Progressive Mode-Seeking on Graphs for Sparse Feature Matching

Chao Wang[1], Lei Wang[1], and Lingqiao Liu[2]

[1] School of Computer Science & Software Engineering
University of Wollongong, NSW, Australia
[2] School of Computer Science, University of Adelaide, Australia

Abstract. Sparse feature matching poses three challenges to graph-based methods: (1) the combinatorial nature makes the number of possible matches huge; (2) most possible matches might be outliers; (3) high computational complexity is often incurred. In this paper, to resolve these issues, we propose a simple, yet surprisingly effective approach to explore the huge matching space in order to significantly boost true matches while avoiding outliers. The key idea is to perform mode-seeking on graphs progressively based on our proposed guided graph density. We further design a density-aware sampling technique to considerably accelerate mode-seeking. Experimental study on various benchmark data sets demonstrates that our method is several orders faster than the state-of-the-art methods while achieving much higher precision and recall.

Keywords: Feature matching, Mode-seeking.

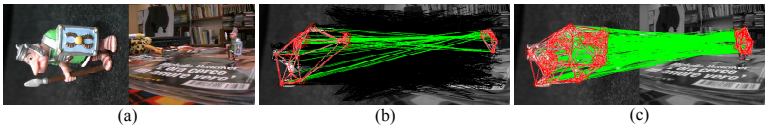
1 Introduction

Matching sparse features between two images is a longstanding research problem for a variety of applications in computer vision, such as motion estimation, object recognition, image retrieval and 3D reconstruction [8]. Since the matches have meaningful interrelations and structures, they are often used to construct an association graph in which graph nodes represent candidate matches while graph edges represent relationships between them. As a result, feature matching is modeled as a node selection problem in an association graph. Although there are many other feature matching algorithms, we restrict ourselves to the ones based on the association graph in this paper.

There have been a myriad of algorithms proposed to address this problem, ranging from classical methods like graph matching [8,2,10,16,22] and hypergraph matching [23,9,15], through various agglomerative clustering approaches [1,24], to recently popular mode-seeking methods [17,14,3,4]. While tremendous progress has been made, current methods are still far from being practical when dealing with many real-world images due to three challenges.

(1) The combinatorial nature makes the matching space of all the possible matches huge. Let n_1 and n_2 denote the numbers of sparse features of two images P and Q respectively, there are $n_1 \times n_2$ possible candidate matches. Generally,

D. Fleet et al. (Eds.): ECCV 2014, Part II, LNCS 8690, pp. 788–802, 2014.



(a) (b) (c)

Fig. 1. (a) Input images P and Q. (b) Two types of feature detectors (MSER [19] and Harris-Affine [20]) extract 2539 SIFT features from image P and 3013 SIFT features from image Q. The 2539 matches produced by SIFT feature matching include only 64 true matches shown with green lines and 2475 false matches shown with black lines. (c) All the 281 true matches. Since there are 2539×3013 possible matches, the probability of true matches is much less than 1%.

we have $n_1, n_2 > 1000$ and $n_1 \times n_2 > 1000,000$. Building a full association graph G^F of millions of nodes is not tractable. To address this issue, most methods establish candidate matches by using discriminative features, such as SIFT [18], at a relatively low cost. However, those candidate matches usually include only a small portion of all the true matches, as shown in Fig.1.

(2) Most possible matches might be outliers. For many real-world image pairs, there are only several hundreds of true matches which account for less than 1% of the total candidate matches, as shown by Fig.1. To detect the inlier nodes from the full association graph is like looking for a needle in a haystack.

There have been a few attempts to handle the outliers. A popular method is to first solve an affine transform by using RANSAC, and then remove outliers with the affine. This naive scheme often fail when there exist significant outliers, non-rigid transforms or many-to-many object correspondences. Graph-matching methods [8,2,10,16,22] impose pair-wise constraints and the hyper-graph matching methods [23,9,15] impose high-order constraints (e.g., projective invariance) on graph nodes. The nodes that do not satisfy those constrains are considered as outliers. These methods work well for rigid transformations but perform poorly in the case of large non-rigid motions. Agglomerative clustering methods [1,24] cluster nodes with a bottom-up aggregation strategy and filter out outlier clusters with small sizes. Such methods are based on a set of heuristic rules and therefore global optimum often cannot be achieved. Recently, mode-seeking methods [17,14,3,4] have received a lot of attentions because they have appealing advantages over other techniques: the structure of the clusters may be rather arbitrary [4], the number of clusters does not need to be known in advance, and the convergence can be guaranteed. They assume that inlier nodes for true matches have larger graph density [17](or authority[4]) than outliers, and remove outliers by eliminating the clusters with small density (or authority). They work very well for the small graph constructed from the SIFT feature matching. When exploring the huge matching space which includes all possible matches, however, they might completely fail because the probability of true matches can be so small (e.g.,$< 1\%$) that their assumptions do not hold anymore.

