

Common Visual Pattern Discovery via Spatially Coherent Correspondences

Hairong Liu, Shuicheng Yan

Department of Electrical and Computer Engineering, National University of Singapore, Singapore

e1eliuh@nus.edu.sg, eleyans@nus.edu.sg

Abstract

We investigate how to discover all common visual patterns within two sets of feature points. Common visual patterns generally share similar local features as well as similar spatial layout. In this paper these two types of information are integrated and encoded into the edges of a graph whose nodes represent potential correspondences, and the common visual patterns then correspond to those strongly connected subgraphs. All such strongly connected subgraphs correspond to large local maxima of a quadratic function on simplex, which is an approximate measure of the average intra-cluster affinity score of these subgraphs. We find all large local maxima of this function, thus discover all common visual patterns and recover the correct correspondences, using replicator equation and through a systematic way of initialization. The proposed algorithm possesses two characteristics: 1) robust to outliers, and 2) being able to discover all common visual patterns, no matter the mappings among the common visual patterns are one to one, one to many, or many to many. Extensive experiments on both point sets and real images demonstrate the properties of our proposed algorithm in terms of robustness to outliers, tolerance to large spatial deformations, and simplicity in implementation.

1. Introduction

Similar objects usually share some common visual patterns, and these common visual patterns are often very distinctive. Moreover, a common visual pattern generally has certain spatial layout, which is less likely produced by noises or outliers. Thus, detection of common visual patterns is valuable for many vision tasks [19], such as object recognition [1], point set matching [10, 3] and 2D and 3D registration [2, 6].

In this paper, we consider feature points, that is, each point is associated with certain coordinates and possible local features. The local features can be any distinctive features, such as SIFT features [9], and can also be none, that is, the feature points only have coordinates. We also distinguish two types of correspondences. The first type is the



Figure 1. Three common visual patterns detected between two images. All candidate feature point matches after pruning are shown in the middle row, our method can reliably locate all three common visual patterns at different scales and establish spatially coherent correspondences between feature points as demonstrated in the bottom row. For better viewing, please see the original color pdf file.

correspondence between visual patterns, which can be one to one, one to many, or many to many. The second type is the correspondence between feature points, which is usually one to one within a certain pair of common visual patterns. For two images, a common visual pattern means a cluster of spatially coherent correspondences between two sets of feature points within two images respectively.

Detecting common visual patterns is a challenging task, even for humans. There are many factors to be considered: 1) two instances of a common visual pattern may be at different scales and orientations; 2) there may be large deformation cross images; 3) there is usually significant occlu-

sions and outliers. Figure 1 shows such a challenging example with three pairs of common visual patterns within two images.

Recently, many robust feature points correspondence pursuing methods have been proposed [1, 8, 16]. However, as [5] pointed out, these methods all deal with weakly supervised cases with relatively low outlier ratio. Moreover, these methods only establish correspondences between two instances of one common pattern, while there may exist multiple common patterns in many real world scenarios.

Our method proposed in this work is motivated by the following insights:

1. There exists large amount of redundancy in pairwise distances among feature points. For example, for a planar common visual pattern with n feature points, there are $n(n-1)/2$ pairwise distances, but at most $3n-6$ pairwise distances are free because of rigidity. Such redundancy means that pairwise distances are enough to fix the spatial layout of the pattern, and also allow relatively large deformation on some individual pairwise distances while keeping the whole layout similar.
2. In a graph whose nodes represent potential correspondences between feature points, if encoding the feature similarity and geometric consistency information into edges, the spatially coherent feature point correspondences constitutes a dense subgraph, which is a weighted counterpart of maximal cliques of unweighted graph. Finding such dense subgraphs is NP-hard in general, but can be efficiently solved in our settings as discussed later.

On the basis of above insights, we propose a novel algorithm which can detect all common visual patterns and establish spatially coherent correspondences between two sets of feature points. As shown in the bottom row of Figure 1, our method can correctly locate the three common visual patterns. The proposed method has several advantages over previous methods. First, it works well in very cluttered situations. For example, for two point sets, it can effectively detect common patterns even when the number of outliers is ten times of the number of inliers as illustrated in the experiment section. Second, it can detect all common visual patterns and automatically determine whether the mappings between common visual patterns are one to one, one to many, or many to many. Third, it is simple, easy to implement and efficient.

2. Literature Review

Detecting common visual patterns refers to establishing correct correspondences between two sets of feature points, and most methods cast feature point correspondence into an optimization problem, with different objective functions and constraints. Shapiro and Brady [15] proposed a spectral technique for point correspondence problem, which was further improved by Carcassoni and Han-

cock [4]. Leordeanu [8] proposed a method using spectral technique by optimizing a quadratic objective function. A general issue for spectral technique is its sensitivity to noises and outliers.

