A constrained clustering based approach for matching a collection of feature sets

Junchi Yan*
East China Normal University Shanghai Jiao Tong University IBM Research – China yanjc@cn.ibm.com

Zhe Ren
Shanghai, China sunshinezhe@sjtu.edu.cn

Hongyuan Zha
East China Normal University Georgia Institute of Technology zha@cc.gatech.edu

Stephen Chu
IBM Watson Research Center Yorktown, New York, USA schu@us.ibm.com

Abstract—In this paper, we consider the problem of finding the feature correspondences among a collection of feature sets, by using their point-wise unary features. This is a fundamental problem in computer vision and pattern recognition, which also closely relates to other areas such as operational research. Different from two-set matching which can be transformed to a quadratic assignment programming task that is known NP-hard, inclusion of merely unary attributes leads to a linear assignment problem for matching two feature sets. This problem has been well studied and there are effective polynomial global optimum solvers such as the Hungarian method. However, it becomes ill-posed when the unary attributes are (heavily) corrupted. The global optimal correspondence concerning the best score defined by the attribute affinity/cost between the two sets can be distinct to the ground truth correspondence since the score function is biased by noises. To combat this issue, we devise a method for matching a collection of feature sets by synergetically exploring the information across the sets. In general, our method can be perceived from a (constrained) clustering perspective: in each iteration, it assigns the features of one set to the clusters formed by the rest of feature sets, and updates the cluster centers in turn. Results on both synthetic data and real images suggest the efficacy of our method against state-of-the-arts.

I. INTRODUCTION

Finding feature correspondence between two or more feature sets is a fundamental problem in computer vision and pattern recognition, which also relates to operational research as it involves solving assignment problems. As a building block, it facilitates various problems e.g. 3-D reconstruction, CAD, visual tracking, common object discovery, among a considerable amount of applications. Most existing methods for feature correspondence have been devoted into the pairwise case i.e. matching two feature sets one time. Nevertheless it is far more common in real-world problems where a batch of feature sets are involved.

This paper focuses on the problem that given multiple (more than two) feature sets, how to find their one-to-one correspondences based on the unary node-wise attributes without considering the higher-order compatibility between nodes. For the classical two-set setting, the problem usually can be transformed to a linear assignment problem whose global optimum can be found in polynomial time using the Hungarian method. While this paper explores the setting where a collection feature sets are involved for feature matching.

II. RELATED WORK

We mention a few of relevant concepts about the correspondence problem, mainly in the context of computer vision.

A. Point registration and feature matching

For point registration, often a parametric transformation is assumed which serves as a regularization or prior term in the objective to account for the motion in an image sequence. The registration problem usually involves two ‘chicken-and-egg’ steps: i) finding correspondence from two point sets; ii) estimating the parameters of the transformation based on the correspondence. For instance, based on an initial correspondence, the iterative closest point (ICP) methods iterate between finding the correspondence via nearest neighbor and updating the transformation with the least square error. Many other methods, such as the Robust Point Matching (RPM) and the Coherent Point Drift (CPD) method are designed to perform pairwise registration. In contrast, the multi-view methods aim to find the correspondence and geometric transform over a batch of point sets.

Similar to the problem considered in this paper, there are several methods that only explore the unary feature attributes associated with the nodes without imposing geometrical priors. Two examples are graph matching and tree structure. The former formulates the feature matching among multiple sets as a matrix low rank sparsity decomposition task; the latter proposes an optimization algorithm to find consistent correspondences over feature sets.

B. Graph matching

In contrast to point registration, there is no parametric transformation imposed in graph matching thus we call it a non-parametric model. Moreover, compared with feature matching, graph matching additionally incorporates the edge attribute which can refer to second-order or even higher-order geometrical information for matching. This lifts the order of the matching problem, and a quadratic assignment programming formulation is derived which in general is known NP-hard and only a few special cases can be solved in polynomial time, such as planar graph, bounded valence graph, and tree structure. Note there are several state-of-the-art multiple graph matching methods by which the multiple feature sets matching problem can be viewed as a special case
in their model by ignoring the edge attributes, however, as will be shown later in this paper, our method specifically leverages the special structure of the problem for the unary feature that enables the proposed clustering based approach. These methods will also be compared in our tests.

