

Feature Correspondence and Deformable Object Matching via Agglomerative Correspondence Clustering

Minsu Cho Jungmin Lee Kyoung Mu Lee

Department of EECS, ASRI, Seoul National University, 151-742, Seoul, Korea

minsucho@diehard.snu.ac.kr jmlee.vision@gmail.com kyoungmu@snu.ac.kr

Abstract

We present an efficient method for feature correspondence and object-based image matching, which exploits both photometric similarity and pairwise geometric consistency from local invariant features. We formulate object-based image matching as an unsupervised multi-class clustering problem on a set of candidate feature matches, and propose a novel pairwise dissimilarity measure and a robust linkage model in the framework of hierarchical agglomerative clustering. The algorithm handles significant amount of outliers and deformation as well as multiple clusters, thus enabling simultaneous feature matching and clustering from real-world image pairs with significant clutter and multiple deformable objects. The experimental evaluation on feature correspondence, object recognition, and object-based image matching demonstrates that our method is robust to both outliers and deformation, and applicable to a wide range of image matching problems.

1. Introduction

Many problems in computer vision involve feature correspondences across images. The development of various local invariant features [13, 14, 15] has brought about significant progress in this area, and many algorithms exist for visual correspondence in a wide range of applications such as object recognition, image matching, 3D reconstruction and motion segmentation. Many of them typically use global and strong constraints for rigid motion [13, 11, 4] or exploit only appearance, discarding information about the spatial layout of features [5, 16].

Recently, robust feature correspondence methods [2, 12, 18] have been proposed for considering geometric distortion of objects between images. They formulate visual correspondence as a graph matching problem by defining an objective function based on both photometric similarity and pairwise geometric compatibility between correspondences. Despite the promising performance demon-



Figure 1. A result of our object-based image matching. It establishes both feature correspondences and their object-based clusters across arbitrary image pairs. Despite deformation of objects and high outlier ratio of candidate feature matches as shown in the middle, our method extracts multiple clusters of reliable matches, each representing a geometrically distinctive group of matches. It enables object-based image matching as shown in the bottom. Two detected clusters of feature correspondences are colored in red and green, respectively.

strated by the methods, they all deal with weakly supervised cases with relatively low outlier ratio, where two images have one common object or a model image is used. In real-world cases, however, image pairs can have significant clutter, multiple common objects, and even many-to-many object correspondences. Therefore, unlike the previous methods, feature correspondence problems need to be interleaved with finding multiple object-level clusters of

correspondences against significant outliers in an unsupervised way. Our goal is to establish both feature correspondences and their object-based clusters against significant clutter and deformation from arbitrary images. As shown in Fig. 1, our method provides object-based image matching from highly cluttered scenes, by taking into consideration both how well the features’ descriptors match and how well their pairwise geometric constraints are satisfied within each cluster. In this work, we focus on image matching with widely-used discriminative local features [13, 14, 15].

Our method is based on the following insights:

1. Bottom-up aggregation strategy: if we start from confident correspondences and progressively merge them with reliable neighbors, inliers can be effectively collected in spite of enormous distracting outliers. For example, seed-based exploration methods [7, 3, 10] demonstrate that object recognition performance can be boosted by such a bottom-up aggregation with iterative match-propagation.

2. Connectedness between parts: for deformable objects, feature correspondences do not form global compactness in their pairwise geometric similarity owing to deformation, but deformed parts are locally connected by some mediating parts. Thus, a connectedness criterion [8] should be considered for clustering the feature correspondences on deformable objects.

On the basis of the above insights, we propose a new algorithm in the framework of hierarchical agglomerative clustering [1, 17], which forms compact correspondence clusters in early stages and progressively merges locally connected clusters adapting to deformed object parts. By simply setting tolerable deformation and reliability thresholds, it detects multiple clusters of reliable matches and the number of reliable clusters is also estimated through the procedure. As shown in Fig. 1, it enables object-based image matching from arbitrary image pairs with significant deformation and clutter. Our experiments validate the effectiveness of our method in feature correspondence, object recognition, and object-based image matching problems.

Our approach has several advantages over previous methods. The methods in [2, 12, 18] commonly require a one-to-one (or one-to-many) feature mapping constraint assuming a weakly supervised model view, and do not deal with multiple object matches present between images. Thus, they are not adequate for general matching of unsupervised images. On the other hand, our approach provides reliable feature correspondence, object-level multi-class clustering, and outlier elimination in an integrated way. Its control parameters are simple and intuitive, and it does not require a global energy formulation [2, 18], strong global constraints [13, 4], nor a specified number of clusters [19]. Moreover, it is very robust to distracting outliers arising from clutter in real-world images.