(3) High computational complexity is often incurred. A common starting point for constructing an association graph is the computation of the similarity matrix. Its time and space complexity is $O(N^2)$ with N denoting the number of graph

nodes ($N = n_1 \times n_2$ in the full graph). Therefore the computational costs in both time and memory are huge for large graphs. In addition to the similarity matrix construction, many other steps involved in existing methods are also expensive in terms of computational cost and memory usage. For examples, finding the principal eigenvectors of the similarity matrix in [16], high-order power iterations in [9], bottom-up building clusters in [1,24], shifting among the power set of a given graph in [17,14], and computing the PageRank matrix in [3] all take at least $O(N^2)$ time and memory. Then the usefulness of these techniques on large graphs is hampered by the high complexity. Several techniques have been developed to reduce the complexity of the classical mode-seeking method[7][13][21][12]. However, they are restricted to traditional data representation as points in a metric feature space, and it is very difficult to adapt them to graph representation. As far as we know, no work has been designed to speed up mode-seeking on graphs.

The above three challenges make detection of all the true matches in the huge matching space extremely difficult. To resolve these challenges, we propose a simple, yet surprisingly effective approach to explore the huge matching space in order to significantly boost true matches while avoiding outliers. The key idea is to perform mode-seeking on graphs progressively. Our method, called the progressive mode-seeking algorithm (PMA), starts from a small graph built by the matches obtained based on SIFT distances as in [1], and then explores a huge matching space in a progressive manner. The high performance of PMA comes from our proposed guided graph density (GGD). Totally different from the traditional graph density [17][3][4] which is calculated based on a single graph, our GGD of a node in one graph is calculated based on another reference graph. More specifically, the GGD of a node in a huge graph is calculated based on a small clean graph which mainly includes true matches. This leads the GGD values of outliers to nearly zero, and therefore makes mode-seeking much more robust to outliers even in a huge matching space. To reduce the complexity, we further design a density-aware sampling technique to considerably accelerate mode-seeking. The resultant method has a time complexity linear in the number of graph nodes.

Our PMA is inspired by the progressive graph matching (PGM) method [5] which performs graph matching progressively. PGM can greatly boost the number of true matches. However, it fails to handle many-to-many object correspondences due to its single cluster assumption, and tends to introduce many outliers because graph matching results are often noisy. Different from PGM, our PMA performs mode-seeking in a progressive manner. It excels in handling many-to-many object correspondences because each cluster of matches naturally corresponds to one object correspondence. Furthermore, it successfully avoids introducing many outliers by suppressing their graph density values.

PMA works well on a very wide variety of images. Experimental study on several benchmark data sets shows that it is several orders faster than the state-of-the-art mode-seeking methods on images with thousands of features, while producing much higher precision and recall.

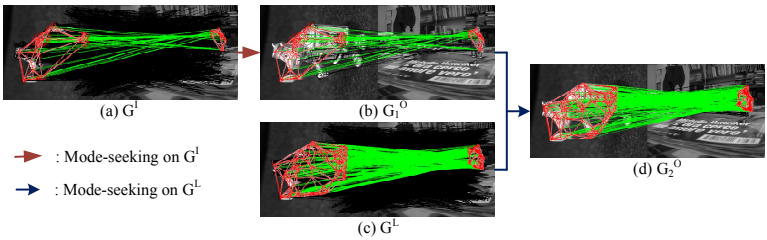


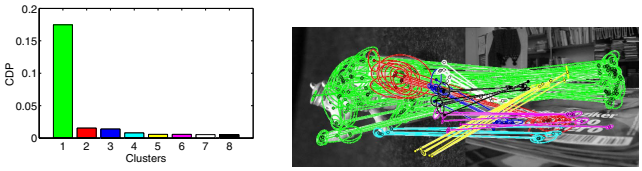
Fig. 2. The framework of our PMA. It performs mode-seeking on G^I to produce G_1^O, and mode-seeking on G^L based on G_1^O to produce G_2^O. G^I is a small graph obtained by the matches obtained based on SIFT distances as in [1], and G^L is a much larger graph covering most true matches. (a) G^I contains 64 true matches and 2475 false matches. (b) G_1^O contains 38 true matches and 21 false matches. (c) G^L contains 252 true matches and 101308 false matches. (d) G_2^O contains 233 true matches and 48 false matches.

To summarize, this paper has three main contributions. Firstly, we propose a novel way to compute graph density which enables a progressive framework for robustly exploring the huge matching space. Secondly, we bring forward a density-aware sampling technique to significantly speed up mode-seeking on graphs. The third is that we design a novel mode-seeking method for clustering graph nodes in order to solve for sparse feature matching.

2 Algorithm

Following [2,4,16,18], an association graph is defined as $G = (V, E, W)$ which consists of nodes V, edges E and attributes W. $\omega(i, j) \in W$ is the attribute of edge $e(i, j) \in E$, characterizing similarity between node v_i and node v_j. In this paper, we use N, the number of nodes, to denote the size of graph G.

