

ROML: A Robust Feature Correspondence Approach for Matching Objects in A Set of Images

Kui Jia¹ · Tsung-Han Chan² · Zinan Zeng³ · Shenghua Gao⁴ · Gang Wang⁵ · Tianzhu Zhang⁶ · Yi Ma⁴

Received: 21 March 2014 / Accepted: 8 September 2015
© Springer Science+Business Media New York 2015

Abstract Feature-based object matching is a fundamental problem for many applications in computer vision, such as object recognition, 3D reconstruction, tracking, and motion segmentation. In this work, we consider simultaneously matching object instances in a set of images, where both inlier and outlier features are extracted. The task is to identify the inlier features and establish their *consistent* correspondences across the image set. This is a challenging combinatorial problem, and the problem complexity grows exponentially

with the image number. To this end, we propose a novel framework, termed Robust Object Matching using Low-rank constraint (ROML), to address this problem. ROML optimizes simultaneously a partial permutation matrix (PPM) for each image, and feature correspondences are established by the obtained PPMs. Two of our key contributions are summarized as follows. (1) We formulate the problem as rank and sparsity minimization for PPM optimization, and treat simultaneous optimization of multiple PPMs as a regularized consensus problem in the context of distributed optimization. (2) We use the alternating direction method of multipliers method to solve the thus formulated ROML problem, in which a subproblem associated with a single PPM optimization appears to be a difficult integer quadratic program (IQP). We prove that under widely applicable conditions, this IQP is equivalent to a linear sum assignment problem, which can be efficiently solved to an exact solution. Extensive experiments on rigid/non-rigid object matching, matching instances of a common object category, and common object localization show the efficacy of our proposed method.

Communicated by Jiri Matas.

✉ Kui Jia
kuijia@umac.mo

Tsung-Han Chan
thchan@ieee.org

Zinan Zeng
zeng.zinan@gmail.com

Shenghua Gao
gaoshh@shanghaitech.edu.cn

Gang Wang
wanggang@ntu.edu.sg

Tianzhu Zhang
tzzhang@nlpr.ia.ac.cn

Yi Ma
mayi@shanghaitech.edu.cn

¹ Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, E11 Avenida da Universidade, Taipa, Macau SAR, China

² MediaTek Inc., No. 1, Dusing 1st Rd., Hsinchu Science Park, Hsinchu City 30078, Taiwan

³ Advanced Digital Sciences Center, 1 Fusionopolis Way, Singapore, Singapore

⁴ School of Information Science and Technology, ShanghaiTech University, No 8 building, 319 Yueyang Road, Xuhui District, Shanghai 200031, China

Keywords Object matching · Feature correspondence · Low-rank · Sparsity

1 Introduction

Object matching is a fundamental problem in computer vision. Given a pair or a set of images that contain com-

⁵ School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

⁶ Institute of Automation, Chinese Academy of Sciences, 95, Zhongguancun East Road, Haidian District, Beijing 100190, China

mon object instances, or an object captured under varying poses, it involves establishing correspondences between the parts or features of the objects contained in the images. Accurate, robust, and consistent matching across images is a key ingredient in a wide range of applications such as object recognition, shape matching, 3D reconstruction, tracking, and motion segmentation.

For a pair of feature sets extracted from two images, finding inliers from them and establishing correspondences are in general a combinatorial search problem. Objects may appear in images with cluttered background, and some parts of the objects may also be occluded. The search space can further explode when a *globally consistent matching* across a set of images is desired. For object instances with large intra-category variations or those captured under varying poses (e.g., non-rigid objects with articulated pose changes), the matching tasks become even more difficult. All these factors make object matching a very challenging task.