Note the incorporation of higher-order information or transform prior as done in graph matching or point registration may be beneficial in certain practical tasks, yet this is a problem in parallel as they use different assumptions and information.

C. Matching cost modeling

It is theoretically proved in [13] that the graph edit costs is critical to the graph edit distance based methods. In complex tasks a manual procedure for cost setting is difficult, or even impossible to apply. To address this issue, [42] aims to learn the edit cost by a probabilistic framework, to reduce the intra-class edit distance and increase the inter-class one. Alternatively, [31] proposes to use self-organizing maps to learn the edit cost. For graph matching, recent work leverage various machine learning algorithms for computing the optimal affinity matrix [5], [11], and these methods in general fall into either supervised [5], unsupervised [11], or semi-supervised [17] learning paradigms, based on to what extent the supervision information is used. The cost modeling is orthogonal to this paper and we assume the cost is given.

D. Correspondence consistency over multiple node sets

Matching consistency around multiple node sets is a widely recognized concept such as [23], [31]. The key idea is illustrated in Fig.1 given the pairwise correspondence set \( X = \{X_{ij}\}_{i,j=1}^{N} \) independently computed by a two-set matching solver from \( N \) feature sets – we call \( X \) as matching configuration in line with [31] where \( X_{ij} \) is the node correspondence permutation matrix for feature set \( S_i \) and \( S_j \) there often exists the inconsistency over different node correspondence transitive paths. Now we further introduce two definitions concerning consistency as presented in [24], [31] (we slightly rewrite them to better fit to our problem).

**Definition 1.** Given \( N \) feature sets \( \{S_k\}_{k=1}^{N} \) and the pairwise matching configuration \( X = \{X_{ij}\}_{i,j=1}^{N} \), the unary consistency of feature set \( S_k \) is defined as \( C_u(k, X) = 1 - \frac{\sum_{i,j=1}^{N} ||X_{ij} - X_{ik}X_{kj}||_F^2}{N(N-1)/2} \in (0, 1] \), where \( \| \cdot \|_F \) is the Frobenius norm.

**Definition 2.** Given \( N \) feature sets \( \{S_k\}_{k=1}^{N} \) and matching configuration \( X \), for any pair \( G_i \) and \( G_j \), the pairwise consistency is defined as \( C_p(X_{ij}, X) = 1 - \frac{\sum_{i,j=1}^{N} ||X_{ij} - X_{ik}X_{kj}||_F^2}{N^2(N-1)/2} \in (0, 1] \), where \( n \) is the number of feature points in each feature set.

In the next section, we will present our method where the above two definitions will be used.

Specifically, the element \( x_{ij}^{st} \) in \( X_{ij} \) denotes the sth node in feature set \( S_i \) corresponds to the tth node in feature set \( S_j \) if \( x_{ij}^{st} = 1 \), otherwise \( x_{ij}^{st} = 0 \).

III. PROPOSED METHOD

For two-set feature matching with set \( S_1 \) and \( S_2 \), by merely taking the unary attributes into consideration, the matching problem can be formulated as a linear assignment problem, for \( n_1 \leq n_2 \) without loss of generality:

\[
\min_{x_{ij}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} c_{ij} x_{ij}
\]

s.t. \( \sum_{j=1}^{n_2} x_{ij} = 1, \sum_{i=1}^{n_1} x_{ij} \leq 1, x_{ij} \in \{0, 1\} \)

Or in a more compact form:

\[
\min_{X} \vec{\text{vec}}(C)^T \vec{\text{vec}}(X)
\]

s.t. \( X_{1n_2} \leq 1 \)

The operation \( \text{vec}(\cdot) \) stacks the columns of the input matrix into a column vector. The superscript \( T \) denotes the transformation of a given matrix or vector, and \( c_{ij} \) is the element in matrix \( C \). Here \( X \) is a partial permutation matrix.