Maciel and Costeria [10] proposed a method based on integer quadratic programming, which can be reduced to an equivalent concave minimization problem. Berg and Malik [1] proposed a more efficient implementation, which however only works for the cases of allowing several feature points from one image to match the same feature point from the second image. Torresani et al. [16] defined a complicated objective function and then optimized by dual decomposition. These methods either have complicated objective function or have complicated constraints, which thus results in high computational cost.

Caetano et al. [3] proposed to establish correspondences between point sets based on Markov random field. However, as it requires that every point in the model must correspond to one point in data, its applicability is limited. Cho [5] proposed a method based on hierarchical agglomerative clustering, which is based on heuristic rules and thus global optimum cannot be guaranteed.

Our method optimizes the same objective function proposed in spectral method [8], but with different constraints. From the prospective of graph matching, our method is very similar to the methods in [7, 14], which translate the point correspondence problem into maximal clique problem, and also similar to the method in [6], which translates the point correspondence problem into a vertex cover problem. Both maximal clique and vertex cover problem are NP-hard. Our method investigates the properties of the solution to this specific problem and can obtain all large maxima (including global maximum) efficiently.

3. Problem Formulation

Given two sets of feature points obtained from two images, P and Q , with n_P and n_Q feature points, respectively. Each feature point p consists of two parts, $p = \{p_d, p_c\}$. p_d represents local features, such as the SIFT features [9], and can also be none, which reduces to point set; $p_c = \{x, y, z\}$, which are the coordinates of the feature point. Note that this algorithm works for both 2D and 3D feature points; for 2D feature points, we simply set $z = 0$.

The product space $C = P \times Q$ contains all possible correspondences, and each correspondence c_i is a pair (i, i') , where $i \in P$ and $i' \in Q$. For a correspondence $c_i = (i, i')$, its *feature similarity score* is a nonnegative function of these two features, denoted by $S_{c_i} = f_1(i_d, i'_d)$. Since the correct correspondences generally have similar features, we may only consider a much smaller set $M = \{c | c \in C, S_c > \varepsilon\}$, where ε is a manually set threshold.

For two correspondences $c_i = (i, i')$ and $c_j = (j, j')$, suppose the distance between i and j in the first image, and the distance between i' and j' in the second image, are l_{ij}

and $l_{i'j'}$, respectively. Obviously, if these two pairs are correct correspondences of a common visual pattern, we may scale the second image by a factor of $l_{ij}/l_{i'j'}$ to align them, and we say that these two pairs vote for the scale factor $l_{ij}/l_{i'j'}$. For a common visual pattern with n features, there are n correct correspondences, and thus the number of such pairs is $n(n-1)/2$, they should all vote for nearly the same scale. Such coherence is less possibly produced by noises or outliers, thus provides a strong spatial cue for common visual pattern detection.

At a scale factor s , the *geometric consistency score* of two correspondences $c_i = (i, i')$ and $c_j = (j, j')$ is expressed in the following form:

$$S_{c_i c_j}(s) = f_2(|l_{ij} - sl_{i'j'}|), \quad (1)$$

where $f_2(x)$ is a nonnegative monotonic decreasing function.

Suppose the set M contains m correspondences, $M = \{c_1, c_2, \dots, c_m\}$ and we build a graph G with m vertices, each vertex of which represents a correspondence in M . This graph is called *dynamic correspondence graph*, since the weight of its edge between node i and node j is defined as follows:

$$w_{ij}(s) = S_{c_i} S_{c_j} S_{c_i c_j}(s), \quad (2)$$

which is a function with respect to the scale factor s .

The weighted adjacency matrix of the dynamic correspondence graph G is an m -by- m matrix, denoted by $A(s)$, and defined as:

$$A(i, j)(s) = \begin{cases} 0, & i = j; \\ w_{ij}(s), & i \neq j. \end{cases} \quad (3)$$

Obviously, $A(s)$ is symmetric and nonnegative.

For a common visual pattern with n feature points, when at the correct scale s_0 , it corresponds to a dense subgraph T of G with n vertices, which is a weighted counterpart of maximal clique. Such a dense subgraph has high *average intra-cluster affinity score* $S_{av}(s_0) = \frac{1}{n^2} \sum_{i \in T, j \in T} A(i, j)(s_0)$. If we represent T by an indicator vector y , such that $y(i) = 1$ if $i \in T$ and zero otherwise, then the average intra-cluster affinity score can be rewritten in a quadratic form, $S_{av}(s_0) = \frac{1}{n^2} \sum_{i \in T, j \in T} A(i, j)(s_0) = \frac{1}{n^2} y^T A(s_0) y = x^T A(s_0) x$, where $x = y/n$. Since $\sum_i y(i) = n$, then $\sum_i x(i) = 1$, which is a constraint over x . In fact, the Motzkin-Straus theorem [12] has established a connection between the maximal cliques and the local maximizers of this quadratic function:

$$\begin{aligned} & \text{maximize} && f(x) = x^T A(s_0) x \\ & \text{subject to} && x \in \Delta \end{aligned} \quad (4)$$

where $\Delta = \{x \in \mathbb{R}^m : x \geq 0 \text{ and } |x|_1 = 1\}$ is the standard simplex of \mathbb{R}^m . Roughly speaking, it states that a subset V of vertices of graph G is a maximal clique if and only if its characteristic vector x^V is a local maximizer of the quadratic function f in Δ , where $x_i^V = 1/|V|$ if $i \in V$,

$x_i^V = 0$ otherwise. Obviously, $f(x)$ is an approximation of $S_{av}(s_0)$ by relaxing x .