In literature, a variety of strategies have been proposed for object matching. In particular, early shape matching works use point sets to represent object patterns (Scott and Longuet-Higgins 1991; Shapiro 1992). To match between a pair of point sets, they build point descriptions by modeling spatial relations of points within each point set as higher level geometric structures, e.g., lines, curves, and surfaces, or more advanced features, e.g., shape context (Belongie et al. 2002). In Besl and McKay (1992), Chui and Rangarajan (2003), Belongie et al. (2002), alternating estimation of point correspondence and geometric transformation is also used for non-rigid shape matching. In general, point set based shape matching is less robust to measurement noise and outliers, with classical techniques such as RANSAC (Fischler and Bolles 1981) available to improve its robustness. The development of local invariant features (Lowe 1999; Mikolajczyk and Schmid 2004) for discriminative description of visual appearance has brought significant progress in object matching and recognition (Serre et al. 2005). For example in Russell et al. (2006), Deselaers et al. (2012), instances of a common object category from an image collection can be located and matched by exploiting the discriminative power of local feature descriptors. The popular Bag-of-Words model for object recognition is also built on matching (clustering) similar local region descriptors. However, local descriptors alone can be ambiguous for matching when there exist repetitive textures or less discriminative local appearance in images. In between of these two extremes, recent graph matching methods (Leordeanu and Hebert 2005; Torresani et al. 2008; Zhou and De la Torre 2012) consider both feature similarity and geometric compatibility between two sets of features, where the nodes of graphs correspond to local features and edges encode spatial relations between them. Mathematically, graph matching is formulated as a quadratic assignment problem (QAP), which is known to be NP-hard. Although

intensive efforts of these methods have been focusing on devising more accurate and efficient algorithms to solve this problem, in general, they can only obtain approximate solutions for QAP, and thus suboptimal correspondences for robust object matching.

Most of these methods focus on establishing correspondences between a pair of images. However, in practice, it is very common that when such a pair of images are available, a set of images are also available that we know a common object is present in them, such as a video sequence with a moving object, or a set of images collected from the Internet that contain instances of a generic object category. In these situations, it is desired that a globally consistent matching can be established. This is a very challenging combinatorial problem. As the number of images increases, the problem complexity explodes exponentially. A straightforward approach is to locally build correspondences between pairs of images. Obviously, pair-wise matching can only get suboptimal solutions, since matching found between pairs of images may not be globally consistent across the whole set. Compared to global matching, pair-wise matching is also less robust to outliers and occlusion of inlier features, as it cannot leverage additional constraints from other images that also contain the same object pattern of interest. In this work, we are thus interested in the following object matching problem.

Problem 1 Given a set of images with both inlier and outlier features extracted from each image, simultaneously identify a given number of inlier features from each image and establish their consistent correspondences across the image set.

In Problem 1, we consider the common scenario in object matching that there is exactly one object instance in each image. The inlier features describe appearance of the salient local regions of the object, and the rest of the features are outliers. One may think of these features as local region descriptors such as SIFT (Lowe 1999) or HOG (Dalal and Triggs 2005), although other types of features can also be used, which will be deliberated in later sections. Under this setting, the object instance contained in each image is naturally represented as a set of inlier features. Without consideration of intra-category variations, the corresponding inlier features extracted from different images, which characterize the same salient local regions of different instances of the object category, would be linearly correlated. When concatenating ordered inlier features in each image as a simple feature of the object instance (thinking of concatenating several SIFT feature vectors as a single vector), and then arraying these features of different images as the columns of a large matrix, this matrix will have low-rank, and ideally rank one. In situations where intra-category variations exist, e.g., variations in inlier features of different images caused by illumination or pose changes, the matrix low-rank property can still hold by decomposing out some errors.

