Best Paper Award at ICME 2017, the Best Student Paper Runner-up Award at MMM 2015, the 2016 China Computer Federation (CCF) Outstanding Doctoral Dissertation Award, and the 2015 ACM Wuhan Doctoral Dissertation Award.

Xiaojie Guo received the B.E. degree in software engineering from the School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China, in 2008, and the M.S. and Ph.D. degrees in computer science from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in 2010 and 2013, respectively. He is currently an Associate Professor with the School of Computer Software, Tianjin University. He was a recipient of the Piero Zamperoni Best Student Paper Award in the International Conference on Pattern Recognition (International Association on Pattern Recognition) in 2010.