



Robust common visual pattern discovery using graph matching

Hongtao Xie ^{a,b}, Yongdong Zhang ^{a,*}, Ke Gao ^a, Sheng Tang ^a, Kefu Xu ^b, Li Guo ^b, Jintao Li ^a

^a Advanced Computing Research Laboratory, Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, China

^b Institute of Information Engineering, Chinese Academy of Sciences, National Engineering Laboratory for Information Security Technologies, China

ARTICLE INFO

Article history:

Received 1 December 2011

Accepted 21 April 2013

Available online 30 April 2013

Keywords:

Common visual pattern

Graph matching

Maximal clique

Quadratic optimization

Feature correspondence

Point set matching

Object recognition

Near-duplicate image retrieval

ABSTRACT

Discovering common visual patterns (CVPs) between two images is a difficult and time-consuming task, due to the photometric and geometric transformations. The state-of-the-art methods for CVPs discovery are either computationally expensive or have complicated constraints. In this paper, we formulate CVPs discovery as a graph matching problem, depending on pairwise geometric compatibility between feature correspondences. To efficiently find all CVPs, we propose a novel framework which consists of three components: Preliminary Initialization Optimization (PIO), Guided Expansion (GE) and Post Agglomerative Combination (PAC). PIO gets the initial CVPs and reduces the search space of CVPs discovery, based on the internal homogeneity of CVPs. Then, GE anchors on the initializations and gradually explores them, to find more and more correct correspondences. Finally, to reduce false and miss detection, PAC refines the discovery result in an agglomerative way. Experiments and applications conducted on benchmark datasets demonstrate the effectiveness and efficiency of our method.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

A common visual pattern (CVP) is the common part of two images, which has coherent spatial layout and similar visual content [1]. Discovering CVPs refers to establishing correct correspondences between two images. CVPs discovery is becoming increasingly important for various applications, such as object retrieval [1], image categorization and recognition [2,3], point set matching [4], near-duplicate image detection [5] and image database browsing [6].

Discovering CVPs is a difficult task, and there are several challenges. Firstly, significant photometric and geometric transformations usually take place between two images, such as occlusions, cropping, adding noise, changes of illumination, scale, viewpoint, and contrast, or even nonrigid deformations. Under the combinations of transformations and deformations, two instances of a CVP may differ not only in visual appearance, but also in 2D layout. Secondly, it lacks a priori knowledge of the CVPs, thus not known in advance the positions, shape, appearances, scales and the total number of CVPs. Finally, it is complex to detect a single CVP between two images, and thus finding all candidate CVPs will inevitably be computationally expensive. Fig. 1 shows such a challenging example with two CVPs between two images.

Recently, several CVPs discovery methods have been proposed [4–10,14–16]. However, existing approaches can only deal with weakly supervised cases with relatively slight occlusions and deformations [1,7], hence limiting the target image categories [4]. Besides, some of them rely on the initialization and do not guarantee the global optimal solution [5,6], or have complicated constraints and high computational cost [8–10,14].

In this paper, we formulate CVPs discovery as a graph matching problem by defining an objective function based on pairwise geometric compatibility between feature correspondences [11]. Firstly, we extract local features [12,13] in each image and get potential feature correspondences between two images. Then, we build a similarity graph, the nodes of which represent potential correspondences, and the edges represent the pairwise geometric consistency between corresponding correspondences. So, the spatially coherent feature correspondences i.e., a CVP constitute a dense subgraph [14], which is a weighted counterpart of maximal clique of un-weighted graph [17]. In this way, the CVPs discovery is formulated as a graph matching problem.

Finding the subgraphs (CVPs discovery) is NP-hard in general, but we solve this problem in an effective and efficient way, which contains three successive steps. Firstly, the Preliminary Initialization Optimization (PIO) algorithm is proposed to reduce the search space of CVPs discovery. PIO takes advantage of the internal homogeneity of CVPs, so it not only has better initializations of candidate CVPs, but also improves time performance. Secondly, we put forward a Guided Expansion (GE) method. GE anchors on the initializations of PIO, and gradually explores their neighboring correspondences, trying to construct more and more complete

* Corresponding author. Fax: +86 010 82546701.

E-mail addresses: xiehongtao@ite.ac.cn (H. Xie), zhyd@ict.ac.cn (Y. Zhang), kegao@ite.ac.cn (K. Gao), ts@ite.ac.cn (S. Tang), xukefu@ite.ac.cn (K. Xu), guoli@ite.ac.cn (L. Guo), jtli@ite.ac.cn (J. Li).



Fig. 1. Two CVPs detected between two images. All feature points (illustrated as ellipses) and candidate correspondences (yellow lines) are shown in the middle row. Despite deformation and high outlier ratio of candidate correspondences, our method can reliably locate all the CVPs and establish spatially coherent correspondences as demonstrated in the bottom row. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

CVPs. Finally, Post Agglomerative Combination (PAC) algorithm is presented to refine the final result and enhance discovery accuracy.

There are several advantages of our method. First, it works well under the circumstance of significant transformations. As shown in the bottom row of Fig. 1, our method can precisely locate the two CVPs, even when severe clutters and deformations take place. Second, it can discover all CVPs, no matter the manner of mappings between them. Third, it is effective and efficient. As illustrated in our experiments, it is faster than the previous methods [14,17,36], and has higher accuracy.

The remaining sections are organized as follows. Section 2 gives a review over previous works, while Section 3 describes the problem formulation of CVPs discovery. Our strategies are elaborated in Section 4, and the experiment results are presented in Section 5. Finally, Section 6 concludes this paper.

2. Literature review

CVPs discovery refers to establishing correct correspondences between two sets of feature points. There are many works on for-

mulating CVPs discovery as graph matching [19,20], and most methods cast it into optimization problems with different objective functions and constraints.

Shapiro and Brady [21] propose a method using spectral technique for point correspondence problem. Leordeanu and Hebert [7] present a spectral method working on a compatibility matrix where the diagonal elements represent one-to-one assignment costs, and other elements represent pairwise agreements between potential correspondences. The correspondences are then obtained by finding the principal eigenvector of this matrix. Afterwards, Cour et al. [22] put forward a spectral relaxation method for the graph matching problem that incorporates more mapping constraints, and present proper bistochastic normalization of the compatibility matrix to improve the overall matching performance. The intuition behind these approaches is that the spectrum of a graph is invariant under permutation and, hence, two isomorphic graphs should have the same spectrum. However, the spectral technique is sensitive to noises and outliers [14,37].

Graph matching has also been modeled as mathematical programming problems and several approaches have been introduced to solve it. Chui and Rangarajan [23] interpret graph matching as a mixed variable optimization problem. The correspondence problem is viewed as a linear assignment solved by deterministic annealing. Berg et al. [1] model the matching problem as a quadratic integer programming problem. Jiang et al. [24] and Li et al. [25] propose linear solution to the feature matching with different geometric constraints, which can be exactly or approximately linearized. Torresani and Kolmogorov [26] define a complicated objective function and then optimized by dual decomposition. These methods commonly require a strict mapping constraint assuming a weakly supervised model view, and do not deal with multiple pattern matches between images. Thus, they are not adequate for general matching of unsupervised images [37]. Besides, they have complicated objective functions of high computational cost [19].

Recently, hyper-graph matching methods are proposed, which incorporate higher-order similarity measures to achieve more accurate matching results [27–30]. Zass and Shashua [29] propose hyper-graph matching, which introduces a novel view that the matching problem and its corresponding solution are related by the Kronecker product. Duchenne et al. [30] put forward tensor matching, which can be interpreted as a higher-order extension of [7]. It uses high-order (mostly 3) constraints instead of unary or pairwise ones between correspondences, which results in a tensor representing affinity between feature tuples. The resulting energy function can then be optimized by the power iteration method. However, all these existing higher-order approaches are unable to effectively incorporate the matching constraints during their approximation stages, and using higher order scores is computationally expensive.

Besides, learning strategies are also applied to graph matching. In [20], an approach is introduced to learn the compatibility functions from examples and is found that linear assignment with such a learning scheme outperforms quadratic assignment solutions. In [31], Leordeanu et al. show for the first time to perform parameter learning in an unsupervised manner, in which no correct correspondences between graphs are given during training. Although these learning methods have better performance than the general approaches, their performances rely heavily on the training samples.