Although original Motzkin-Straus theorem only relates to un-weighted graph, Pavan and Pelillo [13] generalized it to weighed graph and established the connection between dominant sets (dense subgraphs) and the local maximizers of (4).

The adoption of l_1 -norm in (4) has several advantages. First, it has an intuitive probabilistic meaning, and x_i represents the probability of a dense subgraph containing node i . Second, the solution x^* is sparse, and only some components in the same dense subgraph have large values, and other components have relatively smaller values, which means that it can detect clusters, leaving the remaining points not belonging to any clusters, such as outliers, ungrouped. On the contrary, many other algorithms, such as spectral method in [8], are not expected to work well when noises and outliers are significant, due to their insisting on partitioning all the input data into clusters.

The main ideas of our proposed method is illustrated in Figure 2, common visual patterns correspond to the dense subgraphs of G and also the dense blocks of weighted adjacency matrix A . Since each dense subgraph of G corresponds to a local maximizer of function $f(x)$, to detect all common visual patterns, we need to find all local maximizers of $f(x)$, or at least all local maximizers with relatively large function values $f(x)$, which correspond to the true common visual patterns. This is impractical in general; however, because of the characteristics of local maximizers in this specific problem, they can be efficiently obtained through a systematic way of initialization.

4. Algorithm

At a scale s_0 , the optimization problem $f(x) = x^T A(s_0) x, x \in \Delta$ may have many local maxima. Each large local maximum corresponds to a true common visual pattern, while small local maxima usually result from noises and outliers.

Given an initialization $x(1)$, the corresponding local solution x^* of (4) can be efficiently obtained by *replicator equation*, which arises in evolutionary game theory [17]. The discrete-time version of first-order replicator equation has the following form:

$$x_i(t+1) = x_i(t) \frac{(A(s_0)x(t))_i}{x(t)^T A(s_0)x(t)}, \quad i = 1, \dots, n. \quad (5)$$

It can be observed that the simplex Δ is invariant under these dynamics, which means that every trajectory starting in Δ will remain in Δ . Moreover, it has been proven in [17] that, when $A(s_0)$ is symmetric and with nonnegative entries, the objective function $f(x) = x^T A(s_0) x$ is strictly increasing along any nonconstant trajectory of (5), and its asymptotically stable points are in one-to-one correspondence to strict local solutions of (4).

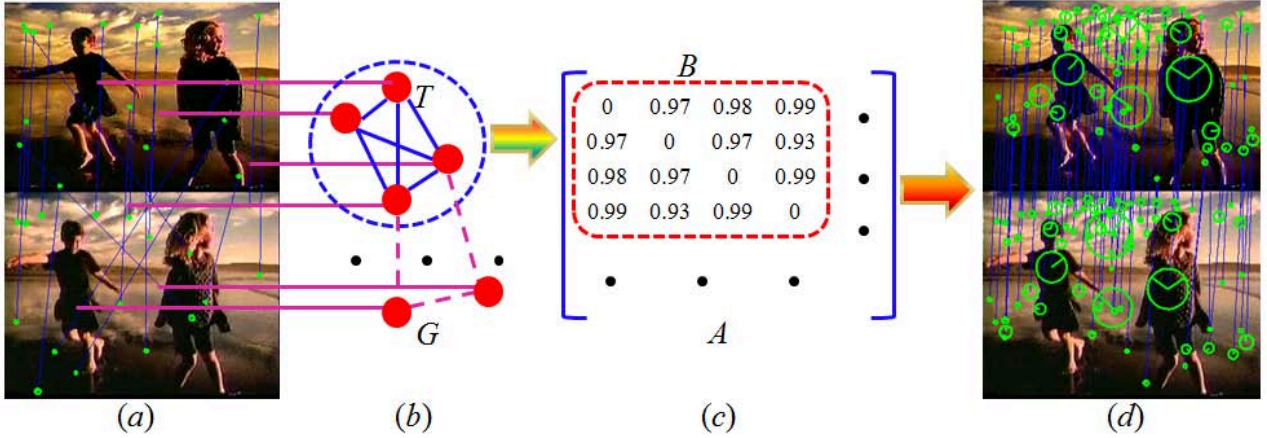


Figure 2. Illustration of the main ideas in our method. Find all candidate correspondences shown in (a) by local features (for clarity, only a small subset of the candidate correspondences are shown), and then form the graph G in (b) and weighed adjacency matrix A in (c). The common visual pattern corresponds to the dense subgraph T of G , and also the dense block B of A after some permutations. For better viewing, please see the original color pdf file.