2. Problem Formulation

We formulate our object-based image matching as unsupervised multi-class clustering of candidate feature correspondences as follows. Suppose two sets of features \mathcal{P} and \mathcal{Q} , each obtained from an image, and a set of candidate correspondences $\mathcal{L} \subset \mathcal{P} \times \mathcal{Q}$. For each candidate correspondence $c_i = (p, q) \in \mathcal{L}$, a unary dissimilarity score measures the dissimilarity between $p \in \mathcal{P}$ and $q \in \mathcal{Q}$. For each pair of correspondences (c_i, c_j) where $c_i, c_j \in \mathcal{L}$, pairwise dissimilarity score evaluates geometric deformation between c_i and c_j . Our goal is to construct clusters of mutually coherent feature correspondences eliminating outliers from \mathcal{L} . The solution is represented by a mutually disjoint set of correspondence clusters $\theta = \{C_1, C_2, \dots, C_n\}$ where $\forall C_k \in \theta, C_k \subset \mathcal{L}$. To reduce notational clutter we will sometimes abbreviate $c_i \in C_k$ as $i \in C_k$. Any kind of mapping constraints used in other methods [2, 12, 18] can be employed, such as: one-to-one constraint allowing one feature from \mathcal{P} to match at most one feature from \mathcal{Q} , or one-to-many constraint allowing one feature from \mathcal{P} to match more than one features from \mathcal{Q} . Our method, however, can dispense with such specific mapping constraints, allowing arbitrary cross-mapping.

3. Dissimilarity of Feature Correspondences

Our problem is settled by the distance metric that measures dissimilarity between two candidate feature matches. This can be regarded as the internal force driving feature correspondence and clustering. Various features and descriptors can be adopted in our framework, depending on specific problems. In this work, we use affine covariant region detectors [14, 15] for features, and SIFT [13] for descriptors, which are widely used in the literature. Thus, we represent a candidate feature correspondence $c_i = (p, q) \in \mathcal{L}$ as a candidate match $m_i = (x_i, x'_i, H_i)$, interchangeably, where x_i, x'_i, H_i denotes center of p , center of q , and homography from p to q , respectively [15].

Photometric dissimilarity $d_{\text{app}}(m_i)$ of a feature correspondence $c_i = (p, q)$ is defined by the Euclidean distance between corresponding SIFT descriptors of feature p and q . Since two features of a true correspondence are likely to have similar local descriptors, a set of candidate correspondences \mathcal{L} is usually constructed using this photometric dissimilarity. Geometric dissimilarity $d_{\text{geo}}(m_i, m_j)$ between two feature correspondences $m_i = (x_i, x'_i, H_i)$ and $m_j = (x_j, x'_j, H_j)$ is defined by

$$d_{\text{geo}}(m_i, m_j) = \frac{1}{2}(d_{\text{geo}}(m_j|m_i) + d_{\text{geo}}(m_i|m_j)), \quad (1)$$

$$d_{\text{geo}}(m_j|m_i) = \frac{1}{2}(|x'_j - H_i x_j| + |x_j - H_i^{-1} x'_j|), \quad (2)$$

$$d_{\text{geo}}(m_i|m_j) = \frac{1}{2}(|x'_i - H_j x_i| + |x_i - H_j^{-1} x'_i|), \quad (3)$$

where $|\cdot|$ denotes the Euclidean distance function. It corresponds to a projection error, which will be small if H_i and H_j are similar to each other. Exploiting the homographies of two correspondences, this geometric dissimilarity provide a discriminative measure for our agglomerative algorithm.

Combining the photometric and geometric dissimilarity, we define an overall pairwise dissimilarity function as

$$d(m_i, m_j) = d_{\text{geo}}(m_i, m_j) + \alpha \max(d_{\text{app}}(m_i), d_{\text{app}}(m_j)), \quad (4)$$

where α denotes a weighting factor for photometric dissimilarity. The max operator means that both of two correspondences in the pair should have low dissimilarity for a confident correspondence pair.

4. Algorithm

In this section we introduce hierarchical agglomerative algorithms and propose a new algorithm in the agglomerative framework for deformable object matching. The pairwise dissimilarity function defined in the previous section is used for the algorithm.