Fig.2 shows the progressive framework of our PMA. First, n_1 and n_2 salient features are extracted from two input images P and Q respectively with multiple types of detectors. N_I candidate matches are then established by the matches obtained based on SIFT distances as in [1], and are taken as the nodes of a small initial graph G^I. We also build a much larger graph G^L and ensure that it covers most of all the true matches. Here, G^L is much smaller than the full graph G^F. Different from other methods, we do not compute the full similarity matrix W. Instead, we only compute the similarity between each node in G^L and each node randomly sampled in G^L, as will be detailed later. Second, we perform mode-seeking on the small graph G^I similar to [17,14,3,4] in order to detect the inlier clusters G_1^O. We find out that this kind of method works well because SIFT distance at low cost can increase the probability of true matches greatly. Finally, we perform mode-seeking on the large graph G^L guided by G_1^O, producing graph G_2^O. A density-aware sampling technique is proposed to considerably accelerate the mode-seeking process. We can further run mode-seeking



(a) (b)

Fig. 3. (a) Top 8 max CDP values. (b)The clusters for top 8 max CDP values. The inlier clusters (denoted with green lines) have CDP values significantly larger than those of the outlier clusters.

on G^L iteratively guided by G_2^O to detect more true matches. In most cases, the first iteration brings significant performance improvement. For efficiency we use only two iterations which already produce satisfying results.

2.1 Mode-Seeking on G^I

Recent methods [3,17,4] define the graph density based on node characteristics such as the probability of visits by the random walker, and are therefore quite different from the classical kernel density estimate (KDE)[7] defined in a metric space. The high computational complexity hampers their usefulness for huge matching spaces. Differently, we define the graph density similar to KDE by representing the kernel on a joint domain. This frames a graph-based analogue to the classical KDE, and therefore makes the fast methods designed to accelerate KDE applicable to the graph density, as will be shown. Our graph density at node v_j is defined as

$$f(j) = \frac{1}{N_I} \sum_{i}^{N_I} K(i,j) \tag{1}$$

with $K(i,j) = g(d_S(i,j), h_S) g(d_G(i,j), h_G)$. $g(d,h)$ is a Gaussian function of d with h denoting the standard deviation. d_S is the Euclidean distance in spatial domain on the image. d_G is the node distance in graph domain and is set as the Symmetric Transfer Error (STE) used in [5,15,1,4]. h_S and h_G denote the kernel bandwidths which determine the resolution of the mode detection. Here we set $h_G = 20$ and $h_S = H/10$ with H^2 denoting the image size.

As pointed out by [7], a truncated Gaussian kernel always provides satisfactory performance, that is, only the nearest neighbors $v_i \in \Omega(j)$ are adopted to calculate the graph density. Let $\Omega(j) = \{v_i \in G^I | d_S(i,j) \leq \gamma h_S, d_G(i,j) \leq \gamma h_G\}$ with $\gamma = 2$, Eq.(1) becomes $f(j) = \frac{1}{n_j} \sum_{v_i \in \Omega(j)} K(i,j)$ with n_j denoting the size of $\Omega(j)$. To efficiently find $\Omega(j)$, we firstly use axis-aligned box windows [7] to obtain the nearest neighbors in the spatial domain, and then test each one in the graph domain.

Similar to the classical methods [7], our mode-seeking is achieved by shifting each node to the local mode in which the local maximum of graph density is attained. The node-shifting $NS(j)$ of v_j is formulated as

$$NS(j) = \arg \max_{v_i \in \Omega(j)} p(j,i)(f(i) - f(j)), \tag{2}$$

where $p(j,i) = K(j,i)/\sum_{v_i \in \Omega(j)} K(j,i)$ denotes the probability of the transition from node v_j to node v_i. $NS(j)$ refers to the neighboring node of v_j with the highest expected graph density increment. Therefore node-shifting is the steepest ascent over the graph density within $\Omega(j)$. Similar to other mode-seeking methods [14,4,3], ours is guaranteed to converge, as proved below.

Theorem 1. A finite sequence of node-shifting from any node converges to a graph density mode.

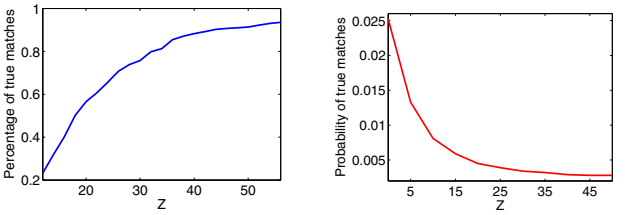
Proof. Since $\Omega(j)$ of any node v_j includes itself, the graph density values of a sequence of shifts from v_j keep strictly increasing until the shifts attain a node whose node-shifting is itself. The final node, therefore, is the density mode $DM(j)$, and the length of the sequence is the graph size N_I at most.

Starting from any node, successive shifts progress toward its graph density mode. The shifting trajectory of nodes sharing a common density mode builds a tree, and leads to a natural cluster. For each node, we only need to compute its node-shifting once. This makes the next node-shifting for any node already exist. Then the cluster label of all nodes associated with each disjoint tree can be assigned in a single tree traversal.