3. Problem formulation

In this section, we first elaborate how to formulate CVPs discovery as a graph matching problem. Then the current methods [14,17,36] for this problem are explained.

Given two sets of local features P and Q , obtained from two images respectively. Each feature point p is represented by $p = \{p_d, p_l\}$. p_d is the local descriptor and $p_l = (p_x, p_y)$, which is the coordinate of point p . Then, we can get the potential correspondences $C = \{c_i | c_i = (i, i'), i \in P \text{ and } i' \in Q\}$, between these two images. C can be obtained by comparing the similarity of local descriptors [12] or returned by bag-of-feature scheme [32,33]. For correspondence c_i , the transformation matrix H_i from i to i' can be calculated as [33]:

$$\begin{bmatrix} i'_x \\ i'_y \end{bmatrix} = s \times \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \times \begin{bmatrix} i_x \\ i_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (1)$$

In (1), there are four parameters: the scale factor s , the rotation factor θ and the translation components t_x and t_y . The parameters s and θ can be efficiently derived from the information of local features. They are calculated as:

$$s = 2^{(s_{i'} - s_i)} \quad \text{and} \quad \theta = \theta_{i'} - \theta_i, \quad (2)$$

where $s_{i'}$ and s_i are the characteristic scales of points i' and i , $\theta_{i'}$ and θ_i are the dominant orientations of these two points. These values are confirmed during the procedure of local feature detection.

Then for two correspondences $c_i = ((i, i'), H_i)$ and $c_j = ((j, j'), H_j)$, the pairwise geometric similarity $S_{pgc}(c_i, c_j)$ between c_i and c_j is defined by Xie et al. [34]:

$$T = \max_{c_i, c_j \in C} \left(\frac{1}{2} (|j'_l - H_j i_l| + |i'_l - H_i j_l|) \right), \quad (3)$$

$$S_{pgc}(c_i, c_j) = T - \frac{1}{2} (|j'_l - H_j i_l| + |i'_l - H_i j_l|),$$

where $|\cdot|$ denotes the Euclidean distance function. Obviously, $S_{pgc}(c_i, c_j)$ will be large if H_i and H_j are similar to each other.

Suppose the set C contains m correspondences, $C = \{c_1, c_2, \dots, c_m\}$ and we build a graph G with m vertices. Each vertex of G represents a correspondence in C and the weight of its edge between node i and node j is set to be $S_{pgc}(c_i, c_j)$. The weighted adjacency matrix of graph G is a $m \times m$ matrix, denoted by M , and defined as:

$$M(i, j) = \begin{cases} 0, & i = j \\ S_{pgc}(c_i, c_j), & i \neq j. \end{cases} \quad (4)$$

Obviously, M is symmetric and nonnegative.

For a CVP with n feature correspondences, it corresponds to a dense subgraph T of G with n vertices [14], which is a weighted counterpart of maximal clique [17]. In [17], Pavan establishes a connection between the weighted maximal cliques and the local maximizers of following quadratic function:

$$\begin{aligned} & \text{maximize } f(x) = x^T M x, \\ & \text{subject to } Ax = 1, \quad x \in \Delta \end{aligned} \quad (5)$$

where $\Delta = \{x \in R^m : x \geq 0 \text{ and } |x|_1 = 1\}$, and $Ax = 1$ reflects the one-to-one constraints: allowing one feature from P to match at most one feature from Q . It indicates that a subset V of vertices of G is the maximal clique if and only if its corresponding vector x is a local maximizer of $f(x)$, where $x_i > 0$ if $i \in V$ and $x_i = 0$ if $i \notin V$. x_i represents the probability of a dense subgraph containing node i . That is the probability of CVP V including correspondence c_i . The indices of all nonzero components of x constitute its support, denoted as $\sigma(x) = \{i | x_i > 0\}$.

As CVPs discovery refers to establishing correct correspondences between two images s , for correspondences c_i and c_j , if $i = j$ or $i' = j'$ then one of them will be deemed as false match and discarded. In other words, it has the one-to-one constraints. In our implementation, we relax the constraints on the solution, as proposed by Leordeanu and Hebert [7]. We first solve the quadratic

function without $Ax = 1$. When we get the candidate correspondences (candidate CVPs), we first confirm the confident correspondences and then remove the correspondences that are incompatible with the one-to-one constraints. By relaxing the constraints, it improves the efficiency of CVPs discovery [7].

As a CVP corresponds to a local maximizer of $f(x)$, to discover all CVPs, it has to find all local maximizers of $f(x)$, or local maximizers with relatively large values $f(x)$, which correspond to the true CVPs. To solve Eq. (5), [14,17,18] use replicator equation [35]:

$$x_i(t+1) = x_i(t) \frac{(Mx(t))_i}{x(t)^T M x(t)}, \quad i = 1, \dots, m. \quad (6)$$

That is, given an initialization vector $x(1)$, the corresponding local solution x^* of Eq. (5) can be obtained by iteratively executing (6). In [14], for each vertex v of G a set T is built, and T is composed of v and its neighboring vertices, which are the vertices that have large similarity with v . Then $x(1)$ is initialized in T , that is $x_i(1) = 1/|T|$ if $i \in T$ and $x_i(1) = 0$ if $i \notin T$. So in [14], the number of initialization vectors is equal to the number of vertices in G . With x^* , the corresponding CVP can be recovered easily, according to the values of x_i^* [14,17,36].

Afterwards, to robustly compute the CVPs, Liu et al. utilize the property of the standard quadratic optimization problem (Eq. (5)), and propose the graph shift algorithm [36]. It starts from each vertex of graph G and iteratively shifts towards the nearest dense subgraph of G along a certain trajectory. As it works in an EM-style (alternating shrinking and expansion), it is robust, especially when there exist large amount of noises.

4. Algorithm

With sufficient initialization vectors and Eq. (6), the solutions of quadratic function (5) can be obtained [14,17,36]. However, they have three major shortcomings:

- They have to initialize a vector $x(1)$ and iteratively executing Eq. (6) at each vertex v of G , which is computationally expensive. Thus at each vertex of a CVP, [14,36] initialize $x(1)$ once. As these initializations usually converge to the same local maximum, it is a great waste of time.
- In the expansion procedure of the graph shift algorithm [36], it does not take into account the spatial layout of the CVPs, and only uses the similarity values between the feature correspondences. As there are large number of outliers and false correspondences, some of the similarity values in M are polluted and not accurate enough [37]. So depending on similarity values alone for expansion may not only increase the execution time (wrong expansion will increase the number of iterations), but also reduce the discovery accuracy.
- Usually, quadratic function (5) has many local maximizers. Some of the maximizers represent CVPs and the others are sets of false correspondences. To reduce the influence caused by false correspondences, [14,17] just drop all the local maximizers with small function values $f(x)$. In fact, many small maxima correspond to parts of the true CVPs [37]. So dropping all small maxima directly will reduce the accuracy of CVPs discovery.

To solve the above three problems, we propose three algorithms: PIO, GE and PAC respectively. Firstly, PIO gets the initial CVPs with the internal homogeneity of them. Then, GE explores the initial CVPs in a guided way, to find more and more correct correspondences. Finally, PAC refines the discovery result, to reduce false and miss detection. With these three algorithms, we can discover CVPs effectively and efficiently.

4.1. Preliminary Initialization Optimization

Intuitively, for the correspondences in each CVP, as they have coherent spatial distribution, the pairwise geometric similarity values among them are high. For a false correspondence, as it is distributed randomly, the pairwise similarity values between it and the other correspondences are low. Based on this property, we can find the confident vertices of graph G . For each vertex i , we calculate its total similarity value with all the other vertices:

$$\nu(i) = \sum_{j \in G} M(i, j), \quad (7)$$

and get the average similarity value:

$$\nu_{ave} = \frac{1}{|G|} \sum_{i \in G} \nu(i). \quad (8)$$

Then, if $\nu(i)$ is larger than ν_{ave} , i is the confident vertex. In this way, most of the false correspondences can be separated out. However, as the amount of confident vertices is still huge, we cannot simply initialize $x(1)$ just on the basis of the confident vertices.