4.1. Hierarchical Agglomerative Clustering

Hierarchical agglomerative clustering algorithms are efficient and have been widely used in various fields [1, 17]. Hierarchical agglomerative clustering does not require explicit global model, and a particular algorithm can be obtained by the definition of the dissimilarity measure between two clusters, which determines the priority of a cluster pair to merge. Let $D(C_a, C_b)$ be a cluster dissimilarity function defined for all possible pairs of clusters. Then, the general agglomerative scheme can be stated as in Algorithm 1. It produces a sequence of agglomerative clusterings, decreasing the number of clusters at each step. The result consists of a sequence of nested data partitions in a hierarchical structure, graphically represented as a dendrogram. A partition can be obtained by various methods [17] (e.g. setting a cut-off threshold on the dendrogram). The main representatives of the algorithms are the single-link, the complete-link, and the average-link (also known as the unweighted pair group method average) algorithms [17].

The single-link algorithm defines the cluster dissimilarity function D as the minimum among all pairwise dissimilarities between elements of two clusters. It is appropriate for the recovery of elongated or connected clusters. In our problem formulation, the single-link dissimilarity between two correspondence clusters C_a and C_b is represented by

$$D_{\text{si}}(C_a, C_b) = \min_{i \in C_a, j \in C_b} d(m_i, m_j). \quad (5)$$

The complete-link algorithm uses the maximum of all pairwise dissimilarities between elements of two clusters,

Algorithm 1 General Hierarchical Agglomerative Scheme

- 1: (Initialization)
 - 2: Set $\theta_0 = \{C_i = \{c_i\}, i = 1, \dots, n\}$ as the initial clustering.
 - 3: $t = 0$
 - 4: (Hierarchical Agglomerative Clustering)
 - 5: **repeat**
 - 6: $t = t + 1$
 - 7: Among all possible pairs (C_r, C_s) in θ , find (C_a, C_b) such that $D(C_a, C_b) = \min_{r,s} D(C_r, C_s)$
 - 8: Define $C_q = C_a \cup C_b$ and produce the new clustering $\theta_t = (\theta_{t-1} - \{C_a, C_b\}) \cup \{C_q\}$.
 - 9: **until** all elements lie in a single cluster.
-

so tends to produce very tight clusters. The complete-link dissimilarity in our formulation is presented as

$$D_{\text{co}}(C_a, C_b) = \max_{i \in C_a, j \in C_b} d(m_i, m_j). \quad (6)$$

The average-link algorithm is a compromise between the single-link and the complete-link, resulting in an intermediate structure between the loosely bound single-link cluster and tightly bound complete-link clusters. The average-link dissimilarity is described in our formulation as

$$D_{\text{av}}(C_a, C_b) = \frac{1}{|C_a||C_b|} \sum_{i \in C_a} \sum_{j \in C_b} d(m_i, m_j). \quad (7)$$

Each linkage model has its own drawbacks for clustering. The single-link clustering can produce straggling clusters. Because the merge criterion is strictly local, a chain of elements can be extended for long distances without regard to the overall shape of the emerging cluster. This effect is called “chaining”. The complete linkage and the average linkage are robust to the chaining effect, but fails when the target clusters are not compact around their centers.

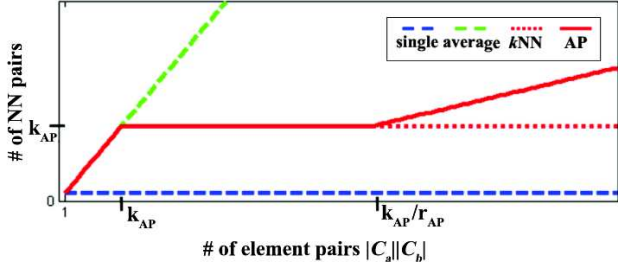
4.2. Adaptive Partial Linkage Model

For deformable objects of images, feature correspondences do not form global compactness in their pairwise geometric similarity due to deformation, and deformed parts are locally connected by some mediating parts. Thus, the linkage model for the problem should adapt to connectedness of elements as well as compactness of elements. As we noted, however, the single linkage model is susceptible to chaining outliers, and the other linkage models are essentially inclined toward a compactness criterion.

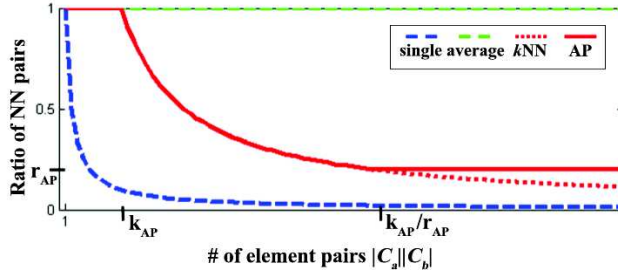
Therefore, we define a k NN linkage model as follows.