To find all large local maximizers $\{x^*\}$, an intuitive way is to give many initializations, and guarantee that at the attractive basin of each local maximum, there is at least one initialization. Because in our context, a local maximizer x^* corresponds to a common visual pattern, thus it possesses two properties:

1. *Local Property.* For each vertex v of graph G , the common visual pattern containing v is a subset of $N(v) \cup v$, where $N(v)$ is the neighborhood of v . Thus, we only need to initialize $x(1)$ in the neighborhood of every vertex of the dynamic correspondence graph G .
2. *Non-Intersection Property.* Different common visual patterns generally do not contain common vertices in G , which means that if two local maximizers, x^* and y^* , correspond to different common visual patterns, then $x^{*T}y^* \approx 0$.

In [13], Pavan and Pelillo proposed to detect dominant sets iteratively, that is, when finding a dominant set, they remove the vertices in this set, and then reiterate on the remaining vertices. This method utilizes non-intersection property; however, according to our experiments, it usually cannot find all large maxima, together with global maxima. Local property is very important, and it not only tells us how to initialize $x(1)$, but also reduce memory and computational cost. Since $A(s_0)$ is usually very sparse, we only need to operate on the neighborhoods of vertices.

When $A(s_0)$ is fixed, the algorithm to find all large local maxima is described in Algorithm 1, which utilizes both local property and non-intersection property. Since at each vertex of a common visual pattern, we initialize $x(1)$ once, and these initializations usually converge to the same local maxima, we need to merge them. Note that such redundancy provides additional robustness to noise and outliers. After sorting, we drop all small local maxima, since they usually come from noises and outliers.

Algorithm 1: Find all large local maxima by replicator equation

Input: Weighted adjacency matrix $A(s_0)$

1. **for** each vertex v of the graph G **do**
 - (a) **Build** set $T = N(v) \cup v$ **and initialize** $x(1)$ **in** T , that is, $x_i(1) = 1/|T|$, if $i \in T$, otherwise set $x_i(1) = 0$;
 - (b) Obtain the corresponding local maximizer x^* by the replicator equation (5);
- end**
2. Sort all local maximizers $\{x^*\}$ according to $f(x^*)$, in descending order, and drop local maximizers with small $f(x^*)$;
3. Merge two local maximizers x^* and y^* by average if $x^{*T}y^* > \eta$, where η is a manually set threshold;

Output: All maximizers corresponding to large local maxima.

For a local maximizer x^* , we need to recover the corresponding common visual pattern. Since x_i^* represents the probability of this common pattern to contain the vertex i , we can recover the corresponding common visual pattern as described in Algorithm 2, which is similar as the spectral method in [8]. Since x^* is sparse, the number of its sufficiently large components usually indicates how large the common visual pattern is. In Algorithm 2, the test in *while* loop guarantees the proximity of this dense subgraph.

In real applications, the scale factor is usually limited within a range, say $R = [s_0, s_1]$, thus we can uniformly sample some scales in this range to detect different common visual patterns. Based on the Algorithm 1 and Algorithm 2, the overall algorithm is summarized in Algorithm 3.

Equation (5) has a nice property: if $x_i(t) = 0$, then $x_i(t+1) = 0$ and $x_i(t)$ does not affect the computation

Algorithm 2: Recover common visual pattern from local maximizer x^*

Input: Local maximizer x^* ;

1. Sort the components of x^* in descending order and initialize a set L to be empty;
2. **while** *true* **do**
 - (a) Select the largest component x_i^* ;
 - (b) Test whether all $A(i, j) > \vartheta, j \in L$, where ϑ is a manually set threshold. If the test is true, then add i to set L and set $x_i^* = 0$; otherwise, break;

end

Output: Common visual pattern L .

Algorithm 3: Detect all common visual patterns at different scale factors

Input: Two sets of feature points, P and Q , and the range of scale factor, $R = [s_0, s_1]$;

1. Sample some scale factors in $R = [s_0, s_1]$;
2. **for** *each sampling scale factor* s **in** R **do**
 - (a) According to (2), build the weighted adjacency matrix $A(s)$;
 - (b) Detect all large local maxima by Algorithm 1;
 - (c) Recover the corresponding common visual patterns by Algorithm 2;

end

Output: All common visual patterns, together with the point correspondences, at different scale factors.

of $x_j(t), j \neq i$, which means that we can ignore $x_i(t)$. This property together with the local property of x^* can greatly reduce the computation and memory cost.