As observed by [17,4], the nodes for true matches usually have larger graph density values than outliers for the graph built by the matches obtained based on SIFT distances[1]. So we can utilize this observation to detect outlier clusters in the small graph G^I. We define the cluster density of each cluster as the sum of the graph density values of its members, and its cluster density percentage (CDP) as the ratio between its cluster density and the sum of all the nodes' graph density values. According to the above observation, the outlier clusters usually have small cluster densities and therefore have small CDP. So CDP provides a reliable measure for detecting and eliminating outliers, as shown in Fig.3. We remove outlier clusters whose CDP is less than a small threshold $t = 0.03$. The final output is the inlier clusters of nodes which compose graph G_1^O with size N_1^O.

2.2 Mode-seeking on G^L

We build G^L by using top Z matches for each feature based on the SIFT feature distances. By testing the ETHZ toys dataset[11], we plot the percentage of all the true matches that G^L includes as a function of Z in Fig.4(a). It can be seen that over 90% true matches for each image pair are included in G^L when $Z = 40$. This suggests that we can perform the mode-seeking on G^L rather than on the full graph G^F to achieve great complexity reduction since the size of G^L is only Zn_1 as opposed to $n_1 \times n_2$ for G^F.



(a) (b)

Fig. 4. (a) The percentage of all the true matches included by G^L as a function of Z. (b) The probability of true matches in G^L as a function of Z. $Z = 1$ in SIFT matching[18].

We also give the probability of the true matches in G^L as a function of Z in Fig.4(b). As can be seen, the probability attains the maximum when $Z = 1$. This indicates that SIFT distance ($Z = 1$) at low cost can greatly increase the probability of true matches. When Z become very large ($Z > 40$), the probability of true matches gradually reduces to a small value (0.0028) and keeps almost unchanged. This is because most matches are overlapped by each other for large Z. It further verifies that exploring the full graph G^F might be unnecessary.

For each node v_j in G^L, we define its **guided graph density (GGD)** as $f(j) = \frac{1}{n_j} \sum_{v_i \in \Omega(j, G^L, G_1^O)} K(i, j)$ with

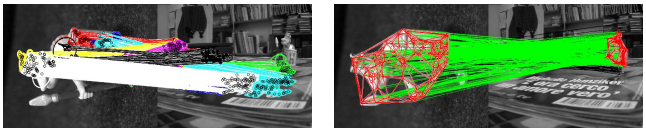
$$\Omega(j, G^L, G_1^O) = \{v_i \in G_1^O | v_j \in G^L, d_S(i, j) \leq \gamma h_S, d_G(i, j) \leq \gamma h_G\} \quad (3)$$

Our GGD searches for Ω in another reference graph, i.e., $\Omega \subset G_1^O$. This is in sharp contrast with the traditional methods in which $\Omega \subset G^L$. As can be seen in Fig.2(b), G_1^O is mainly composed of inliers. Then the GGD values of most outliers in G^L become nearly zero because: (1) the d_S and d_G between outliers and inliers are often very large, and (2) the nodes of G_1^O might be far away from many outliers so that the nearest neighbor set Ω of many outliers are empty. Many mode-seeking methods fail when exploring huge matching space because their assumption that inlier clusters often have larger graph density than outliers does not hold when the probability of inliers is very low, say the case in G^L. Our GGD solves this problem nicely because it makes the assumption hold again by suppressing the graph density of outliers. Then we can perform mode-seeking based on GGD to detect and eliminate outliers, as done in Section 2.1. The output N_2^O nodes compose a G_2^O. Fig.5 shows the impact of GGD.

The mode-seeking on G^L is guaranteed to converge, and the proof is the same as that for Theorem 1. So our PMA is guaranteed to converge.

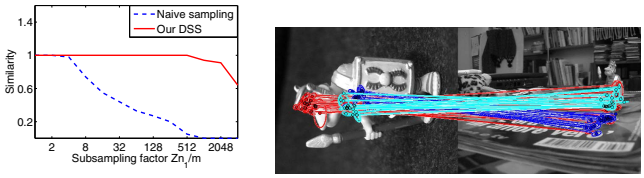
2.3 A Density-Aware Sampling Technique

To accelerate the classical mode-seeking method, D.Freedman et al.[12] approximate the whole feature space by using a greatly reduced number of points



(a) (b)

Fig. 5. (a) Mode-seeking result on graph G^L with $\Omega \subset G^L$. None of the top 8 clusters is inlier cluster. Each color indicates on cluster of matches. (b) Mode-seeking result on graph G^L with GGD ($\Omega \subset G_1^O$). The inlier cluster is detected in a clean manner.