For a CVP, the correspondences in it have high pairwise geometric similarity, and the similarity values between the correspondences in it and those outside are low. As a cluster should have high internal homogeneity and there should be high inhomogeneity between the entities in a cluster and those outside [17]. So a CVP has the same property as a cluster [37]. Thus, to approximately find the CVPs, we can use a clustering method. As we already have the similarity matrix M , we use affinity propagation [38] to cluster the vertices of G and take the confident vertices as “preferences” [38]. For efficiency, the maximum number of iterations of affinity propagation is set to be 30–50. Then, for each cluster, a set T is made up of the vertices in this cluster and $x(1)$ is set in T . Compared to [14,36], for a graph G of 3000 vertices the number of $x(1)$ returned by PIO is about 300 times less. So it significantly reduces

the number of executing GE, which is complex, and improves the efficiency of CVPs discovery. What's more, as PIO initializes $x(1)$ in a meaningful way (clustering), it also improves CVPs discovery accuracy.

Fig. 2 shows the initial CVPs (indicated by different colors) detected by PIO and the feature correspondences. The correspondences in a CVP are painted with the same color. We can see that most of the initializations are located in the true CVPs, and the number of initializations is much lower than the number of candidate correspondences (the middle row of **Fig. 1**). So PIO can effectively reduce the search space of CVPs discovery. As the initializations are approximately located in the true CVPs, it can also improve the discovery accuracy, as illustrated in the experiments.

4.2. Guided Expansion

As PIO just simply initializes CVPs with clustering, it can only find the seeds of potential CVPs, and the initializations usually cover partial CVPs and contain noises. Thus, we have to exploit the initializations.

For the quadratic function (5), its solution x^* must satisfy the Karush–Kuhn–Tucker (KKT) conditions, i.e., the first-order necessary conditions for local optimality. That is, there exist $n+1$ real constraints (Lagrange multipliers) μ_1, \dots, μ_n and λ , with $\mu_i \geq 0$ for all $i = 1, \dots, n$, then:

$$(Mx^*)_i - \lambda + \mu_i = 0, \quad (9)$$

for all $i = 1, \dots, n$, and $\sum_{i=1}^n \mu_i = 0$.

Since both x_i^* and μ_i are nonnegative for all $i = 1, \dots, n$, Eq. (9) means that if $x_i^* > 0$, then $\mu_i = 0$. Hence, the KKT conditions can be rewritten as [17]:

$$(Mx^*)_i \begin{cases} = \lambda, & i \in \sigma(x^*) \\ \leq \lambda, & i \notin \sigma(x^*) \end{cases}. \quad (10)$$

Note that $x^T M x^* = \sum_i x_i^* (Mx^*)_i = \lambda$. The Eq. (10) means that if x^* is a maximizer (a CVP V) of function (5), then all the vertices in V ($i \in V$) satisfy $(Mx^*)_i = \lambda$ and the other vertices ($j \notin V$) satisfy $(Mx^*)_j \leq \lambda$.

For a subgraph Q of graph G , suppose one of its CVPs is x_Q^* . By adding zeros to the components whose index are in the set $G-Q$, we can expand x_Q^* to x_G^* , with $\sigma(x_Q^*) = \sigma(x_G^*)$ and $f(x_Q^*) = f(x_G^*)$. Then we get the following equation:

$$(Mx_G^*)_i \begin{cases} = \lambda_1, & i \in \sigma(x_G^*) \\ \leq \lambda_1, & i \in Q, i \notin \sigma(x_G^*) \\ (Mx_G^*)_i, & i \in G-Q \end{cases}, \quad (11)$$

where $\lambda_1 = f(x_Q^*) = f(x_G^*)$. If $(Mx_G^*)_i \leq \lambda_1$ for all $i \in G-Q$, as x_G^* satisfies Eq. (10), it also is a CVP of graph G . If $(Mx_G^*)_i > \lambda_1$ for some $i \in G-Q$, then x_G^* does not satisfy the KKT condition, thus it is not a CVP of G .

Thus, with Eq. (11) we can judge whether the CVP x_Q^* of a subgraph Q is also the CVP of graph G . If it is, no further computation is required. As subgraph Q has much less vertices than graph G , it is easy to deal with and the computational complexity is reduced. If not, we can initialize $x(1)$ in T_G , which includes $\sigma(x_Q^*)$ and the vertices that satisfy $(Mx_G^*)_i > \lambda_1$ for all $i \in G-Q$. Then the CVP of G is found by replicator equation. However, simply initializing $x(1)$ in T_G has one drawback. As there are large amount of outliers and false correspondences, parts of the similarity values in M are polluted and not accurate enough [37]. So some vertices that satisfy $(Mx_G^*)_i > \lambda_1$ for all $i \in G-Q$ may be false correspondences. So we have to refine T_G .

As a CVP of two images has coherent spatial layout, which remains stable under various disturbances, we can refine T_G with this property. Firstly, with the CVP x_Q^* of subgraph Q , we get the transformation matrix A that reflects the geometric change of the correspondences in x_Q^* . As the correspondences in x_Q^* have similar

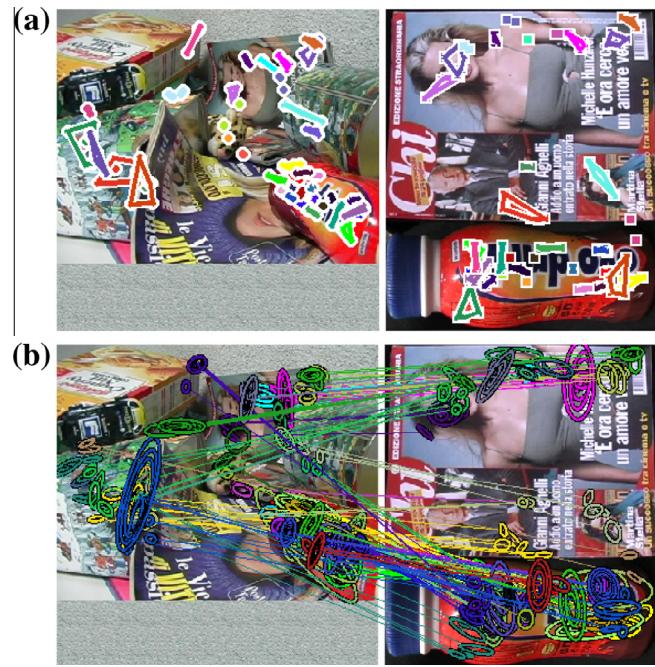


Fig. 2. (a) The initial CVPs (indicated by different colors) detected by PIO. (b) The feature correspondences in the initial CVPs and the correspondences in a CVP have the same color. These source images are same as in **Fig. 1**. We can see that most of the initializations are located in the true CVPs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

geometric change, matrix A can be calculated efficiently [12,33]. Then, the vertices (correspondences) that satisfy $(Mx_G^*)_i > \lambda_1$ for all $i \in G-Q$, will be removed from T_G if they are not consistent with A .

Algorithm 1. Guided Expansion

1. For each cluster (subgraph Q) returned by PIO, compute its CVP x_Q^* with replicator equation.
2. Repeat
3. Build a super graph G' , which includes $\sigma(x_Q^*)$ and the neighbor vertices of $\sigma(x_Q^*)$.
4. If $G' == Q$
5. return x_Q^*
6. End
7. Get $T_{G'}$ with Eq. (11) and refine it.
8. Initialize $x(1)$ in $T_{G'}$, and calculate the CVP $x_{G'}^*$ of G' with replicator equation.
9. If $x_{G'}^* == x_Q^*$
10. return $x_{G'}^*$
11. End
12. $x_Q^* = x_{G'}^*$ and $Q = G'$
13. Until $G' == G$.

Our Guided Expansion algorithm is explained in Algorithm 1. This algorithm works in an expectation maximization style. It first finds the CVP of a subgraph. Then, the expansion procedure expands current subgraph to its neighborhood. These two steps iterate until a final result is reached. As it operates on small subgraphs, and utilizes the properties of quadratic function and the spatial coherent of CVPs, it is efficient.