The time complexity of computing feature similarity score for all possible correspondences is $O(n_P n_Q)$, and $O(lm^2)$ for computing $A(s)$ at l scales. At a certain scale s , $A(s)$ is usually very sparse, thus the neighborhood size of the vertices is much smaller than m . If we suppose the average neighborhood size is d , since the computation of (5) is within neighborhood, the computation complexity is $O(kmd^2)$, where k is the average number of iterations of the equation (5), and the memory requirement is $O(d^2)$.

5. Experiments

We evaluate our proposed algorithm on four tasks: finding correspondences between 2D sets of points, multiple common patterns detection, salient points based face matching, and near-duplicate image retrieval. In all our experiments, $\eta = 0.001$, $\varepsilon = 0.1$ and $\vartheta = 0.5$.

5.1. Finding Correspondences between Point Sets

For fair comparison with spectral method in [8], we conduct the same experiments as described in [8]: first generate data set Q of 2D model points by randomly selecting n_Q^i inliers in a given region of the plane, then obtain the

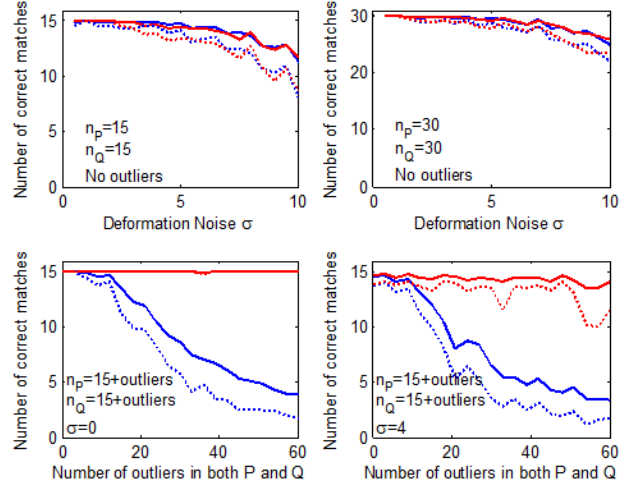


Figure 3. Performance curves for our method vs. the spectral method in [8]. The mean performance is shown as a solid red line (our method) and a solid blue line (method in [8]). One *std* below the mean is shown as red dotted lines for our method and blue dotted lines for the spectral method in [8]. First row: no outliers, varying deformation noise, and the number of correct matches is plotted. Second row: the number of outliers in each P and Q is varied with (right) and without (left) deformation.

corresponding inliers in P by disturbing independently the n_Q^i points from Q with white gaussian noise $N(0, \sigma)$, and then rotate and translate the whole data set Q with a random rotation and translation. 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_P^o and n_Q^o control the numbers of outliers in P and Q , respectively. As pointed out in [8], it is a challenging problem because of the homogeneity of data and the large search space.

Since the points themselves are not distinctive, we set all feature similarity scores to be 1, thus $M = C$. We rely fully on the geometric consistency information to find the correspondences. As in [8], in this experiment, we only consider point sets of the same scale, and the geometric consistency score is defined as follows:

$$S(c_i, c_j) = \begin{cases} 4.5 - \frac{(l_{ij} - l_{i'j'})^2}{2\sigma_d^2} & \text{if } |l_{ij} - l_{i'j'}| < 3\sigma_d; \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The parameter σ_d controls the sensitivity of the score on deformations. The larger the σ_d is, the more deformations we can accommodate, but also the more pairwise relationships between wrong assignments will get a positive score.

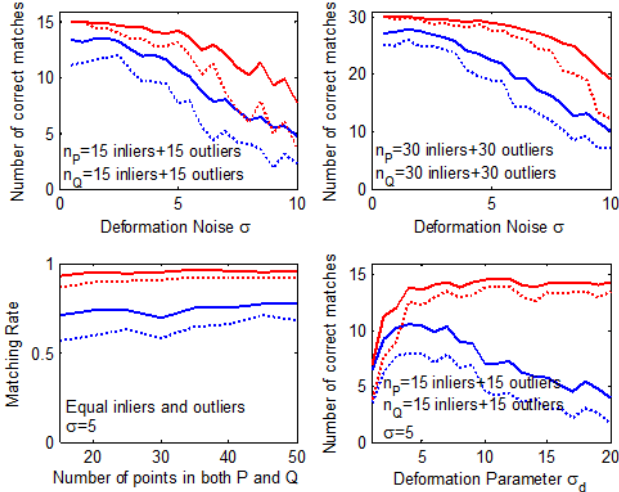


Figure 4. The meaning of curves are the same as in Figure 3. First row: varying deformation noise with equal inliers and outliers. Second row: keep equal inliers and outliers, change total numbers (left), change parameter σ_d (right) under fixed deformation $\sigma = 5$.