(a) (b)

Fig. 6. Impact of our density-aware sampling (DAS) technique. (a)Two kinds of similarities as functions of sub-sampling factor Zn_1/m are shown. One similarity is between the result by original mode-seeking method (without sampling) and the result by the naive sampling method. The other similarity is between the result by original mode-seeking method (without sampling) and the result by our density-aware sampling (DAS) method. (b)The result by the naive sampling method with $m = 200$. Three outlier clusters are detected. The result by our DAS with $m = 200$ is given in Fig.2(d).

randomly sampled from the distribution defined by KDE. The speed-up is proportional to the sub-sampling factor. However, different from sampling metric feature space for KDE, directly sampling the graph density in Eq.(1) can not produce graph node samples. A naive sampling method to solve this issue is to randomly sample from the set of graph nodes. Since most graph nodes might be outliers, the number of samples need to be sufficiently large in order to cover the modes of inliers with a large probability. Then the complexity reduction will be limited. To solve this problem, we propose a simple approach called the density-aware sampling (DAS) technique which samples graph nodes according to the graph density in mode-seeking on G^I (or the GGD in mode-seeking on G^L). In DAS, the probability of accepting v_i is $f(i)/\sum_i f(i)$. As mentioned before, inliers often have larger graph density values (or GGD values) than outliers, so DAS tends to sample more inliers, thus solving the problem of the naive sampling method nicely.

We use DAS to sample m nodes of G (G can be G^I and G^L) to approximate G, obtaining graph G^{sample}. Then instead of computing modes directly on G, we perform mode-seeking on G^{sample} to obtain the density modes. We further map backwards from each node v_i in G to the closest sample s_{i*} by $s_{i*} = \arg\max_{s_i \in \Omega(i)} K(i, s_i)$ with $\Omega(i) \subset G^{sample}$. Finally, we set the graph density mode $DM(i)$ of v_i to $DM(s_{i*})$ in order to obtain the final clustering results.

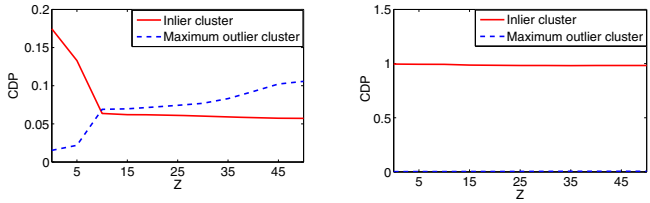
We adopt DAS in the mode-seeking on both G^I and G^L. When performing it on G^L, we randomly sample $R = min(m, N_1^O)$ nodes from G_1^O which is the mode-seeking result on G^I to further reduce the complexity. The details are given in Algorithm 1.

To show the impact of our DAS, we measure the similarity between the result obtained by the original mode-seeking method and those obtained by two sampling techniques: the naive sampling method and our DAS. The similarity between any two graphs G^1 and G^2 is calculated by the normalized intersect $|G^1 \cap G^2|/|G^1 \cup G^2|$ with $|G|$ denoting the size of G. Fig.6(a) plots the similarity as a function of the sub-sampling factor Zn_1/m. As can be seen, DAS greatly reduces the required sample number m, thus leading to a significant speed-up Zn_1/m. Generally, we set $m = 200$ to get $Zn_1/m > 500$. The result obtained by the naive sampling method with $m = 200$ is shown in Fig.6(b), and that obtained by our DAS is given in previous Fig.2(d). The improvement by our DAS can be clearly seen.

Algorithm 1: mode-seeking on G^I (G^L)
Input G^I (G_1^O and G^L)
Output G_1^O (G_2^O)
1. Sampling: sample m nodes of G^I
(sample m nodes of G^L and R nodes of G_1^O)
2. Node-shifting: solve for the node-shifting for all the m matches with Eq.(2)
3. Tree traversal: assign each sample s_i to its density mode $DM(s_i)$ by a tree traversal along $NS(s_i)$
4. Map backwards: for each node v_i, find s_{i*} by $\arg\max_{s_i \in \Omega(i)} K(i, s_i)$, then let $DM(i) = DM(s_{i*})$
5. Remove outliers: compute CDP for each cluster, and remove outlier clusters with $CDP < t$

3 Analysis

Complexity. In mode-seeking on G^I, in order to cluster m samples, we need to search for the nearest neighbors $\Omega(j)$ for each node v_j. Assuming that the matches are uniformly distributed in image, the time complexity for finding $\Omega(j)$ is $O(mr)$ by using the axis-aligned box windows, with $r = (2\gamma h_S)^2/H^2$ denoting the area ratio of the region we explore to the whole image. Then computing the graph density for all the m nodes takes $O(m^2 r)$ time, and node-shifting also takes $O(m^2 r)$ time. Mapping backwards takes $O(N_I mr)$ time. Therefore the time complexity of mode-seeking on G^I reaches $O(2m^2 r + N_I mr) \approx O(N_I mr)$ since $N_I \gg m$. In mode-seeking on G^L, searching for $\Omega(j)$ on graph G_1^O takes $O(Rr)$ time, and clustering the m samples takes $O(2Rmr)$ time. Mapping backwards takes $O(Zn_1 mr)$ time. Then the time complexity is about $O(Zn_1 mr)$ since $Zn_1 \gg R$. So the total time complexity of our PMA is $O(N_I mr + Zn_1 mr) \approx O(Zn_1 mr)$ which is linear in the number of graph nodes Zn_1.