The cost matrix \( C \) can be modeled in different ways based on the applications. In line with [1], we compute it by an exponential cost of the feature vector distance. Formally, let the feature vectors of feature set \( S_i \) be \( F_i = [f_{i1}, f_{i2}, \ldots, f_{iM}] \) where \( \{f_{ij}\}_{j=1}^{M} \in \mathbb{R}^D \) is the feature vector of feature \( t \) in \( S_i \), thus \( F_i \in \mathbb{R}^{D \times n_i} \). Then the cost between assigning feature \( s \) in one feature set to feature \( t \) in the other is computed by:

\[
\{c_{st} = \exp \left( -||F_s - F_t||_2/\sigma^2 \right) \}_{s=1,t=1}^{n_1,n_2}
\]

we set the scale parameter \( \sigma^2 \) to 0.15 in this paper which is found insensitive to the performance, and \( \| \cdot \|_2 \) is the \( \ell_2 \) norm.

The above linear assignment problem has been well studied in early years and polynomial solvers that can attain global optimum are devised such as the Hungarian method [25] and its Jonker-Volgenant alternative [26].

As a common preprocessing step, we replace the inequality constraint with the equality constraint as follows:

\[
\min_{X} \vec{\text{vec}}(C)^T \vec{\text{vec}}(X)
\]

s.t. \( X_{1n_2} = 1, 1^T \) \( n_1 \times 1 \times X = 1_{n_2}^T, X \in \{0, 1\}^{n_1 \times n_2} \) \n
This can be fulfilled by adding dummy nodes to feature set \( S_1 \) of smaller size (i.e. adding slack variables to the assignment matrix and augment the affinity matrix by zeros) for \( n_1 \neq n_2 \), then \( n_1 = n_2 = n \). This is a standard technique from linear
initial two-set feature correspondences: \( \mathbf{X}_i \) \( i \in [1,N] \), \( \mathbf{X}_j \) \( j \in [1,N] \), \( i \neq j \), by the Hungarian method; we propose an iterative such as k-mean cannot be directly applied. To address node correspondence. Hence traditional clustering methods be assigned to the same cluster since we assume one-to-one problem, there is one combinatorial constraint imposed on the correspondence across the sets can be viewed as one cluster comprised of the corresponding feature points from each set. Different colors correspond to different feature sets. The feature points viewed as a procedure to group each feature node in each set.

In this illustration, there are 5 feature sets with 5 features in each of the set. Each color of the 6th feature set are assigned to the clusters based on their distance to the cluster centers which can be computed using the Hungarian methods.

Bottom: a more intuitive illustration of our method matchCluster whereby different shapes denote clusters and same color indicate one feature set.

programming and is adopted as a robust means by the literature e.g. [16], [24], [31] to handle unmatchable nodes.

However, directly applying the Hungarian method to obtain the mathematical global optimum may not lead to the perfect matching in real-world problems. There are two reasons for the existence of such an ambiguity: i) modeling the cost/affinity function is non-trivial and it is difficult to fit a score function fully unbiased with the matching accuracy; ii) the noises further make the correlation between matching accuracy and score function more biased, especially only two feature sets are given. In contrast, a collection of feature sets is expected to provide additional global information to help disambiguate the local noises by information fusion.

As a result, we consider the multiple feature set matching problem as a clustering task in the feature space: each feature correspondence across the sets can be viewed as one cluster comprised of the corresponding feature points from each set. From this perspective, the matching problem can be viewed as a procedure to group each feature node in each set into different clusters. Different from the classical clustering problem, there is one combinatorial constraint imposed on the clustering procedure: any two feature nodes in one set cannot be assigned to the same cluster since we assume one-to-one node correspondence. Hence traditional clustering methods such as k-mean cannot be directly applied. To address this constrained clustering problem, we propose an iterative algorithm as described in Alg.1 we first generate a group of initial two-set feature correspondences: \( \{X_i\}_{i=1}^{N} \), \( \{X_j\}_{j=1}^{N} \) where \( S_r \) is a reference feature set. Using this reference feature set \( F_r \) and the correspondences: \( \{X_i\}_{i=1}^{N} \), \( \{X_j\}_{j=1}^{N} \), we can obtain a series of aligned feature matrix: \( F'_1, F'_2, \ldots , F'_N \) by \( F'_r = F_r X_r \), then we choose a certain feature set \( F_k \), and compute the mean of the rest feature sets: \( F_k = \sum_{i=1,i \neq k}^{N} F'_i / (N-1) \). Then we choose the node-to-node cost between \( F_k \) and \( F'_k \), and we compute the correspondence between \( F_k \) and \( F'_k \), we can obtain a series of matched feature sets: \( F_1, F_2, \ldots , F_N \) by Alg.1.