Algorithm 2. Post Agglomerative Combination

1. Set $S_0 = \{C_i\}, i = 1, \dots, n$ returned by GE as the initial candidate patterns;
2. $t = 0$;
3. (Hierarchical Pattern agglomerating)
4. Repeat
5. $t = t + 1$;
6. Among all possible pairs (C_i, C_j) in S_{t-1} , find (C_a, C_b) such that $S(C_a, C_b) = \max_{i,j} S(C_i, C_j)$;
7. If $S(C_a, C_b) \geq \sigma_D$
8. Merge C_a and C_b into a single pattern C_q ;
9. $S_t = (S_{t-1} - \{C_a, C_b\}) \cup \{C_q\}$;
10. End
11. Until $|S_t| == 1$ or $\max_{i,j} S(C_i, C_j) < \sigma_N$;
12. CVPs = $\{C_i | C_i \in S_t \text{ and } |C_i| > \sigma_N\}$

4.3. Post Agglomerative Combination

As the amount of initialization vectors obtained by PIO is still large (illustrated in Fig. 2), quadratic function (5) has many local maximizers. Thus, we propose Post Agglomerative Combination algorithm to refine the final result. For the candidate patterns C_i and C_j returned by GE, we define their similarity $S(C_i, C_j)$ as:

$$S(C_i, C_j) = \begin{cases} \frac{1}{|C_i||C_j|} \sum_{a \in C_i} \sum_{b \in C_j} M(a, b) + |C_i \cap C_j|, & i \neq j \\ 0, & i = j \end{cases}. \quad (12)$$

The PAC algorithm is summarized in Algorithm 2. At each agglomerating step, for two candidate patterns corresponding to a same CVP, as the similarity value between them is large, they

are likely to merge into a larger pattern. For patterns with false correspondences, as the similarity values between them and other patterns are small, they are likely to remain as smaller patterns. The agglomerating step is iterated until the maximum similarity is lower than a threshold σ_D . Typically, a CVP is likely to get enough numbers of correspondences. Thus, we can select reliable CVPs by choosing well-formed large patterns with a threshold σ_N . Additionally, other information such as location and geometric consistency of correspondences [33] can be applied to further refine the final result.

4.4. Complexity analysis

For a graph G with m vertices, it has $m(m - 1)/2$ edges. So the time complexity of PIO is $O(m^2)$. The major computation load is caused by the replicator equation, which evolves toward the CVP of current subgraph. Suppose the average number of iterations for the replicator equation is l , then the time complexity of the replicator dynamics procedure is $O(lm^2)$. If the number of initialize CVPs returned by PIO is k ($k \ll m$), the time complexity of GE is $O(klm^2)$, and the time complexity of PAC is $O(k^2)$. Thus the total time complexity of our method is $O(m^2)$. For the methods in [14,36], they have to execute replicator equation at each vertex of G , so the time complexity of them is $O(m^3)(O(mlm^2))$.

For our algorithms and the methods in [14,36], most of the memory space cost is caused by the storage of the weighted adjacency matrix M . So the space complexity of all these methods is $O(m^2)$.

These algorithms solve the CVPs discovery problem in three successive steps: initialization, expansion and post-processing. Together, they improve the efficiency and effectiveness of CVPs discovery, as illustrated in Section 5.

5. Experiments

In this section, we first evaluate our method on finding correspondences between 2D point sets and CVPs discovery in real images. Then we use object recognition and near-duplicate image retrieval as examples to show the applications of CVPs discovery and demonstrate the good performance of our method. In our experiments local features are obtained by the Hessian-Affine detector [39] and the SIFT descriptor [12].

The performance of our approach depends on two parameters: thresholds σ_D and σ_N . In the following, we will firstly study their impacts and select the optimal values.

5.1. Finding correspondences between 2D point sets

We first conduct an experiment on finding correspondences between point sets, and compare our method with graph shift in [36], iterative matching method (IPFP) in [16] and reweighted random walks for graph matching (RRWM) in [15], which are the state-of-art approaches to find correct correspondences. IPFP and RRWM are generally seen as among the most efficient methods in graph matching. For IPFP, it is initialized with the output of spectral matching [7].

We generate data set Q of 2D model points by randomly selecting n_Q^i inliers in a given region of the plane. Then we obtain the corresponding inliers in P by disturbing independently the n_Q^i points from Q with white Gaussian noise $N(0, \sigma)$, and rotating and translating the whole data set Q with a random rotation R and translation T . Next we add n_Q^o and n_P^o outliers in Q and P , respectively, by randomly selecting points in the same region as the inliers from Q and P , respectively, from the same random uniform distribution over the $x-y$ coordinates. The range of the $x-y$ point coordinates

in Q is $256\sqrt{n_Q/10}$ to enforce an approximately constant density of 10 points over a 256×256 region, as the number of points varies. The total number of points in Q and P are $n_Q = n_Q^i + n_Q^o$ and $n_P = n_P^i + n_P^o$. The parameter σ controls the level of deformation between two point sets, while n_Q^o and n_P^o control the numbers of outliers in Q and P , respectively. As explained in [7], 2D point sets match is a challenging problem. As the points are non-discriminative and translated and rotated arbitrarily, any of the n_P points from P can potentially match any of the n_Q model points from Q . In other words, there are $n_Q \times n_P$ vertices in graph G . So it maximizes the search space and leaves the task of finding a solution entirely to the relative geometric information between pairs of correspondence. Since the points do not have scale and orientation information, in this experiment, we only consider point sets of the same scale, and the rotation factor θ in Eq. (1) is set based on R .

5.1.1. Impact of parameters

To study the impact of the parameters σ_D and σ_N , we experiment with different data settings. The performance of findings correspondences between 2D point sets for different parameter values is shown in Fig. 3. We evaluate the performance by counting how many correspondences are consistent with the ground truths.

From Fig. 3, we can see that the performance of our method has a consistent behavior, while changing the values of parameters. Too large values of σ_D will allow more candidate patterns, which may include false correspondences, merge into a larger pattern. On the contrary, if σ_D is set to be a small value, many candidate patterns, which belong to a same CVP, will remain scattered. Similarly, too large and too small values of σ_N both reduce the performance of our method. The maximum score is reached for σ_D

ranging from 15 to 20, and σ_N ranging from 20 to 25, depending on the number of points and the degree of noises. So, in all ours following experiments these two parameters are set as: $\sigma_D = 15$ and $\sigma_N = 20$.

5.1.2. Robustness to deformations and outliers

Figs. 4 and 5 compare the performance of our method with graph shift in [36], RRWM in [15] and IPFP in [16]. All the algorithms run on the same data sets over 30 trials for each value of the varying parameter, and the mean performance curves are plotted. We score the performances of these methods by counting how many matches agree with the ground truths.

In the first row of Fig. 4, there are no outliers, and we vary the noise σ from 0.5 to 10 (in the step of 0.5). Obviously, our method is better than graph shift, RRWM and IPFP, especially when the deformation noise is severe. In the second row of Fig. 4, we keep the deformation parameter σ equal to 0 and the number of inliers fixed, while change the number of outliers. We can see that the curves of graph shift, RRWM and IPFP are unstable, so our method is more robust than the other methods in dealing with outliers. We can also observe that the performance of these algorithms improves as the number of inliers increases. It is because that as the number of inliers increases, each correct correspondence establishes pairwise consistency with more correct correspondences, thus the CVPs becomes more robust to deformations and outliers.

In Fig. 5, we first keep equal number of inliers and outliers, and then we set the noise σ to be 4 and change the number of outliers. Obviously, our method works better than graph shift, RRWM and IPFP. From these comparisons, we can see that the deformation noises have stronger influence on