Motivated by these observations, we propose in this paper a novel and principled framework, termed *Robust Object Matching using Low-rank constraint* (ROML), for identifying and matching inlier features of object instances across a set of images. ROML leverages the aforementioned low-rank property, via minimizing rank of a matrix and sparsity of a matrix (for decomposing out sparse errors), to simultaneously optimize a partial permutation matrix (PPM) for each image, and feature correspondences are established by the obtained PPMs (cf. (1) for the definition of PPM). The so formulated ROML problem concerns with simultaneous optimization of multiple PPMs, which belongs to a more general class of multi-index assignment problem (MiAP) and is proven to be NP-hard (Burkard et al. 2009). Exact solution methods are prohibitively slow for practical use. In this work, we treat simultaneous optimization of multiple PPMs involved in ROML as a regularized consensus problem in the context of distributed optimization (Bertsekas and Tsitsiklis 1989). We use the Alternating Direction Method of Multipliers (ADMM) (Boyd et al. 2011) to solve the ROML problem, in which a subproblem associated with a single PPM optimization appears to be a difficult integer quadratic program (IQP). We prove that under widely applicable conditions, this IQP is equivalent to a linear sum assignment problem (LSAP) (Burkard et al. 2009), which can be efficiently solved to an exact solution using the Hungarian algorithm (Kuhn 1955). Extensive experiments on rigid/non-rigid object matching, matching instances of a common object category, and common object localization show the efficacy of our proposed method. A MATLAB implementation of our method and the data used in the experiments can be found from our project website: <https://sites.google.com/site/kuijia/research/roml>.

A preliminary work of this paper has appeared in Zeng et al. (2012). In the present paper, we have made significant improvement over (Zeng et al. 2012) in the following aspects. In addition, we have also completely rewritten the paper to present our ideas more clearly.

- Although (Zeng et al. 2012) proposes to optimize a set of PPMs via rank and sparsity minimization for robust feature matching, however, its solution of each PPM optimization is obtained by sequentially solving two costly subproblems: a quadratic program over the continuous-domain relaxation of PPM, followed by a binary integer programming that projects the relaxed PPM into its feasible set. In fact, the second subproblem is irrelevant to the original objective function, and consequently, the thus obtained PPM is only suboptimal. In contrast, we propose in the present paper a new method to solve the PPM optimization and prove that under widely applicable conditions, the PPM optimization step is equivalent to an LSAP, which can be efficiently solved to an exact solution using the Hungarian algorithm. Extensive experiments in

Sect. 5 show the great advantage of the proposed ROML over the method in Zeng et al. (2012) in terms of both matching accuracy and efficiency.

- We present mathematical analysis in this paper to show that the proposed ROML formulation belongs to the NP-hard MiAP. We also discuss the suitability of ADMM for approximately solving ROML from the perspective of distributed optimization. These analysis and discussion put ROML in a broader context, which are overlooked in Zeng et al. (2012).

2 Related Works

There is an intensive literature on object/shape matching between a pair of images (Conte et al. 2004). Representative works include shape context (Belongie et al. 2002), graph (Leordeanu and Hebert 2005; Torresani et al. 2008) and hyper-graph (Lee et al. 2011; Zass and Shashua 2008; Duchenne et al. 2011) matching. In this section, we briefly review several existing methods that use multiple images/point sets for object matching, and also the more general MiAP.

Maciel and Costeira (2003) first proposed to use PPM to model both feature matching and outlier rejection in a set of images. They formulated optimization of PPMs as an integer constrained minimization problem. To solve this combinatorial problem, they relaxed both the objective function and integer constraints, resulting in an equivalent concave minimization problem. However, the complexity of concave minimization is still non-polynomial. Moreover, matching criteria used in the cost function of Maciel and Costeira (2003) were locally based on pair-wise similarity of features in different images. Instead, our method is based on low-rank and sparse minimization (via convex surrogate functions), whose problem size is polynomial w.r.t. the numbers of features and images, and whose cost function is also globally defined over features in all the images.

Rank constraints have been used in Oliveira et al. (2005), Oliveira et al. (2006) for point matching across video frames. They constructed a measurement matrix containing image coordinates of points extracted from a moving rigid object. Motivated by factorization model in shape-from-motion (Tomasi and Kanade 1992), they assumed this measurement matrix was low-rank, and used rank constraints to optimize PPMs for establishing point correspondences across frames. The method in Oliveira et al. (2005) is limited in several aspects: (1) an initial template of point set without outliers is assumed given; (2) every inlier point is required to be visible in all frames; (3) matching across frames is a bootstrapping process—points in a subsequent frame are to be aligned to those of previously matched frames, thus matching errors will inevitably propagate and accumulate; (4) an