In the above assignment procedure in one iteration is illustrated in Fig.2. We term Alg.1 by a superscript ‘fast’ because it is initialized by a linear number of two-set matchings regarding \( N \). However, as discussed in a similar situation in [24], since our method is an alternating updating procedure, the final solution can

![Algorithm 1](image1.png)

**Algorithm 1** Fast constrained clustering based multiple feature sets matching (matchCluster<sup>fast</sup>)

**Input:** features matrix of each feature set: \( \{F_i\}_{i=1}^{N} \)

1. Randomly choose a reference feature set \( F_r \) and compute two-set correspondences \( \{X_i\}_{i=1}^{N} \), \( \{X_j\}_{j=1}^{N} \) by the Hungarian method;
2. Compute the aligned feature matrix by \( \{F'_i = F_r X_r\}_{i=1}^{N} \);
3. for \( k = 1 : N, k \neq r \) do
4. Compute the mean feature matrix \( F_k = \sum_{i=1,i \neq k}^{N} F'_i / (N-1) \);
5. Update \( X_k, k \) by computing the correspondence between \( F_k \) and \( F'_k \) via the Hungarian method;
6. end for

![Algorithm 2](image2.png)

**Algorithm 2** Constrained clustering based multiple feature sets matching (matchCluster)

**Input:** features matrix of each feature set: \( \{F_i\}_{i=1}^{N} \)

1. Compute the two-set feature correspondences \( \{X_i\}_{i=1}^{N} \) for each pair of sets independently by the Hungarian method;
2. Compute unary feature set consistency \( \{C_r(k, X)\}_{k=1}^{N} \) by Def.1, and choose the reference \( S_r = \max_r \{C_r(k, X)\} \) ;
3. Compute pairwise consistency \( \{C_k(X_r, X)\}_{i=1,i \neq r}^{N} \) by Def.2;
4. Compute the aligned feature matrix by \( \{F'_i = F_r X_r\}_{i=1,i \neq r}^{N} \);
5. for set \( S_r \) in descending order by \( \{C_r(X_r, X)\}_{i=1,i \neq r}^{N} \)
6. Compute the mean feature matrix \( F = \sum_{i=1,i \neq r}^{N} F'_i / (N-1) \);
7. Update \( X_{rk} \) by computing the correspondence between \( F_r \) and \( F'_k \) via the Hungarian method;
8. end for

![Fig. 3](image3.png)

**Fig. 3.** Average accuracy and clustering error curves as a function of the iteration number by matchCluster and matchCluster<sup>fast</sup> on the CMU Hotel sequence test by 20 tests with 28 feature sets and 20 features per each trial.
be sensitive to the quality of the initial point. Moreover, the updating order may also influence the iteration path.

There is one useful observation in [24], [31] that matching accuracy, though not observable in testing stage, is correlated with the pairwise consistency \( \{ C_p(X_i, X_j) \}^N_{i,j=1, \neq r} \) as defined in Def. (1). Moreover, the most consistent feature set by Def. (1) is expected to generate the most accurate initial matching set \( \{ X_{ri} \}^N_{r=1, \neq i} \) from \( X \). Thus we adopt this strategy and describe the improved algorithm in Alg. (2) while the expense is the additional time cost for computing the whole \( X \) with \( O(N^2) \) times of pairwise Hungarian method and the computing of \( \{ C_{u}(k, X) \}^N_{k=1} \) and \( \{ C_{p}(X_i, X_r) \}^N_{i,r=1, \neq r} \). One illustration of the usefulness of the above strategy is plotted in Fig. [3] in comparison with a random selection (matchCluster\textsuperscript{fast}).