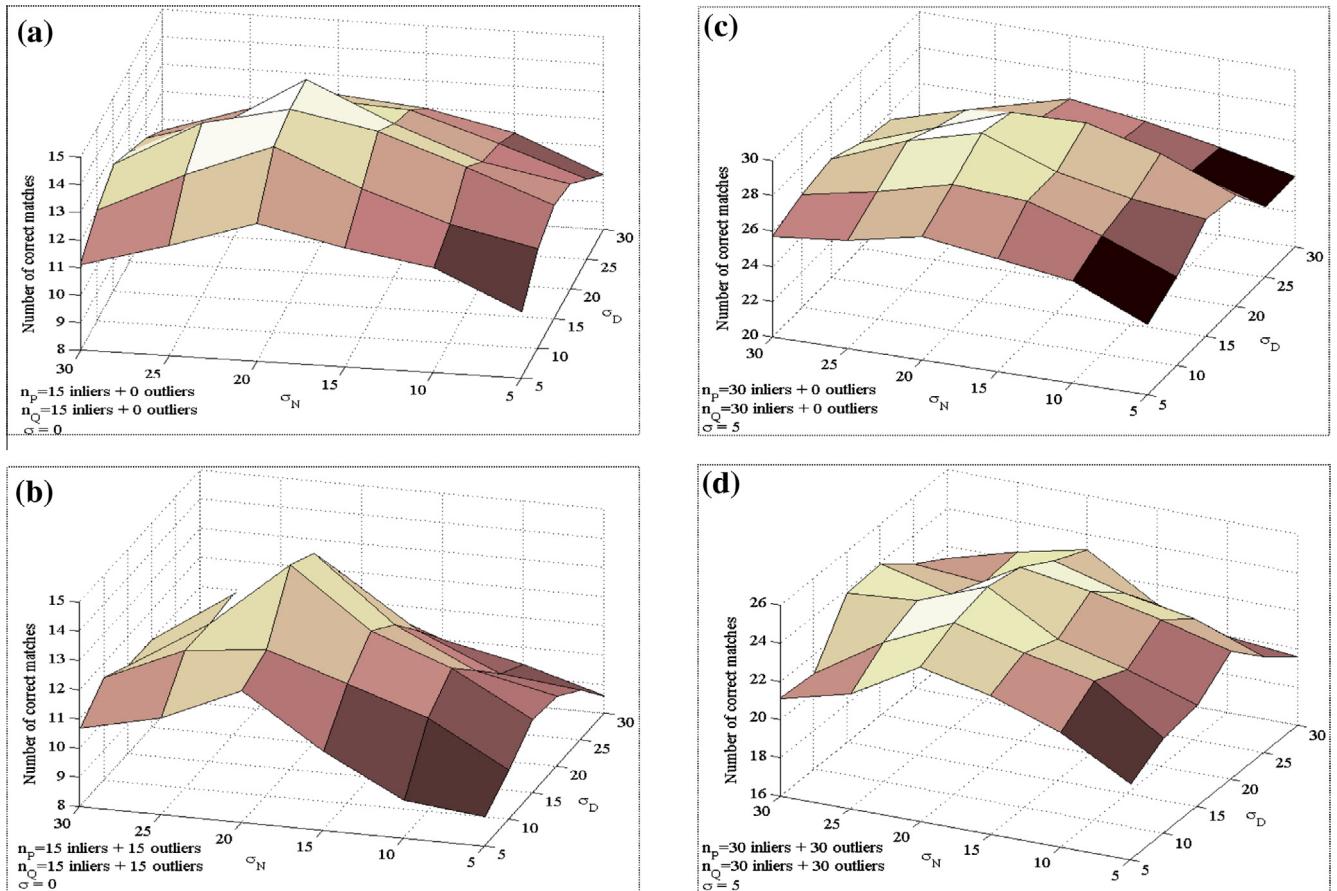


Fig. 3. Performance curves of our method with different parameter values, while varying the number of inliers and outliers, and the degree of deformation noises.

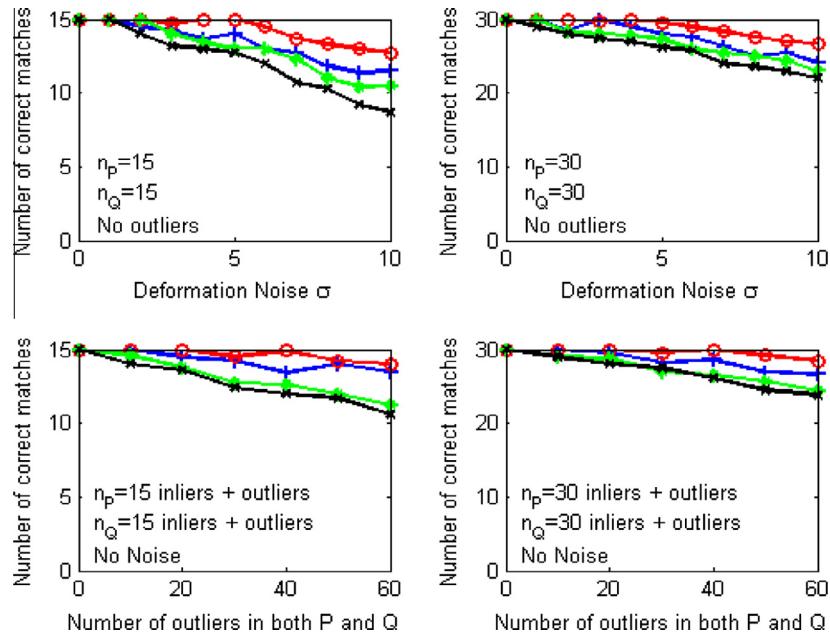


Fig. 4. Performance curves for our method vs. graph shift, RRWM and IPFP. The mean performance is shown as a red line (our method), a blue line (graph shift), a green line (RRWM) and a black line (IPFP). First row: no outliers, varying deformation noise. Second row: no deformation noise, varying the number of outliers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

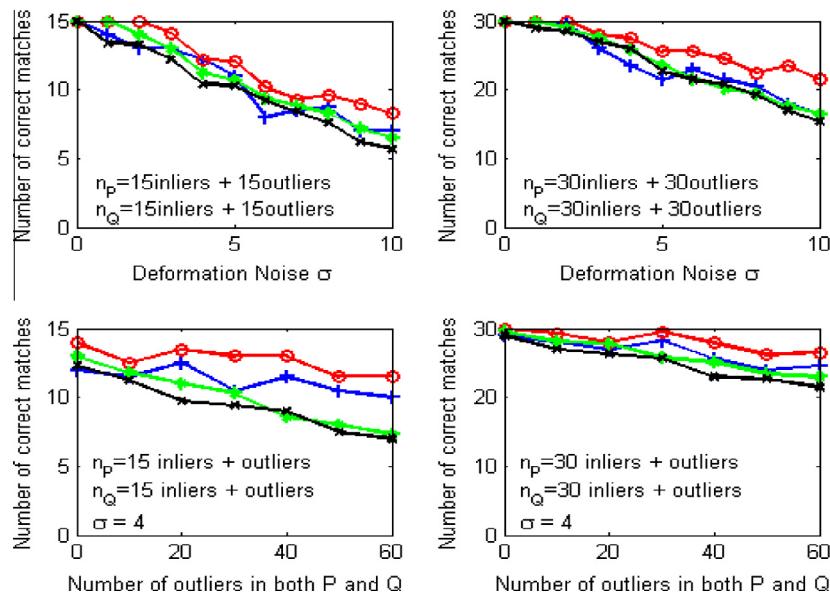


Fig. 5. The meaning of curves is same as in Fig. 4. First row: varying deformation noise with equal inliers and outliers. Second row: the number of outliers in each P and Q is varied with same deformation noise.

the CVPs discover accuracy than the outliers. This is because that the deformation noises change the positions of inliers. So with severe deformation noises, the correct point correspondences may be deemed as false matches. From these figures, we can see that our method consistently shows higher average performance.

5.1.3. Time performance analysis

Fig. 6 compares our method with graph shift [36], RRWM [15] and IPFP [16] in finding correct correspondences in graph G of different sizes. We can see that the time cost of these methods increases when the number of vertices increases. However, our method is much faster and the curve of our method rises less rapidly than that of graph shift and RRWM. The reason is that the time

complexity of our method is $O(m^2)$, while the time complexity of graph shift is $O(m^3)$ and RRWM has much iteration. So with the increasing of vertices, the time caused by graph shift and RRWM is becoming larger and larger than our method. Compared to graph shift, our method has introduced much additional computation in GE and PAC. Nevertheless, our method costs less time than graph shift. This is because that preliminary initialization of our method reduces the search space of CVPs discovery, which significantly improves the efficiency of our method.