$$D_{k\text{NN}}(k, C_a, C_b) = \frac{1}{|\Gamma|} \min_{\Gamma} \sum_{(i,j) \in \Gamma} d(m_i, m_j), \quad (8)$$

$$s.t. \Gamma \subset C_a \times C_b, |\Gamma| = \min(k, |C_a||C_b|),$$



(a) Number of the nearest neighbor pairs used for dissimilarity



(b) Ratio of the nearest neighbor pairs used for dissimilarity

Figure 2. AP linkage compared to the other linkages. (a) shows changes of the number of NN pairs considered for the cluster dissimilarity between C_a and C_b as the number of all the possible element pairs $|C_a||C_b|$ increases. (b) depicts the ratio of the number of NN pairs to $|C_a||C_b|$. Two parameters, k_{AP} and r_{AP} controls compactness and connectedness of emerging clusters in the AP linkage. For kNN linkage, $k = k_{AP}$ in this graphs.

where $|C_a|$ denotes the number of elements in the cluster C_a . $|C_a||C_b|$ represents the number of possible element pairs between two clusters. Our kNN linkage model uses the average of k minimum dissimilarity among all possible dissimilarities of element pairs between the two clusters. While recovering elongated or connected clusters, it is robust to the chaining effect since it considers k supporting element pairs rather than just one element pair as in the single linkage. When $|C_a||C_b|$ is smaller than k , the kNN linkage is equivalent to the average linkage as shown in Fig. 2. Thus, the kNN -link clustering forms compact clusters in the early stages of agglomerative clustering, and progressively relaxes compactness and shifts to connectedness. Note that the compactness in the early stages is appropriate in the clustering procedure. In agglomerative clustering, merging chaining outliers in the early stages can be critical in the next steps. The early compactness of the kNN linkage helps to avoid this problem. However, it can still suffer from straggling when $|C_a||C_b|$ grows much larger than k .

To avoid this ‘‘asymptotic chaining effect’’, we propose the adaptive partial (AP) linkage model that uses an adaptive method for determining k of the kNN linkage as follows.

$$D_{AP}(C_a, C_b) = D_{kNN}(k, C_a, C_b), \quad (9)$$

$$\begin{cases} k = k_{AP} & \text{if } |C_a||C_b| \leq k_{AP}/r_{AP} \\ k = r_{AP}|C_a||C_b| & \text{if } |C_a||C_b| > k_{AP}/r_{AP}, \end{cases} \quad (10)$$

where k_{AP} and r_{AP} denote the control parameters. As shown in Fig. 2(a), the AP linkage is equivalent to the kNN linkage when $|C_a||C_b|$ is smaller than k_{AP}/r_{AP} . Otherwise, it increases the supporting NN pairs to $r_{AP}|C_a||C_b|$. In terms of the ratio of the NN pairs to the total number of element pairs $|C_a||C_b|$ as shown in Fig. 2(b), the kNN linkage continuously converges to the single linkage as $|C_a||C_b|$ increases. The AP linkage, however, always examines at least r_{AP} of all the element pairs. Thus, the AP linkage effectively avoids the asymptotic chaining effect of the kNN linkage and accommodates deformation of objects supported by intermediate joint correspondences. Moreover, this characteristic of AP-link clustering can be adjusted or learned by control parameter k_{AP} and r_{AP} , depending on the specific problems. Integrating the AP linkage in the framework of hierarchical agglomerative clustering, we can progressively aggregate elements considering both compactness and connectedness.

4.3. Agglomerative Correspondence Clustering

To solve feature correspondence and clustering problem defined in Sec.2, we propose an algorithm based on the AP linkage, which reflects the connectedness between deformable object parts as well as the compactness in the object parts. The algorithm is summarized in Algorithm 2. At each agglomerating step, clusters with inlier correspondences are likely to merge into larger clusters, and clusters with outliers are likely to remain as smaller clusters. The agglomerating step is iterated until the minimum dissimilarity $\min_{r,s} D_{AP}(C_r, C_s)$ is larger than a tolerable dissimilarity threshold δ_D . After the iteration ends, isolated clusters are obtained.¹

Note that different kinds of specific mapping constraints such as one-to-one or one-to-many feature mapping can be adopted in our algorithm. In that case, all the correspondences conflicting with those in the merged cluster are removed from \mathcal{L} at the step 10 of Algorithm 2. Our method, however, can be applied to general cases, allowing for any cross-mapping.