IV. Experiment

A. Test data settings and compared methods

1) Synthetic feature set matching: The random graph test provides a controlled setting to evaluate the performance of our methods and other peer methods. For each trial, a reference feature set with \( n_o \) nodes is created by assigning a random attribute vector of dimension \( D \) to each of its nodes. In our test, we set \( D = 5 \) for all synthetic tests. The attributes are uniformly sampled from the interval \([0, 1]\). Then the ‘perturbed’ feature sets are created by adding a Gaussian deformation disturbance to the attributes \( \tilde{q}_{id} \), which is sampled from \( N(0, \varepsilon) \) i.e. \( \tilde{q}_{id} = q_{id} + N(0, \varepsilon) \) where the superscript ‘p’ and ‘r’ denotes for ‘perturb’ and ‘reference’ respectively. Optionally, each ‘perturbed’ feature set is added by \( n_o \) outliers sampled from the same distribution as inliers.

2) CMU hotel sequence: The CMU motion sequences (http://vasc.ri.cmu.edu/idb/html/motion/) are widely used for feature matching [19] and graph matching [22]–[24], [31] etc. Here we use it to test feature matching solvers. In our experiment, we focus on the hotel sequence (111 frames) because matching the other sequence house by different methods always produces the perfect matching results. We use 15, 20, 25 feature points from each frame respectively, where the unary feature vector is computed by the SIFT descriptor [34].

3) Willow-ObjectClass: This dataset released in [5] is constructed using images from Caltech-256 and PASCAL VOC2007. Each object category contains different number of images: 109 Face, 50 Duck, 66 Wine bottle, 40 Motorbike, and 40 Car images. For each image, 10 feature points are manually labeled on the target object. We focus on the testing on the face category because the matching accuracy for other images is much lower (below 0.25) if only unary SIFT feature is considered. This is because the objects are from different angles, different textures and colors, as well as different poses.

4) Comparing methods: We compare several state-of-the-art general multi-view matching methods including matchOpt [23], [24], matchSync [8], Composition based Affinity Optimization (CAO) and its consistency-enforcing variant CAO [31], [33]. In addition, we also involve the baseline two-set matching by the Hungarian method. There are two cases for applying the Hungarian method: i) Hung\textsuperscript{pair} which performs the Hungarian method independently on each pair of the feature sets. This directly produces the whole matching configuration \( X \); ii) randomly select one feature set \( S_i \), and preform the Hungarian method independently on each of the other feature sets with it. Then the pairwise matchings are computed by \( X_{ij} = X_i X_j \). Note the former has a quadratic \( O(N^2) \) time cost in terms of the number of feature sets \( N \), and the latter is in linear with \( N \). For our methods, we also test two versions: i) matchCluster which decides the base feature set and alternating order by the consistency in definition Def. (1) and Def. (2) respectively; ii) the fast version matchCluster\textsuperscript{fast} that randomly sets the base feature set and alternating order. Note we do not include any point registration method such as [12], [14], [38] in our evaluation because we focus on the non-parametric node correspondences and impose no parametric transformation prior nor regularization for fair comparison.

B. Results and discussion

The matching accuracy and run-time on the synthetic random tests are plotted in Fig. [4] and Fig. [5]. The former concerns the performance as a function of different disturbances, and the latter concerns the performance by changing the number of feature sets. The matching accuracy and run-time on the CMU Hotel sequence and Willow-ObjectClass Face is illustrated in Fig. [6] and Fig. [7] respectively. Fig. [8] plots the visual matching illustration for the face and hotel image collections.

We make several observations based on the results:

1) The synthetic tests suggest our methods (in green) are competitive especially in the presence of many outliers;
2) In real image tests, matchOpt performs even better than ours, while ours perform competitively against other state-of-the-arts. No outlier is added in real image tests;
3) matchCluster outperforms its variant matchCluster\textsuperscript{fast} especially in real image tests. While in the synthetic test, the accuracy margin is much reduced;
4) Though being most fast, while Hung\textsuperscript{lin} always obtain the worst accuracy. This is reasonable as the information across feature sets are not fully explored by Hung\textsuperscript{lin}.