When the number of vertices in G is about less than 2000, our method has comparable time performance with IPFP, and IPFP is faster than our method when this number is about larger than 2000. However in CVPs discovery, the number of correspondences

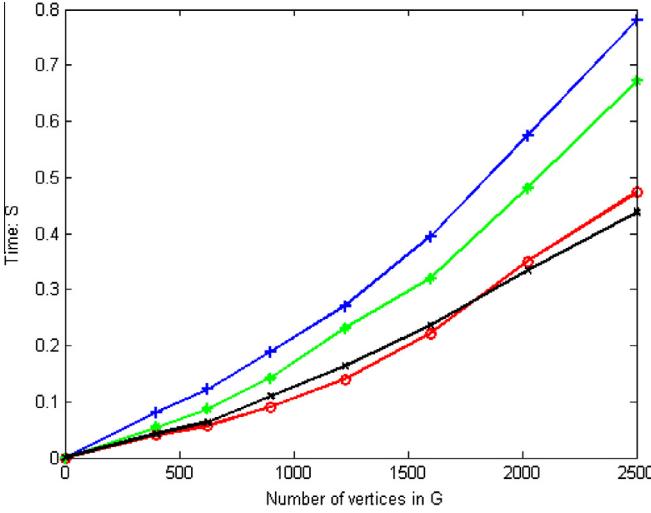


Fig. 6. The time cost of our method (red line), graph shift (blue line), RRWM (green line) and IPFP (black line) to find correspondences between point sets, when varying the number of vertices in G . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

between two images is generally less than 2000. Besides, the accuracy of IPFP is much lower than our method. So our method is more suitable for CVPs discovery than IPFP.

5.2. CVPs discovery in real images

To demonstrate the performance of our method for CVPs discovery in real images, we conduct experiment on IPDID dataset [40]. This set includes 10 image collections and each collection has 200 partial-similar images. There are 10 visual patterns in this set and one in each collection: American Flag, Beijing Olympic Badge, Disney Logo, Google Logo, iPhone, KFC Logo, Mona Lisa Smile, Rockets Logo, Starbucks Logo, and Exit Sign. All of these patterns are under varying viewpoints, rotation, scale, lighting and background changes. Note that each image has one to several visual patterns.

In the experiment, we first choose 10 images in each collection. Then we detect CVPs between the 45 ($10 \times 9/2$) pairs of images in each collection. For the metric of performance evaluation, we use F-measure [6]. F-measure is a popular measure as the weighted mean of precision and recall. For CVPs discovery, precision measures the accuracy of detection, while recall measures the ability of completely locating a common pattern. As the number of

patterns in each image is different, we redefine the measure of precision and recall. For the patterns $\{(C_k^i, C_k^j), k = 1, \dots, N\}$ detected between image pair i and j , suppose the corresponding manually labeled pattern for C_k^i is $L(C_k^i)$ and the number of patterns in image i and j is n_i and n_j , respectively, then precision, recall and F-measure are defined as:

$$\begin{aligned} \text{precision} &= \frac{\sum_{k=1}^N \frac{\text{area}(C_k^i \cap L(C_k^i))}{\text{area}(L(C_k^i))} \times \frac{\text{area}(C_k^j \cap L(C_k^i))}{\text{area}(L(C_k^j))}}{N}, \\ \text{recall} &= \frac{\sum_{k=1}^N \frac{\text{area}(C_k^i \cap L(C_k^i))}{\text{area}(L(C_k^i))} \times \frac{\text{area}(C_k^j \cap L(C_k^j))}{\text{area}(L(C_k^j))}}{n_i \times n_j}, \\ F - \text{measure} &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \end{aligned} \quad (13)$$

where $\text{area}(\bullet)$ is the area of the pattern. The F-measure for a collection is the average score of all the image pairs in this collection.

Fig. 7 compares our method with graph shift [36], RRWM [15] and IPFP [16] using F-measure under different image collections, leading to two major observations. First, our method has better performance than the others for all the image collections. Compared to graph shift, our method achieves 19.6% improvement in the average F-measure on IPDID set. Second, for some image collections, such as KFC Logo, Mona Lisa Smile and Starbucks Logo, all the methods have good performance, but for some other collections, such as American Flag and iPhone, the F-measure of all the methods is low. This is because that for American Flag, there are many repeated structures (five-pointed star and horizontal stripe). So there are many feature matches between two images and the similarity value between these correspondences is high. On the contrary, for iPhone, as it has black surface and the icons have uniform colors, we cannot detect enough local features in it, so the number of feature matches is rare. As our method is based on the similarity between feature correspondences, when the discriminability between correspondences is low, it cannot detect the CVPs with high accuracy.

The former experiments compare our method with the state-of-the-art in the effectiveness and efficiency of CVPs discovery. To pinpoint the potential of CVPs discovery, we further use object recognition and near-duplicate image retrieval as examples to show the applications of CVP and the good performance of our method.

5.3. Object recognition

In this experiment, we test our method on object recognition problem with the Toys dataset [41]. This set includes 40 model images (9 objects in total) and 23 test images. Each test image

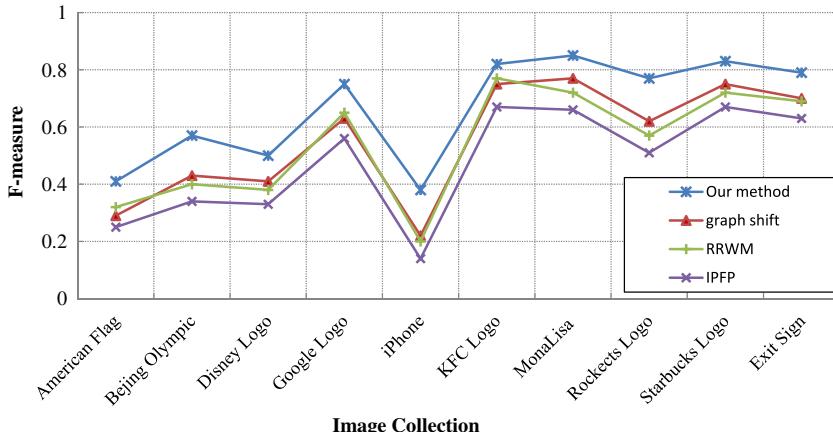


Fig. 7. F-measure performance of our method, graph shift, RRWM and IPFP for different image collections of IPDID dataset [40].

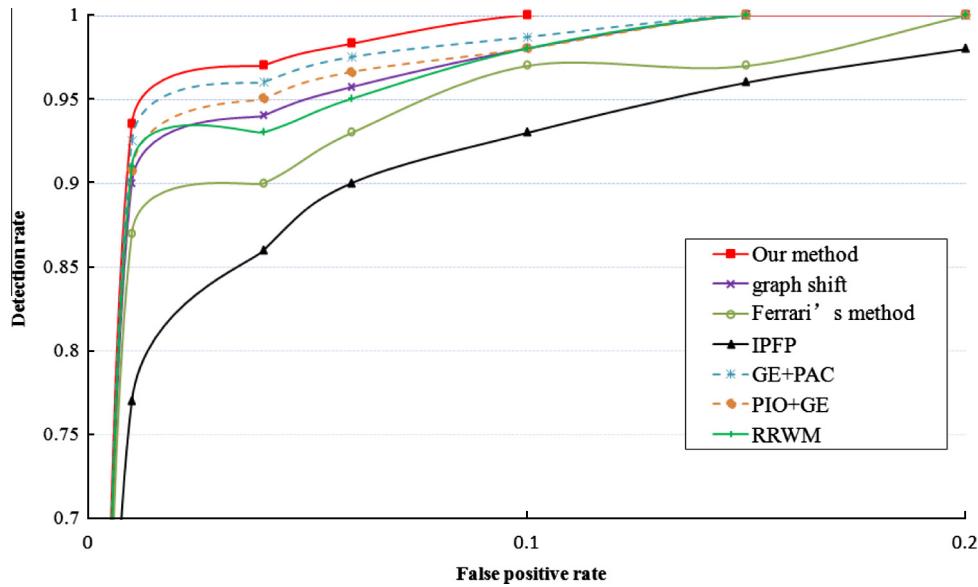


Fig. 8. ROC curves of object recognition for various methods on Toys dataset [41].

contains several objects, and these objects have different affine changes and nonrigid deformations. It is a popular dataset for object recognition evaluation.

The performance is quantified by processing all pairs of model and test images, and compared with graph shift [36], RRWM [15], IPFP [16] and Ferrari's method [41]. Besides, we also compare our method to two variants: GE + PAC and PIO + GE. That is the CVPs discovery without initialization and post-processing respectively. The initial feature correspondences between two images are obtained in the same manner as described in [12]. Our method has the same adjacency matrix M as [7,36].

The ROC curves in Fig. 8 depict the detection rate vs. false positive rate of different methods. As shown in Fig. 8, our method outperforms the methods of [15,16,36,41]. It achieves about 99% detection

rate with less than 1% false positive rate. Besides it also has excellent time performance. On average, our method only need 0.68 s to find the CVPs between a pair of 720×576 images (about 3600 correspondences), while graph shift needs 1.93 s. The IPFP runs faster than our method, but it has much lower detection accuracy.