Figure 3 and Figure 4 compare the performance curves of our proposed method and spectral method in [8]. We keep the sensitivity parameter fixed, $\sigma_d = 5$. Both algorithms ran on the same data sets over 30 trials for each value of the varying parameter, and both the mean performance curves and the curves of one standard deviation below the mean are plotted. We score the performances of these two methods by counting how many matches agree with the ground truths. In the first row of Figure 3, there are no outliers, and we vary the noise σ from 0.5 to 10 (in step of 0.5). The performance curves are nearly the same, which indicates that both algorithms have similar ability in tolerating noise. In the second row of Figure 3, we keep the deformation parameter σ and the number of inliers fixed, and change the number of outliers. Obviously, our method is nearly not affected by outliers. Especially, when there is no deformation noise (left), our method has very high probability to find all correct matches. However, the performance of spectral method in [8] is continuously degrading as the number of outliers increases, even when there is no deformation noise. When the number of outliers exceeds the number of inliers, its performance becomes unacceptable. Since both methods operate on the same matrix A and optimize the same quadric function $f(x) = x^T A x$, the only difference is the constraints put on x : spectral method in [8] requires $|x|_2 = 1$ and our method requires $|x|_1 = 1$. Such a minor difference results in totally different meanings: *spectral method tries to partition all data, no matter inliers and outliers; our method, however, only tries to select some highly correlated data, which are usually inliers, and ignores outliers.* In the first row of Figure 4, we keep equal number of inliers and outliers, and change deformation noises. Obviously, our method works much better. We

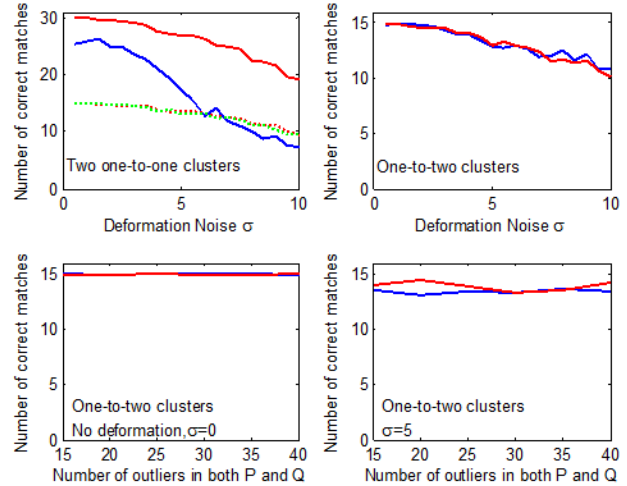


Figure 5. Performance curves on the cases with multiple common patterns. First row, left figure: two one-to-one common patterns, varying deformation noise. The red solid curve (our method) and the blue solid curve (spectral method) show the total numbers of correct matches, the red and green dashed curves show the number of each common pattern (our method). First row, right figure: one-to-two common patterns, varying deformation noise, and the number of correct matches for two common patterns are plotted (our method). Second row: one-to-two common patterns, varying number of outliers, without (left) and with (right) deformation noise (our method).

fix the deformation parameter $\sigma = 5$, keep the number of inliers and outliers equal, and change the total number of points in P and Q , the performance of both algorithms improves as the number of points increases, which is demonstrated on the left of the second row of Figure 4. This happens because as the number of inliers increases, each correct correspondence establishes pairwise relationships with more correct correspondences, thus becomes more robust to deformations and outliers. On the right, we also test the sensitivity of both methods to parameter σ_d , the spectral method in [8] is very sensitive to σ_d , but our method is much more robust.

5.2. Multiple Common Patterns Detection

Spectral method in [8] can only find one common pattern, which is usually not the case in real situations. Our method can find all common visual patterns and we conduct experiments to demonstrate this property. We first produce P and Q with two one-to-one common visual patterns, the size of each common pattern is fixed at 15. The result is demonstrated in the left figure of the first row of Figure 5. Note that for one certain common pattern, the other common pattern can be considered as outliers, thus spectral method works badly. The right figure of the first row demonstrated the result on one-to-two common patterns, which cannot be dealt with by the spectral method, our method works remarkably well. The second row demonstrates the effect of outliers, and the left figure shows the

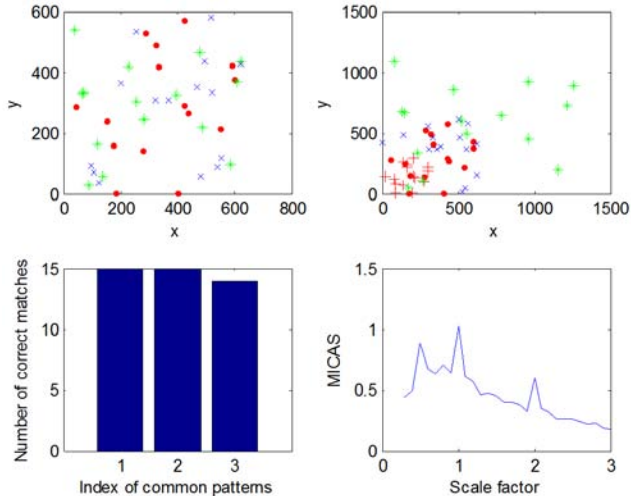


Figure 6. Common visual pattern detection on multi-scales. First row: P and Q . Second row: number of correct matches for three common patterns (left) and the average intra-cluster affinity score as scale varies. For better viewing, please see the original color pdf file.