(a) (b)

Fig. 7. The effect of our guided graph density (GGD) ($\Omega \subset G_1^O$) on eliminating outliers. (a) The result by mode-seeking with $\Omega \subset G^L$. With the increase of outliers, the max CDP for outlier clusters becomes much larger than the CDP for the inlier cluster. (b) The result by our mode-seeking with $\Omega \subset G_1^O$. The CDP for the inlier cluster is nearly independent of the number of outliers, and accounts for the majority of all the graph density values. No outlier cluster whose CDP is larger than t is detected.

Revisiting the literature on the subject, mode-seeking by using the methods [17,14,3,4] takes $O((Zn_1)^2)$ time on graph G^L. So the speed-up by our PMA is a factor of $(Zn_1)^2/(N_I mr + Zn_1 mr)$. In practical cases of interest, n_1 is about 2000, and N_1 is set to 2000. Then we have $(Zn_1)^2/(N_I mr + Zn_1 mr) \approx 2439$. So this is quite an impressive speed-up.

Effect of our GGD. Taking the image pair in Fig.1(a) as an example, we gradually increase the percentage of outliers by increasing Z. Fig.7 shows the results by mode-seeking with $\Omega \subset G^L$ and with our GGD ($\Omega \subset G_1^O$). With the increase of outliers, the max CDP for outliers becomes larger than the CDP for the inlier cluster. This is why many existing mode-seeking methods fail. By using GGD, no outlier cluster is detected at all because the GGD values of outliers become nearly zero. Then the CDP for the inlier cluster accounts for the majority of all the graph density values, and is almost independent of the number of outliers, as shown in Fig.7(b).

Limitations. Our method starts from mode-seeking on a small graph G^I built the matches obtained based on SIFT distances as in [1], and assumes that the nodes for true matches usually have larger graph density values than outliers in the small graph G^I. Although this assumption has been widely used [17,4], we did find that it does not hold for two extreme cases: large smooth regions such as white plates with little texture, and tiny-sized objects with few features extracted. To handle these cases will be our future work.

4 Experiments

We compare our PMA with three leading graph-based feature matching methods: the progressive graph matching (PGM) [5], the agglomerative correspondence clustering (ACC) [1] and the mode-seeking via random walk (RRWM) [4]. PGM improves the true matches progressively by exploring the full matching space

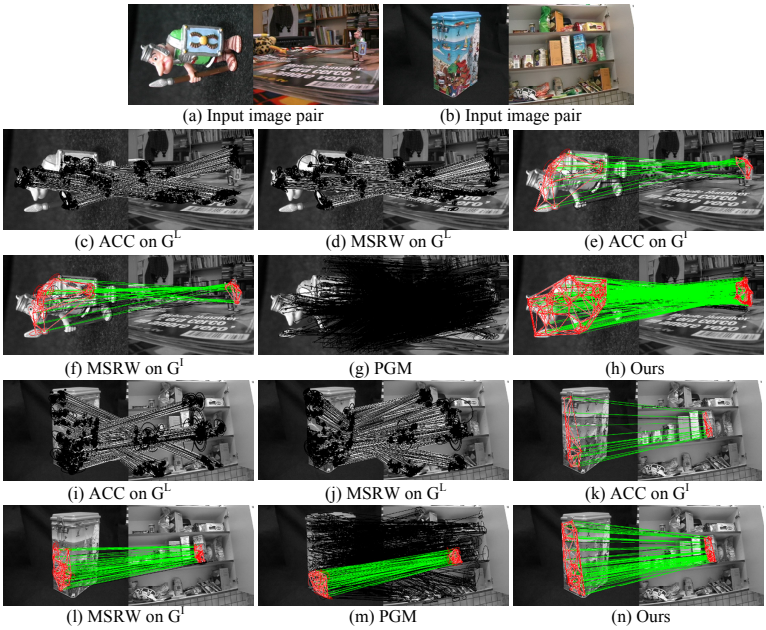


Fig. 8. Comparison on two image pairs of ETHZ toys dataset [11]. True matches are shown with green lines and outliers are shown with black lines.

based on an initial graph. We set G^I as its initial graph as done by our PMA. Both ACC and RRWM are only based on the similarity matrix of the association graph. We perform ACC and RRWM on graph G^L as done by PMA. We also run ACC and RRWM on graph G^I as their original work did for comparison. The algorithms of [17,3] were not compared in this paper because the source code provided by the authors on the internet produce 'out of memory' problem when handling our data sets.

We tested the above methods on three challenging benchmark data sets: ETHZ toys dataset [11], Co-recognition dataset [6] and Intra-class dataset [1]. ETHZ toys dataset includes 9 different rigid/non-rigid object pairs with significant transforms and clutters. Co-recognition dataset contains 6 image pairs with

Table 1. Precision/Recall (%) by four methods. ACC and RRWM denote their results on graph G^I. Since both ACC and RRWM fail to handle G^L, we do not show their results on G^L.