This agglomerative framework is useful for multi-class clustering with many distracting outliers. Particularly in our problem of object-based image matching with real-world images, significant outlier correspondences disturb the capture of inliers since some of the outlier pairs also have high similarity incidentally. However, after iterations of our algorithm under a tolerable dissimilarity value δ_D , outliers are

¹Other cluster isolation criteria can also be used such as dissimilarity increment criterion in [9]

Algorithm 2 Agglomerative Correspondence Clustering

- 1: (Initialization)
 - 2: Establish a set of candidate correspondences
 $\mathcal{L} = \{c_1, c_2, \dots, c_n\}$
 - 3: Set i th cluster $C_i = \{c_i\}$ ($i = 1, \dots, n$)
 - 4: $\theta_0 = \{C_1, C_2, \dots, C_n\}$
 - 5: (Agglomerative Clustering)
 - 6: **repeat**
 - 7: $t = t + 1$
 - 8: Find C_a, C_b
s.t. $D_{AP}(C_a, C_b) = \min_{r,s} D_{AP}(C_r, C_s)$
 - 9: Merge C_a and C_b into a single cluster C_q
 - 10: Remove from θ_{t-1} all correspondences in conflict with those in C_q (given a mapping constraint)
 - 11: $\theta_t = (\theta_{t-1} - \{C_a, C_b\}) \cup \{C_q\}$
 - 12: **until** $|\theta_t| == 1$ or $\min_{r,s} D_{AP}(C_r, C_s) > \delta_D$
 - 13: (Cluster Selection)
 - 14: Eliminate unreliable clusters from θ
-

likely to form many small correspondence clusters while inliers lie in large clusters. Therefore, we can select reliable clusters of inliers by choosing well-formed large clusters, eliminating small and trivial outlier clusters. Typically, inlier clusters are likely to get enough numbers of correspondences and spread over sufficient areas. Thus, for evaluating reliability of a cluster, we propose simple reliability criteria as follows.

1. Both of its convex hulls should be larger than $\tau_a\%$ of their entire image area.
2. The number of elements in the cluster should be more than τ_m .

All trivial clusters against the above criteria are removed, and thus outliers are effectively eliminated even if no prior mapping constraints are provided.

5. Experiments

We evaluate the robustness of our method on three tasks. The experiments demonstrate the performance of our method on feature correspondence, object recognition, and object-based image matching, respectively.

5.1. Feature Correspondence with Outliers, Deformation, and Multiple Object Matches

In this experiment, we generate synthesized images to simulate cluttered scenes where deformable objects appear. The purpose is to quantitatively evaluate the feature correspondence performance of our algorithm on various scenes.

We generate 3×3 tiled images pairs, which contain N_c common sub-image(s) as shown in Fig. 3. The ETHZ toys dataset² is used for generating the tiled images. N_c com-



Figure 3. An example of synthesized image pairs for experiment. 3×3 tiled image forms each of the two images, and the pair contains N_c common sub-image(s) as indicated by dashed lines ($N_c = 1$ for this example). Deformation is applied to the right image using TPS. See the text.

mon sub-images of the tiled image pair are randomly taken from model images of the dataset³, the other sub-images are randomly selected from test images of the dataset. The position of all the sub-images on the tiled images are also randomly determined. Then, we impose deformation on one of the tiled image pair using Thin-Plate-Spline (TPS) model as shown in the right of Fig. 3. To deform the image, we select 9×9 crossing points from a 10×10 meshgrid of the entire image and use them as the control points of TPS. All the control points are perturbed by Gaussian noise of $N(0, \sigma_n)$ independently, then TPS warping is applied. Since we can identify the true corresponding points between the image pair using the TPS model, we can quantitatively evaluate the feature correspondence scores, while controlling the amount of outlier, deformation, and the number of common objects.

We compare the performance of our method with the spectral method of [12]. To obtain initial candidate feature matches for both algorithms, we use the MSER feature detector [14] and the SIFT descriptor [13]. The best 1200 candidate correspondences among all possible ones are collected according to the distance of SIFT descriptor. For fair comparison, our dissimilarity measure of (4) was commonly used for both algorithms with $\alpha = 0$. The control parameters are fixed as $k_{AP} = 10$ and $r_{AP} = 0.05$. For each agglomeration step, we eliminate the candidate matches conflicting with merging matches based on the one-to-one mapping constraint as in [12], and the agglomeration iterated until a single cluster remains. Thus, cluster selection is not performed in this experiment. To generate the affinity matrix M used in [12], we set $M(i, j) = \max(\beta - d(m_i, m_j)/\sigma_s^2, 0)$ based on our dissimilarity measure. As noted in [12], the parameters of affinity function, σ_s and β controls the sensitivity to deformation just like the control parameter k_{AP} and r_{AP} in our algorithm. We tuned σ_s and β to generate its best average in our trials: $\beta = 40$, $\sigma_s = 1.0$.