results without noises, while the right figure shows the results with noise of $\sigma = 5$. The number of outliers changes from 15 (same as inliers of one common pattern) to 40, the result strengthens our conclusion: our method is very robust to outliers.

We also conduct an experiment on multi-scale common visual patterns, which is demonstrated in Figure 6. The first row shows P and Q . P contains three parts, after adding noise ($\sigma = 5$), one copy of red part is directly added into Q (red dot) and another copy of red part is scaled by 0.5, then added into Q (red plus). The blue part has been added noises, scaled by 2 and then added into Q (blue star). Both P and Q are then added with some outliers. In the second row, the left figure shows that our method can correctly detect the three common visual patterns. We also plot the maximal average intra-cluster affinity score (MICAS) as a function of the scale factor, which is in the right figure. As expected, it correctly indicates at which scale factor, there is common visual pattern. More specifically, at the scale 0.5, 1 and 2, this curve reaches the local maxima.

We generate synthesized images of ETHZ toys dataset [22] to demonstrates the ability of simultaneously detecting multiple common visual patterns of our proposed method. We use SIFT features to find candidate correspondences. As Figure 7 shows, despite large deformations, our method can correctly detect multiple common visual patterns at different scales.

5.3. Salient Points based Face Matching

In this subsection, we conduct an experiment on face matching. The face data is from [11], and there are 107 2D face images from 11 different people and each face is represented by 7 salient points. For each face image, we



Figure 7. Multiple common patterns detection under different scales and large deformation. For clarity, only parts of correspondences are plotted.

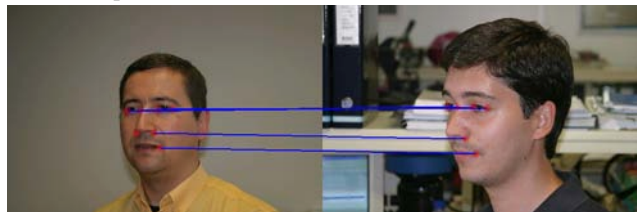


Figure 8. Exemplar correspondences between salient points of two faces.

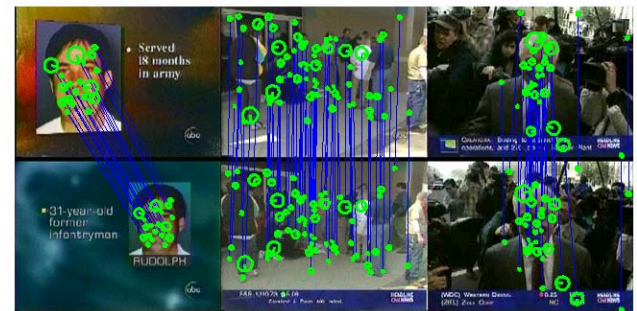


Figure 9. Common visual patterns on three near-duplicate images detected by our proposed method.

match it with all other face images and establish correspondences between salient points. The total number of possible correspondences is about $107 \times 106 \times 7 = 79394$. An exemplar matching result is demonstrated in Figure 8. We score the matching methods by the percent of correct correspondences, which is 97.1% of our proposed method, compared to about 95.7% of the subspace method in [11].

5.4. Near-duplicate Image Retrieval

In this subsection we show an application of our proposed method in near-duplicate image retrieval, which plays an important role in many real-world multimedia applications. The experiment is conducted on the Columbia database, which contains 150 near-duplicate pairs and 300

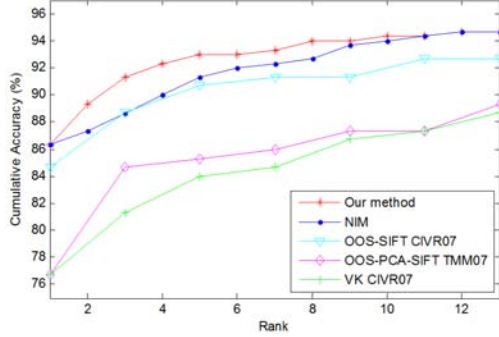


Figure 10. Comparison of cumulative accuracy of near-duplicate image retrieval on Columbia database.