Data sets	ACC[1]	RRWM[4]	PGM[5]	Our PMA
ETHZ	63/15	66/23	11/66	**82/74**
Co-recognition	67/68	69/75	13/69	**88/79**
Intra-class	71/24	52/22	25/81	**72/83**

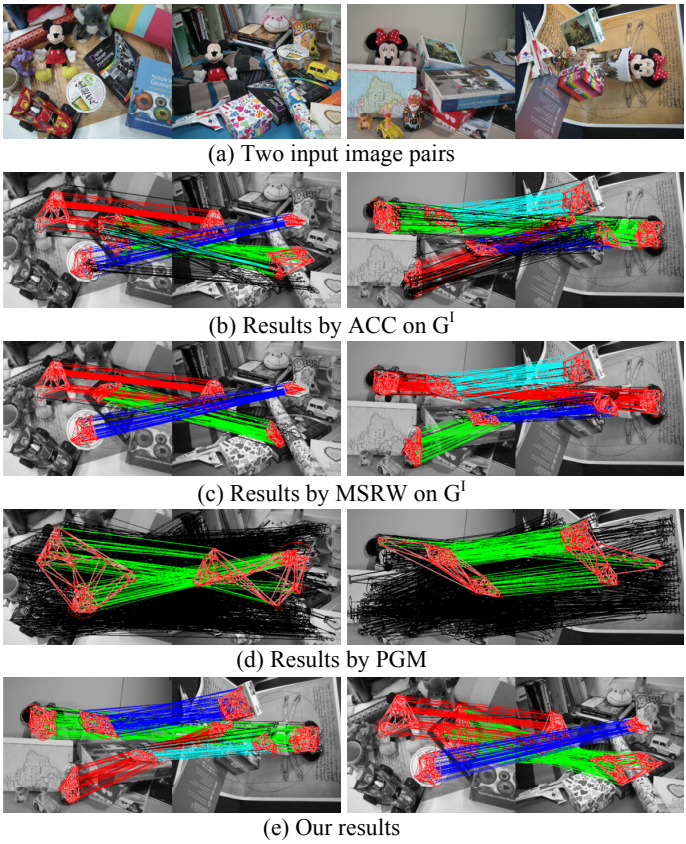


Fig. 9. Comparison on two image pairs of Co-recognition dataset [6]

complex many-to-many object correspondences. Intra-class dataset consists of 30 image pairs of large intra-class variation. The ground truth feature matches are manually constructed for each image pair to enable quantitatively evaluation. Here, we use the MSER[19] and the Harris affine detectors [20] with SIFT descriptor [18]. Our testing environment is MS Windows 7 Professional with Intel Core i5-3550 CPU 3.3GHz, 16GB RAM.

Fig.8 shows the matching results for two examples of ETHZ toys dataset. A prominent phenomenon observed is that the results on graph G^L by both ACC and RRWM are totally cluttered by the background outliers, with precisions and recalls close to zero. Actually, this happens for each image pair in the data sets. In the following parts we will not show the results on graph G^L by ACC or RRWM. The results obtained by both ACC and RRWM on graph G^I are much better, with about 20% of all the true matches detected. PGM fails to

handle these two examples due to the significant outliers. In comparison, our PMA successfully detects about 80% of all the true matches with much higher precisions, despite large object transforms and outliers.

Fig.9 demonstrates a comparison on two challenging image pairs in Co-recognition data set. These image pairs have three and four object correspondences respectively. Since PGM assumes that the true matches belong to a single cluster, it can not deal with many-to-many object correspondences. Therefore the results on these two examples are far from being satisfactory. ACC and RRWM accurately detect the three inlier clusters but also introduce many outliers. Our PMA solves this problem effectively by successfully distinguishing the inliers from outliers.

Fig.10 illustrates feature matching for two image pairs in Intra-class data set. To solve this, it is required to address appearance difference as well as intra-class variation. As can be seen, PGM cannot distinguish true matches from outliers. Both ACC and RRWM fail to recover most inlier matches. In contrast, the result by our PMA has much more inlier matches with less outliers.

Table 1 gives quantitative results on the three benchmark data sets. As mentioned before, since both ACC and RRWM fail to handle G^L and the precisions are nearly zero, we do not show their results. As can been seen, our PMA largely outperforms the other methods in both precision and recall.

Table 2. Average running time (in second) by four methods. ACC and RRWM denote their running time on graph G^L.

Data sets	ACC[1]	RRWM[4]	PGM[5]	Our PMA
ETHZ	245257	16462	23	**6**
Co-recognition	264379	19873	28	**7**
Intra-class	673	65	10	**2**

Table 2 gives the computational time. For ETHZ toys dataset [11] and Co-recognition dataset [6], our PMA is more than 1000 times faster than the leading mode-seeking method RRWM on graph G^L. We also test RRWM on the small graph G^I. The average time to process each image pair in the three data sets is 4, 6 and 2 seconds respectively. As shown in Table 2, our PMA has comparable performance. Compared with the ACC and PGM methods which are not mode-seeking methods, our PMA is also much faster. For Intra-class dataset [1], the feature numbers of most images are very small (in some cases, even less than fifty). Therefore the speed-up by our method is less significant than that on the other two data sets. However, more than 10 times speed-up over RRWM on G^L can still be observed.