³All the model images are tightly segmented for eliminating background.

²<http://www.robots.ox.ac.uk/~ferrari/datasets.html>

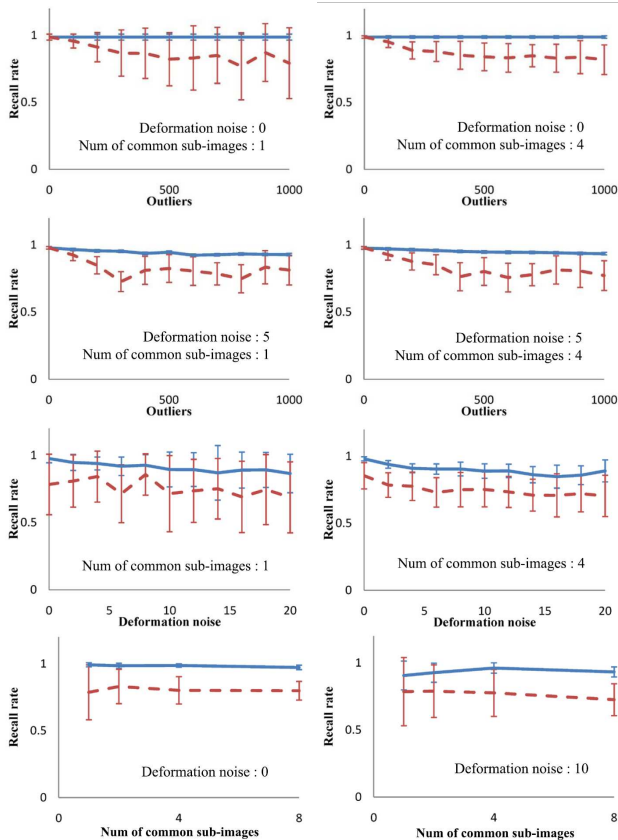


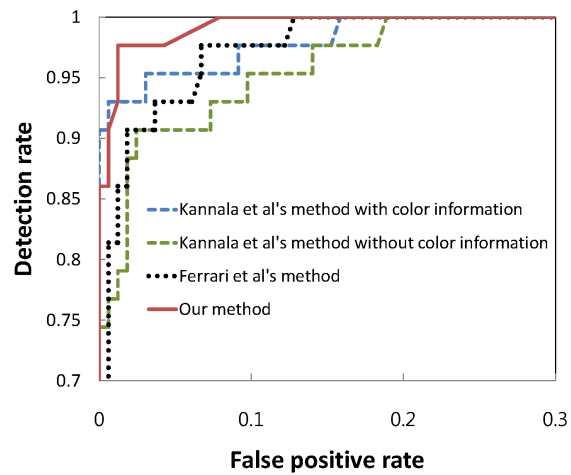
Figure 4. Performance comparison for our method vs. spectral method [12]. The average recall rate represented as a solid blue line for ours and a red dashed line for the method of [12]. *std* is shown as vertical bar. The first two rows: both deformation noise and the number of common sub-image pairs are fixed, varying the number of outliers. The third row: the number of common sub-image pairs is fixed and all candidate matches are used, varying the deformation noise. The fourth row: deformation noise is fixed and all candidate matches are used, varying the number of common sub-image pairs.

We measure the performance by calculating the recall rate of each algorithm under the one-to-one feature mapping constraint. Both algorithms ran 30 times to produce the mean values and the standard deviations. Figure 4 shows the performance curves of our method vs. the spectral method [12] as we vary the number of outliers, the amount of deformation, and the number of common sub-images or objects. The first two rows show the performance with varying the number of outliers. To control the number of outliers in the experiments, all the candidate correspondences are classified into two sets, inliers and outliers in advance. Then, starting from the set of only inliers, we gradually increase the outliers in the set. The third row shows the performance curve varying deformation factor σ_n , and the last row varying the number of common sub-images N_c .

The experiments demonstrate that our method clearly



(a) Image matching examples between model and test images



(b) ROC curves on the ETHZ toys dataset

Figure 5. Object recognition on the ETHZ toys dataset

outperforms than the spectral method [12] except the cases that few outliers exist. When the amount of deformation or the number of common objects is varied without controlling outliers, our method consistently shows higher average performance with lower variance. However, both performance curves of two algorithms have the similar decreasing rate. It suggests that the major factor explaining the performance difference between two algorithms is the amount of outliers. The spectral method is prone to the adverse effect of outliers since all candidates are analyzed as a whole in the framework. Thus, the eigenvectors of the affinity matrix M in the spectral method [12] are gradually perturbed by increasing outliers. Our method, however, starts from confident correspondence pairs and progressively merge them with reliable neighbors in a bottom-up way exploiting both compactness and connectedness. It avoids the adverse effect of distracting outliers.