Our analysis focuses on the first three bullets as 4) is obvious: For 1) and 2), this perhaps suggests our method is more suited in two cases: i) when the deformation noises associated with the feature sets are evenly distributed around the template feature set — that is the setting of our synthetic test; ii) when there are many outliers. For 3), we think the reason is that the real images are more heterogeneous which results in initial pairwise matchings with varying qualities. While in our synthetic tests, the noises are evenly distributed on the disturbed feature sets, thus their pairwise matching quality are relatively homogenous. This fact reduces the improvement space for adaptively deciding the updating order.

V. Conclusion and Future work

We propose a novel method for matching feature sets with two advantages: i) the consistency of correspondences of the multiple feature sets is satisfied; ii) the global information of
all feature sets is jointly explored. We show that given both simulated feature sets and real image sequences, the proposed method performs competitively with state-of-the-arts.

Future work involves designing and evaluating cost functions such as the K-nearest neighbor distance from the cluster. We will also apply our method to matching graphs by approximately extracting node signature e.g. the distribution of the length of the shortest paths from the node to other nodes.

Acknowledgement This work is supported by STCSM (15JC1401700, 14XD1402100, 13511504501).

REFERENCES

[1] G. Scott and H. Longuet-Higgins, An algorithm for associating the features of two images, in Proc. the Royal Society of London. Series B: Biological Sciences, 1991.

[2] Z. Zeng, T.-H. Chan, K. Jia and D. Xu, Finding correspondence from multiple images via sparse and low-rank decomposition, in ECCV, 2012.

[3] J.-G. Yu, G.-S. Xia, A. Samal and J. Tian Globally Consistent Correspondence of Multiple Feature Sets Using Proximal Gauss-Seidel Relaxation, in Pattern Recognition, 2016.
[41] M. Cho, K. Alahari, and J. Ponce. Learning graphs to match. In ICCV, 2013.

[40] A. V. Aho, M. Ganapathi, and S. W. Tjiang Code generation using tree code generation using tree

[39] E. M. Luks Isomorphism of graphs of bounded valence can be tested in polynomial time, Journal of Computer and System Sciences

[38] A. Myronenko and X. Song, Point set registration: Coherent point drift, TPAMI, 2010.

[37] X. Shi and H. Ling and J. Xing and W. Hu, Multi-target Tracking by Rank-1 Tensor Approximation, In CVPR, 2013.

[36] A. Myronenko and X. Song, Point set registration: Coherent point drift, TPAMI, 2010.

[35] J. Yan, Y. Li, W. Liu, H. Zha, X. Yang, and S. Chu, Graduated consistency-regularized optimization for multi-graph matching, In ECCV, 2014.

[34] D. Lowe, Object recognition from local scale-invariant features, In ICCV, 1999.

[33] J. Yan, Y. Li, W. Liu, H. Zha, X. Yang, and S. Chu, Multi-graph matching via affinity optimization with graduated, consistency regularization. TPAMI, 2016.

[32] S. Gold and A. Rangarajan, A graduated assignment algorithm for graph matching and map inference, In ACCV, 2009.

[31] J. Yan, M. Cho, H. Zha, X. Yang, and S. Chu, Graduated consistency-regularized optimization for multi-graph matching, In ECCV, 2014.

[30] Y. Tian, J. Yan, H. Zhang, Y. Zhang, X. Yang, and H. Zha, On the convergence of graph matching: Graduated assignment revisited, In ECCV, 2012.

[29] C.-H. Lee and A. Joshi, On convex relaxation of graph matching and dynamic programming, In CVPR, 2015.

[28] S. Gold and A. Rangarajan, A new point matching algorithm for non-rigid registration, CVIU, 2003.

[27] Q. Ngoc, A. Gautier, and M. Hein, A flexible tensor block coordinate ascent scheme for hypergraph matching, In CVPR, 2015.

[26] M. Leordeanu, A. Zanfir, and C. Sminchisescu, Semi-supervised learning and optimization for hypergraph matching. In ICCV, 2011.