The space complexity of our PMA is also linear in the number of graph nodes. By using 16GB RAM, our method can handle about 819×2000 nodes at most.

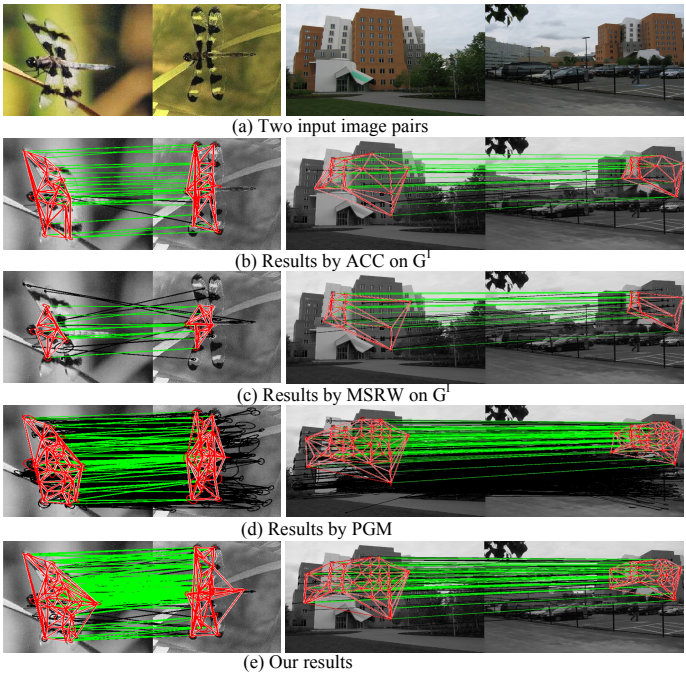


Fig. 10. Comparison on two image pairs of Intra-class dataset [1]

5 Conclusion

Feature matching is a long-standing and important problem for many applications in computer vision. This paper tried to address it by focusing on a novel issue: efficiently exploring the huge matching space based on the graph-based method. The crucial component of our proposed algorithm is to compute the graph density for one graph based on a reference graph. This enables a progressive mode-seeking framework for robustly exploring the huge matching spaces. To reduce the complexity of mode-seeking, we utilize the property that inliers often have larger graph density values than outliers and propose a simple density-aware sampling scheme. Results on several standard data sets demonstrated that our method significantly outperforms state-of-the-art methods, in terms of precision, recall and run time.

References

1. Cho, M., Lee, J., Lee, K.M.: Feature correspondence and deformable object matching via agglomerative correspondence clustering. In: ICCV (2009)
2. Cho, M., Lee, J., Lee, K.M.: Reweighted random walks for graph matching. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 492–505. Springer, Heidelberg (2010)

3. Cho, M., Lee, K.M.: Authority-shift clustering: Hierarchical clustering by authority seeking on graphs. In: CVPR (2010)
4. Cho, M., Lee, K.M.: Mode-seeking on graphs via random walks. In: CVPR (2012)
5. Cho, M., Lee, K.M.: Progressive graph matching: Making a move of graphs via probabilistic voting. In: CVPR (2012)
6. Cho, M., Shin, Y.M., Lee, K.M.: Co-recognition of image pairs by data-driven monte carlo image exploration. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 144–157. Springer, Heidelberg (2008)
7. Comaniciu, D., Meer, P.: A robust approach toward feature space analysis. TPAMI 24(5), 603–619 (2002)
8. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. IJPRAI, 265–298 (2004)
9. Duchenne, O., Bach, F., Kweon, I., Ponce, J.: A tensor-based algorithm for high-order graph matching. In: CVPR (2009)
10. Duchenne, O., Joulin, A., Ponce., J.: A graph-matching kernel for object categorization. In: ICCV (2011)
11. Ferrari, V., Tuytelaars, T., Gool, L.V.: Simultaneous object recognition and segmentation from single or multiple model views. IJCV 67(2), 159–188 (2006)
12. Freedman, D., Kisilev, P.: Fast mean shift by compact density representation. In: CVPR (2009)
13. Georgescu, B., Shimshoni, I., Meer, R.: Mean shift based clustering in high dimensions: A texture classification example. In: ICCV (2003)
14. Jouili, S., Tabbone, S., Lacroix, V.: Median graph shift: A new clustering algorithm for graph domain. In: ICPR (2010)
15. Lee, J., Cho, M., Lee, K.M.: Hyper-graph matching via reweighted random walks. In: CVPR (2011)
16. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: ICCV (2005)
17. Liu, H., Latecki, L.J., Yan, S.: Fast detection of dense subgraph with iterative shrinking and expansion. TPAMI (2013)
18. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV (1999)
19. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC (2002)
20. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. In: IJCV (2004)
21. Paris, S., Durand, F.: A topological approach to hierarchical segmentation using mean shift. In: CVPR (2007)
22. Torresani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 596–609. Springer, Heidelberg (2008)
23. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: CVPR (2008)
24. Zhang, W., Wang, X., Zhao, D., Tang, X.: Graph degree linkage: agglomerative clustering on a directed graph. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 428–441. Springer, Heidelberg (2012)