non-duplicate images (600 images in total). For near-duplicate images, we expect that they have a large common visual patterns. For fair comparison, we first rank all images using global features as done in [21], then re-rank the images in top 50 based on the size of detected common visual patterns. We use SIFT features to find all possible point correspondences. The feature similarity score and geometric consistency score are defined as follows:

$$S(c) = \exp\left(-\frac{(i_d - i'_d)^2}{2\sigma_f^2}\right), \quad (7)$$

$$S(c_i, c_j)(s) = \begin{cases} 1 - \frac{(l_{ij} - sl_{i'j'})^2}{9\sigma_d^2} & \text{if } |l_{ij} - sl_{i'j'}| < 3\sigma_d, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where σ_f and σ_d control the sensitivity to noises of features and geometric deformations, respectively. In this experiment, we set $\sigma_f = 250$ and $\sigma_d = 10$, respectively. For each pair of images, we search for 11 scales, and the average time is 0.44s, thus it costs about 2 hours in total. Figure 9 demonstrates the common visual patterns detected on three near-duplicate images. Such common visual patterns usually indicate similar objects or scenes. In Figure 10, the retrieval performance is plotted and compared with NIM method [21], OOS-SIFT method [18], OOS-PCA-SIFT method [20] and Visual Keywords (VK) method [20]. Obviously, our method outperforms all other methods and gets the best cumulative accuracies (ratio between correctly retrieved images in top N images and total number of query images), which verifies that our method can correctly detect common visual patterns in real images.

6. Conclusions and Future Work

We have presented an effective solution to detecting common visual patterns in two images. Our method takes the advantages of the fact that the noises and outliers generally cannot form the pattern which is distinctive in both feature representation and spatial layout. In the point correspondence space, all common patterns form strongly connected clusters, no matter they are one-to-one, one-to-many, or many-to-many. Instead of using spectral method, which is very sensitive to noises and outliers, although guarantees to find global solution, we utilize l_1 -norm and find all large

local maxima (including the global maximum) through a systematic way of initialization. The experiments show that our method is very robust to outliers, and can correctly find all common visual patterns. Our future work shall focus on: 1) estimating the scale factor automatically, and 2) making the geometric consistency score affine-invariant.

7. Acknowledgement

This work is supported by National Research Foundation/Interactive Digital Media Program, under research Grant NRF2008IDMIDM004-029, Singapore.

References

- [1] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. *CVPR*, 2005. 1, 2
- [2] P. Besl and H. McKay. A method for registration of 3-D shapes. *TPAMI*, 1992. 1
- [3] T. Caetano, T. Caelli, D. Schuurmans, and D. Barone. Graphical models and point pattern matching. *TPAMI*, 2006. 1, 2
- [4] M. Carcassoni and E. Hancock. Alignment using spectral clusters. *BMVC*, 2002. 2
- [5] M. Cho, J. Lee, and K. Lee. Feature Correspondence and Deformable Object Matching via Agglomerative Correspondence Clustering. *ICCV*, 2009. 2
- [6] O. Enqvist, K. Josephson, and F. Kahl. Optimal Correspondences from Pairwise Constraints. *ICCV*, 2009. 1, 2
- [7] R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *TPAMI*, 1989. 2
- [8] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. *ICCV*, 2005. 2, 3, 4, 5, 6
- [9] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2
- [10] J. Maciel and J. Costeira. A global solution to sparse correspondence problems. *TPAMI*, 2003. 1, 2
- [11] M. Marques and J. Costeira. Subspace matching: Unique solution to point matching with geometric constraints. *ICCV*, 2009. 7
- [12] T. Motzkin and E. Straus. Maxima for graphs and a new proof of a theorem of Turan. *Canad. J. Math*, 1965. 3
- [13] M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *TPAMI*, 2007. 3, 4
- [14] M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. *TPAMI*, 1999. 2
- [15] L. Shapiro and J. Brady. Feature-based correspondence: an eigenvector approach. *IVC*, 1992. 2
- [16] L. Torresani and V. Kolmogorov. Feature correspondence via graph matching: Models and global optimization. *ECCV*, 2008. 2
- [17] J. Weibull. *Evolutionary game theory*. MIT Press, 1997. 3
- [18] X. Wu, W. Zhao, and C. Ngo. Near-duplicate keyframe retrieval with visual keywords and semantic context. *ACM CIVR*, 2007. 8
- [19] J. Yuan and Y. Wu. Spatial random partition for common visual pattern discovery. *ICCV*, 2007. 1
- [20] W. Zhao, C. Ngo, H. Tan, and X. Wu. Near-duplicate keyframe identification with interest point matching and pattern learning. *TMM*, 2007. 8
- [21] J. Zhu, S. Hoi, M. Lyu, and S. Yan. Near-duplicate keyframe retrieval by nonrigid image matching. *ACM MM*, 2008. 8
- [22] V. Ferrari, T. Tuytelaars, and L. Van. Simultaneous object recognition and segmentation from single or multiple model views. *IJCV*, 2006. 7