From Fig. 8, we can also observe that GE + PAC and PIO + GE have better performance than graph shift, even without optimized initialization and post-processing respectively. This is because that graph shift does not have initialization and post-processing operations either. However, the recognition values of GE + PAC and PIO + GE are weaker than the best result. This indicates the importance of the procedures of PIO and PAC.

Examples of discovered CVPs are shown in Fig. 9, we can see that despite large deformations and viewpoint changes, our

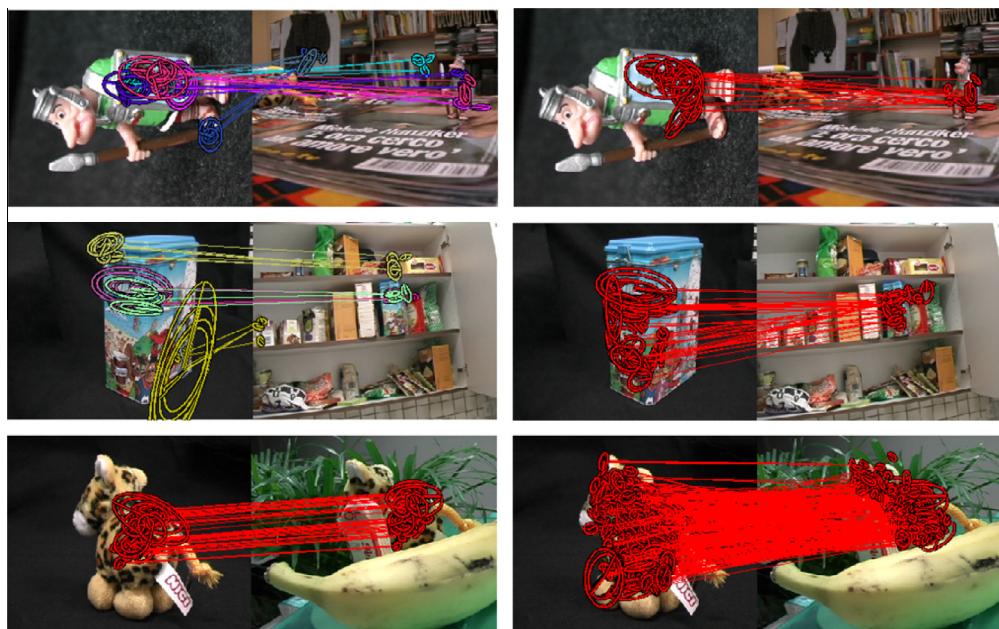


Fig. 9. CVPs discovery results on the image pairs of Toys dataset [41]. Left: the CVPs discovered by graph shift [36]. Right: the CVPs discovered by our method. All the detected CVPs are indicated by different colors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1

Search accuracy for four methods on Ukbench [42], Oxford [44], and Holiday [33] datasets.

	Ukbench	Oxford	Holiday
Baseline	0.842	0.607	0.781
Baseline + Ransac	0.864	0.661	0.813
Baseline + graph shift	0.898	0.684	0.834
Baseline + RRWM	0.906	0.715	0.816
Baseline + IPFP	0.864	0.678	0.787
Our method	0.937	0.718	0.882

method can correctly detect all the CVPs, while graph shift can only detect parts of the CVPs. The CVPs detected by our method contain enough numbers of correspondences and spread over sufficient areas.

5.4. Near-duplicate image retrieval

In this experiment we show an application of CVPs discovery in near-duplicate image retrieval, which plays an important role in many applications. For near-duplicate images, they have obvious CVPs. The experiment is conducted on Ukbench dataset [42], Holiday dataset [33] and Oxford dataset [44]. There are 10,200, 1491 and 5062 images in these sets, respectively. The images in these sets are modified through a serial of photometric and geometric transformations, such as blurring, cropping, adding noise, the changes of lighting, viewpoint, color and camera lens. They are the benchmark datasets for near-duplicate image retrieval evaluation.

The near-duplicate image retrieval system is built upon the scheme proposed by Jegou et al. [33]. For a query image, we first get the candidate similar images with the method in [33], then we re-rank the top 50 candidate images based on the sizes of detected CVPs. The performance metric applied in our experiment is mean average precision (mAP) [33]. We compare our method with the following approaches: (1) Baseline, the method in [33]; (2) Baseline + Ransac [43], the search result of Baseline is refined by Ransac; and (3) Baseline + graph shift, Baseline + RRWM and Baseline + IPFP, the search result of baseline is re-ranked based on the sizes of CVPs detected by graph shift, RRWM and IPFP respectively. For (2) and (3), we both re-rank the top 50 candidate images.

Table 1 compares the above six approaches, leading to two major observations:

- (1) Our method significantly improves the search precision. Compared to Baseline + graph shift, our method achieves 4.3%, 5% and 5.8% improvements in mAP on the three sets respectively. Compared to the baseline [33], the search accuracy gain obtained by our method is obvious. However, our method needs more time than the baseline. So our method is appropriate for the applications which have high demand on image retrieval accuracy, while have little requirement on efficiency, such as offline image database management and infringement detection.
- (2) Compared to Baseline + Ransac, our method has much better performance. This is because that Ransac is based on the global geometric changes between the query image and the

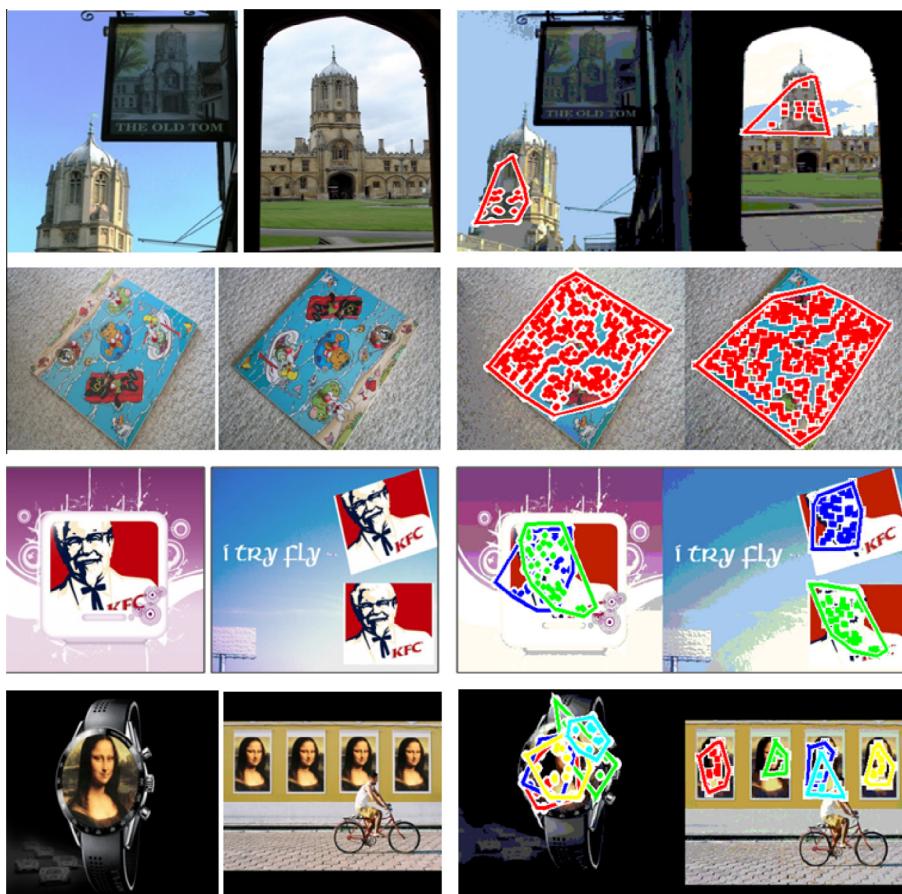


Fig. 10. CVPs discovery results on the near-duplicate image pairs. Left: the source near-duplicate image pairs. Right: the CVPs discovered by our method. There are multiple CVPs in the third and fourth rows, which are indicated by different colors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

candidate images, so it is sensitive to noises and outliers [33]. As there are many false matches between the query image and candidate images, it is not easy for Ransac to find the correct geometric transforms. However, as our method can deal with the effect caused by noises and outliers, it has higher retrieval accuracy.