5.2. Object Recognition

In this experiment, we test our method on view-based object recognition problem with the ETHZ toys dataset. The performance was quantified by processing all pairs of model and test images, and compared with Ferrari *et al.*'s [6] and Kannala *et al.*'s [10] reported in their papers. The ROC curves in Fig. 5 depicts the detection rate versus false-positive rate of all the methods, while varying the detection threshold from 0 to 200 matches. A model object is detected if the number of produced matches, summed over all its model views, exceeds this threshold. For our algorithm, the initial candidate feature matches are obtained in the same way of the previous experiment, and control parameters of AP linkage are fixed as: $k_{AP} = 10$ and $r_{AP} = 0.01$. and we set $\delta_D = 25$ for the tolerable deformation and $\tau_a = 1\%$, $\tau_m = 0$ for cluster selection. As shown in the plot above, our method outperforms the state-of-the-arts recognition methods of both [6] and [10], and achieves 98% detection with 1% false-positives. The results demonstrate impressive performance of our algorithm on inlier collection in feature-based image matching for recognition. Examples of model-test view matching are also shown in Fig. 5. Note that unlike [6] and [10], in this experiment, we did not use the color information, which might improve our results further.

5.3. Object-based Image Matching

In this section we test our method on the task of object-based image matching using cluttered image pairs. We used all the 23 test images of the ETHZ toys dataset without any model image, which include several deformable objects with significant clutter, occlusion, and view changes. We attempted to detect object matching between all the combination pairs of 23 images, 253 image pairs. All the settings of our algorithm was fixed as the previous experiment, except $\tau_m = 10$. For each combination image pairs, we assigned a matching score as the total area of the convex hull of the detected correspondence clusters. Figure 6(a) shows the several image pairs with high scores, where convex hulls of detected clusters are indicated by colored lines. In spite of significant deformation and view changes, our method successfully detects matching image pairs and common objects. For evaluating object-based image matching performance, we ranked all the combination pairs based on the matching scores, and examine from the top score image whether the image pair have common objects or not. The right graph of Fig. 6(a) shows how the precision and recall rate changes as we increase the number of image pairs from the top score. It demonstrates that most of image pairs including matching objects are highly ranked by our method. To evaluate the detection rate of the object matching, we checked all detected correspondence clusters based on the object matching criterion: 50% overlap with the ground-

truth common object region. Our method detected 83.6% (51 out of 61) among all the object correspondence present in the 253 image pairs.

Figure 6(b) shows the matching results on various images collected from Caltech dataset and Flickr site, which include multiple object correspondences with clutter and severe deformation. In this experiment, we fixed the parameters as follows: $k_{AP} = 30$, $r_{AP} = 0.05$, $\delta_D = 40$, $\tau_a = 3\%$, $\tau_m = 10$. As shown in Fig. 6(b), despite appearance change and geometric deformation, our method detects multiple object-level correspondences across cluttered images. The last two images in the bottom show object matching results within a single image. The results demonstrate that our method is useful for unsupervised object-based matching and category learning.

6. Conclusion

We have presented a novel approach to unsupervised object-based image matching across arbitrary images. Our method establishes both feature correspondences and their object-based clusters, effectively eliminating outliers from highly noisy candidate matches. The method is simple and constructed in a general clustering framework, which makes it applicable to many other unsupervised clustering problems. We demonstrate that our method not only achieves good performance, but also has several advantages for feature matching and object-based clustering problem. We believe our method is useful for a variety of vision applications dealing with real-world images such as unsupervised object matching, content-based image retrieval, unsupervised category recognition, and reconstruction.