[25] M. Leordeanu, A. Zanfir, and C. Sminchisescu, Semi-supervised learning and optimization for hypergraph matching. In ICCV, 2011.

[24] J. Yan, J. Wang, H. Zha, and X. Yang, Consistency-driven alternating optimization for multigraph matching: A unified approach. TIP, 2015.

[23] J. Yan, Y. Tian, H. Zha, X. Yang, Y. Zhang and S. Chu, Joint optimization for consistent multiple graph matching. In ICCV, 2013.

[22] J. Yan, C. Zhang, H. Zha, W. Liu, X. Yang, and S. Chu, Discrete hyper-graph matching. In CVPR, 2015.

[21] E. M. Loiola and N. Abreu and P. O. Boaventura-Netto and P. Hahn and T. Queiroz, A survey for the quadratic assignment problem. In SODA, 1995.

[20] X. Zhou, M. Zhu, and K. Daniilidis, Multi-image matching via fast alternating minimization, In ICCV, 2015.

[19] J. Maciel and J.P. Costeira, A global solution to sparse correspondence problems, PAMI, 2003.

[18] M. Cho, S. Kwak, C. Schmid, and J. Ponce, Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In CVPR 2015.

[17] M. Leordeanu, A. Zanfir, and C. Sminchisescu. Semi-supervised learning and optimization for hypergraph matching. In ICCV, 2011.

[16] A. Wong and M. You, Entropy and distance of random graphs with application to structural pattern recognition, PAMI, 1985.

[15] Y. Chen and G. Medioni, Object modeling by registration of multiple range images, IVC, 1992.

[14] J. Yan, J. Wang, H. Zha, and X. Yang, Consistency-driven alternating optimization for multigraph matching: A unified approach. TIP, 2015.

[13] Y. Zhang, Iterative point matching for registration of free-form curves and surfaces, IVC, 1994.

[12] H. Chui and A. Rangarajan, A new point matching algorithm for non-rigid registration, CVIU, 2003.

[11] M. Leordeanu, R. Sukthankar, and M. Hebert. Unsupervised learning and optimization for hypergraph matching. In TPAMI, 2013.

[10] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, IVC, 1994.

[9] Y . Chen and G. Medioni, Object modeling by registration of multiple range images, IVC, 1992.

[8] D. Pachauri, R. Kondor, and S. Vikas. Solving the multi-way matching problem by permutation synchronization. In NIPS, 2013.

[7] D. Eppstein, Subgraph isomorphism in planar graphs and related problems, in SODA, 1995.

[6] P. Foggia, G. Percannella, and M. Vento. Graph matching and learning for each pair of feature sets independently, and our method matchCluster. Red (green) lines indicate wrong (correct) correspondences.

[5] M. Neuhaus and H. Bunke. Self-organizing maps for learning the underlying cost function, IEEE Transactions on PAMI, 2003.

[4] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. IJPRAI, 2004.

[3] E. M. Luks Isomorphism of graphs of bounded valence can be tested in polynomial time, Journal of Computer and System Sciences

[2] J. Munkres, Algorithms for the assignment and transportation problems, Journal of the Society for Industrial & Applied Mathematics, vol. 5, no. 1, pp. 32–38, 1957.

[1] D. Lowe, Object recognition from local scale-invariant features, In ICCV, 1999.
[42] M. Neuhaus and H. Bunke Automatic learning of cost functions for graph edit distance, *Information Sciences*, 177(1):239–247, 2007.

[43] Z. Niu and R. Martin and M. Sabin and F. Langbein and H. Bucklow Applying database optimization technologies to feature recognition in CAD, *Computer-aided design and applications*, 12(3):373–382, 2015.

[44] Z. Niu and R. Martin and M. Sabin and F. Langbein and M. Sabin Rapidly finding CAD features using database optimization, *Computer-aided design and applications*, 69:35–50, 2015.

[45] Y. Yan and E. Ricci and R. Subramanian and G. Liu and O. Lanz and N. Sebe LA Multi-task Learning Framework for Head Pose Estimation under Target Motion, *PAMI*, 2016.