Fig. 10 shows the CVPs discovery results on the near-duplicate image pairs. We can see that our method can steadily detect all the CVPs, even when there are multiple CVPs (one-to-two and one-to-four respectively) between two images. Although the images are diverse and contain large changes in scale, orientation, viewpoint and nonrigid deformations, the detected CVPs cover sufficient areas. All these experiments illustrate the effectiveness and efficiency of our method in CVPs discovery.

6. Conclusion

This paper has proposed an efficient CVPs discovery scheme, which contains three steps: PIO, GE and PAC. Firstly, PIO gets the initial CVPs with the internal homogeneity of them. Then, GE explores the initializations in a guided way, to find more and more correct correspondences. Finally, PAC refines the discovery result, to reduce false and miss detection. These three algorithms solve CVPs discovery through a successive and systematic way. Compared to the state-of-the-art methods, we can not only reduce the complexity of CVPs discovery, but also improve the discovery accuracy. Experiments show that our method can effectively and efficiently detect all CVPs automatically, despite variations caused by transformations and deformations. In future work, we will develop methods of using CVPs to provide query expansion and database browsing.

Acknowledgements

We would like to thank Dr. Hairong Liu (National University of Singapore) for providing us the source code of [36] for comparison and for giving us precious suggestions.

This work is supported by the National Nature Science Foundation of China (61271428, 61273247); National Key Technology Research and Development Program of China (2012BAH39B02); Co-building Program of Beijing Municipal Education Commission; "Strategic Priority Research Program" of the Chinese Academy of Sciences (Grant No. XDA06030200).

References

- [1] T. Berg, J. Malik, Shape matching and object recognition using low distortion correspondences, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 26–33.
- [2] Wei-Ta Chu, Cheng-Jung Li, Sheng-Chun Tseng, Travelmedia: an intelligent management system for media captured in travel, *Journal of Visual Communication and Image Representation* 22 (2011) 93–104.
- [3] K. Grauman, T. Darrell, Unsupervised learning of categories from sets of partially matching image features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 19–25.
- [4] J. Yuan, Y. Wu, Spatial random partition for common visual pattern discovery, in: Proceedings of the IEEE International Conference on Computer Vision, 2007, pp. 1–8.
- [5] Gangqiang Zhao, Junsong Yuan, Mining and cropping common objects from images, in: ACM Proceedings of the International Conference on Multimedia, 2010, pp. 975–978.
- [6] H.-K. Tan, C.-W. Ngo, Localized matching using earth mover's distance towards discovery of common patterns from small image samples, *Image and Vision Computing* 27 (10) (2009) 1470–1483.
- [7] M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in: IEEE International Conference on Computer Vision, 2005, pp. 1482–1489.
- [8] L. Torresani, V. Kolmogorov, C. Rother, Feature correspondence via graph matching: models and global optimization, in: European Conference on Computer Vision, 2008, pp. 596–609.
- [9] T. Tuytelaars, C.H. Lampert, M.B. Blaschko, W. Buntine, Unsupervised object discovery: a comparison, *International Journal of Computer Vision* 88 (2010) 284–302.
- [10] P. Hong, T.S. Huang, Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs, *JDAM* 139 (2004) 113–135.
- [11] Hongtao Xie, Ke Gao, Yongdong Zhang, Jintao Li, Huamin Ren, Common visual pattern discovery via graph matching, in: ACM Proceedings of the International Conference on Multimedia, 2011, pp. 1385–1388.
- [12] D. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* 2 (60) (2004) 91–110.
- [13] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (10) (2005) 1615–1630.
- [14] H. Liu, S. Yan, Common visual pattern discovery via spatially coherent correspondences, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1609–1616.
- [15] M. Cho, J. Lee, K.M. Lee, Reweighted random walks for graph matching, in: European Conference on Computer Vision, 2010, pp. 492–505.
- [16] M. Leordeanu, M. Hebert, R. Sukthankar, An integer projected fixed point method for graph matching and map inference, in: Neural Information Processing Systems, 2009.
- [17] M. Pavan, M. Pelillo, Dominant sets and pairwise clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (1) (2007) 167–172.
- [18] S. Bulo, M. Pelillo, A game-theoretic approach to hypergraph clustering, in: Advances in Neural Information Processing Systems, 2009.
- [19] Marwan Torki, Ahmed Elgammal, One-shot multi-set non-rigid feature-spatial matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3058–3065.
- [20] T.S. Caetano, Julian J. McAuley, Li Cheng, Quoc V. Le, Alex J. Smola, Learning graph matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (6) (2009) 1048–1058.
- [21] L. Shapiro, J. Brady, Feature-based correspondence: an eigenvector approach, *Image and Vision Computing* 10 (1992) 283–288.
- [22] T. Cour, P. Srinivasan, J. Shi, Balanced graph matching, in: Neural Information Processing Systems, 2006.
- [23] H. Chui, A. Rangarajan, A new point matching algorithm for non-rigid registration, *Computer Vision and Image Understanding* 89 (2003) 114–141.
- [24] H. Jiang, M.S. Drew, Z. Li, Matching by linear programming and successive convexification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (6) (2007) 959–975.
- [25] Hongsheng Li, Edward Kim, Xiaolei Huang, Lei He, Object matching with a locally affine-invariant constraint, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1641–1648.
- [26] L. Torresani, V. Kolmogorov, Feature correspondence via graph matching: models and global optimization, in: European Conference on Computer Vision, 2008, pp. 596–609.
- [27] Jungmin Lee, Minsu Cho, Kyoung Mu Lee, Hyper-graph matching via reweighted random walks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1633–1640.
- [28] M. Leordeanu, A. Zanfir, C. Sminchisescu, Semi-supervised learning and optimization for hypergraph matching, in: Proceedings of the IEEE International Conference on Computer Vision, 2011.
- [29] R. Zass, A. Shashua, Probabilistic graph and hyper graph matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
- [30] O. Duchenne, F. Bach, I. Kweon, J. Ponce, A tensor based algorithm for high-order graph matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1980–1987.
- [31] Marius Leordeanu, Martial Hebert, Unsupervised learning for graph matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2009, pp. 864–871.
- [32] J. Sivic, A. Zisserman, Efficient visual search of videos cast as text retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (4) (2009) 591–606.
- [33] H. Jegou, M. Douze, C. Schmid, Improving bag-of-features for large scale image search, *International Journal of Computer Vision* 87 (3) (2010) 316–336.
- [34] Hongtao Xie, Ke Gao, Yongdong Zhang, Jintao Li, Yizhi Liu, Pairwise weak geometric consistency for large scale image search, in: ACM International Conference on Multimedia Retrieval, 2011.
- [35] J. Weibull, *Evolutionary Game Theory*, MIT Press, 1997.
- [36] H. Liu, S. Yan, Robust graph mode seeking by graph shift, in: International Conference on Machine Learning, Haifa, Israel, 2010.
- [37] Minsu Cho, Jungmin Lee, Kyoung Mu Lee, Feature correspondence and deformable object matching via agglomerative correspondence clustering, in: IEEE International Conference on Computer Vision, 2009, pp. 1280–1287.
- [38] J. Frey, D. Dueck, Clustering by passing messages between data points, *Science* 315 (2007) 972–974.
- [39] K. Mikolajczyk, Binaries for affine covariant region descriptors, 2007. <<http://www.robots.ox.ac.uk/vgg/research/affine/>>
- [40] Zhipeng Wu, Q. Xu, S. Jiang, Q. Huang, P. Cui, L. Li, Adding affine invariant geometric constraint for partial-duplicate image retrieval, in: International Conference on Pattern Recognition, 2010, pp. 842–845.

- [41] V. Ferrari, T. Tuytelaars, L. Gool, Simultaneous object recognition and segmentation from single or multiple model views, *International Journal of Computer Vision* 67 (2) (2006) 159–188.
- [42] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2161–2168.
- [43] O. Chum, J. Matas, S. Obdrzalek, Enhancing RANSAC by generalized model optimization, in: *Proceedings of the Asian Conference on Computer Vision*, 2004, pp. 27–30.
- [44] J. Philbin, O. Chum, M. Isard, et al., Object retrieval with large vocabularies and fast spatial matching, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.