Acknowledgements

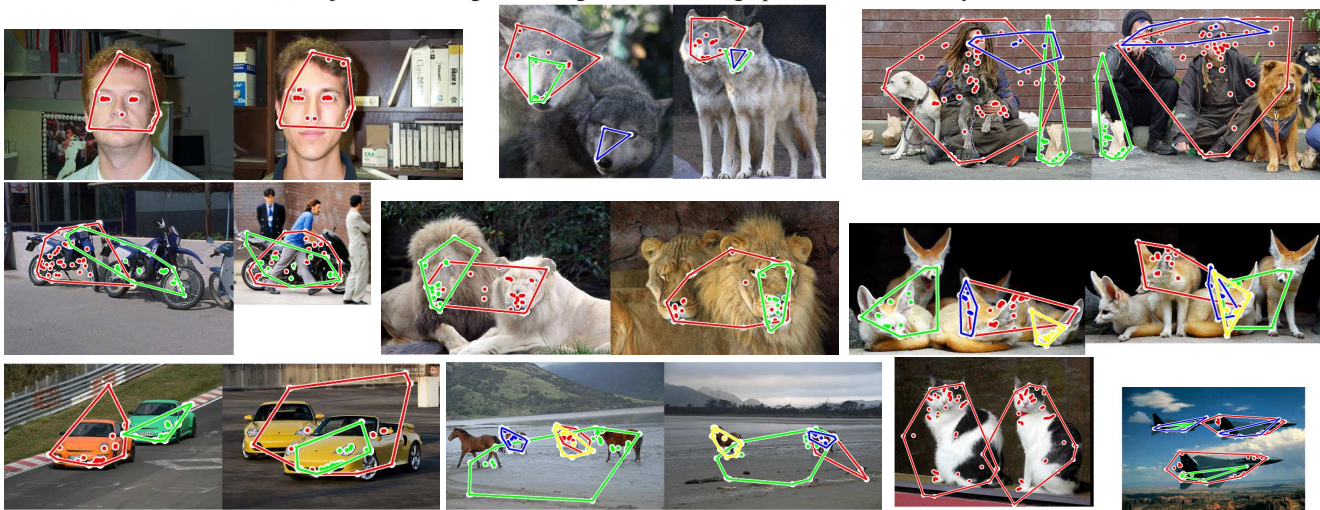
This work was supported in part by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2008-314-D00377), and in part by the IT R&D program of MKE/IITA (2008-F-030-01, Development of Full 3D Reconstruction Technology for Broadcasting Communication Fusion).

References

- [1] A.K.Jain and R.C.Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1998. 2, 3
- [2] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. *CVPR*, pages I: 26–33, 2005. 1, 2
- [3] M. Cho, Y. M. Shin, and K. M. Lee. Co-recognition of image pairs by data-driven monte carlo image exploration. *ECCV*, pages IV: 144–157, 2008. 2
- [4] O. Chum and J. Matas. Matching with PROSAC: Progressive sample consensus. *CVPR*, pages I: 220–226, 2005. 1, 2



(a) Object-based image matching on the test image pairs of the ETHZ toys dataset



(b) Object-based image matching on various images.

Figure 6. Object-based image matching results. All the detected correspondence clusters are indicated by different colors. (a) Results on the test image pairs of the ETHZ toys dataset. Four matching pairs with high scores are shown. The right graph presents the precision and recall rate on the top score pairs, the number of which is increasing. See the text. (b) Results on various images from Caltech dataset and Flickr. The last two examples shows object matching within a single image.

- [5] G. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. *ICCV*, pages 634–640, 2003. [1](#)
- [6] V. Ferrari, T. Tuytelaars, and L. Gool. Simultaneous object recognition and segmentation from single or multiple model views. *IJCV*, 67(2):159–188, 2006. [7](#)
- [7] V. Ferrari, T. Tuytelaars, and L. J. V. Gool. Simultaneous object recognition and segmentation by image exploration. *ECCV*, 2004. [2](#)
- [8] B. Fischer and J. M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *PAMI*, 25(4):513–518, 2003. [2](#)
- [9] A. L. N. Fred and J. M. N. Leitao. A new cluster isolation criterion based on dissimilarity increments. *PAMI*, 25(8):944–958, Aug. 2003. [4](#)
- [10] J. Kannala, E. Rahtu, S. Brandt, and J. Heikkila. Object recognition and segmentation by non-rigid quasi-dense matching. *CVPR*, pages 1–8, 2008. [2](#), [7](#)
- [11] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. *ICCV*, pages II: 508–515, 2001. [1](#)
- [12] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. *ICCV*, pages II: 1482–1489, 2005. [1](#), [2](#), [5](#), [6](#)
- [13] D. G. Lowe. Object recognition from local scale-invariant features. *ICCV*, pages 1150–1157, 1999. [1](#), [2](#), [5](#)
- [14] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *BMVC*, 2002. [1](#), [2](#), [5](#)
- [15] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, Oct. 2004. [1](#), [2](#)
- [16] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. *ICCV*, 2005. [1](#)
- [17] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, 3rd Edition*. Academic Press, 2006. [2](#), [3](#)
- [18] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. *ECCV*, pages II: 596–609, 2008. [1](#), [2](#)
- [19] S. X. Yu and J. B. Shi. Multiclass spectral clustering. *ICCV*, pages 313–319, 2003. [2](#)