initial rough estimate of point correspondences for a new frame is assumed given in their algorithm, which may be only valid for slow motion objects. The aspects (1) and (3) have to some extent been alleviated in Oliveira et al. (2006), but Oliveira et al. (2006) cannot cope with the other limiting aspects. As a globally consistent and robust matching framework, our method has no such limitations. More importantly, we note that the mechanism of rank constraints used in Oliveira et al. (2005, 2006) is different from that of our method. Methods (Oliveira et al. 2005, 2006) can only apply to matching of rigid objects using image coordinates as features, while our method considers low-rank assumption on a type of generally defined features, which take image coordinates and region descriptors as instances. Consequently, our method is able to apply in more general scenarios, such as matching of objects with non-rigid deformation.

Recently, a low-dimensional embedding method was proposed in Torki and Elgammal (2010) for feature matching. Given feature points extracted from each of a set of images, it can learn an embedded feature space, which encodes information of both region descriptors and the geometric structure of points in each image. Torki and Elgammal (2010) used k-means clustering in the embedded space for feature matching. As we will show in Sect. 5, k-means based on Euclidean distances of embedded features is not a good way to establish correspondences. There is no explicit outlier rejection mechanism in Torki and Elgammal (2010) either. Compared to Torki and Elgammal (2010), our method uses the low-rank and sparse constraints to optimize PPMs, which integrates correspondence and outlier rejection in a single step.

As mentioned in Sect. 1, our ROML formulation for multi-image object matching belongs to a more general class of MiAP for data association (Burkard et al. 2009), with other vision applications such as multi-target tracking (Collins 2012). MiAP is proven to be NP-hard, and only implicit enumeration methods such as branch-and-bound are known to give an exact solution, which are however prohibitively slow for practical use. Classical approximate solution methods include greedy, Greedy Randomized Adaptive Search Procedure (GRASP) (Robertson 2001), and relaxation based methods (Poore and Robertson 1997).

Greedy approaches build a matching that has the lowest cost at each iteration, which has the obvious weakness that decisions once made, are fixed and may later be shown to be suboptimal. GRASP improves over greedy approaches by progressively constructing a list of best candidate matches and randomly selecting one from them. The process is repeated until all matches are exhausted. A final local search over the neighborhood of obtained matches may be used to further optimize the solution. In Poore and Robertson (1997), Poore and Robertson presented a Lagrangian relaxation method that progressively relaxes the original and intermediate recovery MiAPs to linear assignment problems,

by incorporating constraints of each MiAP into its objective function via the Lagrangian. However, this method involves implicitly enumerative procedure, and is difficult to implement and analyze.

Collins (2012) recently proposed an iterated conditional modes (ICM) like method for video based multi-target tracking. His method is based on factoring the global decision variable for each target trajectory into a product of local variables defined for a target matching between each pair of adjacent frames. It then pair-wisely builds target matchings between adjacent frames by optimizing the corresponding local variables, but using a global cost function as matching criteria. However, the cost function in Collins (2012) is defined by enumerating for every possible target trajectory a constant-velocity motion energy, and the number of candidate trajectories grows exponentially with the number of frames. Both factors make it less applicable to the feature-based object matching problem considered in this paper. Nevertheless, our ROML formulation bears some spiritually similar idea with Collins (2012), in the sense that we also factor the global decision variables for multi-image feature matching into separate components. The difference is that we factor these global variables as a set of PPMs, each of which is to be optimized to identify inlier features from an image and re-arrange them in a proper order. We then treat the joint optimization of these PPMs as a regularized consensus problem in the context of distributed optimization (Bertsekas and Tsitsiklis 1989), and solve it using a ADMM-based method (Boyd et al. 2011). By this means the original NP-hard combinatorial problem boils down as to iteratively solve a set of independent pair-wise matching problems, which turn out to be easily solved.

In the preparation of this paper, we notice that a related method called permutation synchronization (Pachauri et al. 2013) was recently proposed, which also addresses the MiAP by optimizing a permutation matrix for each feature set. However, Pachauri et al. (2013) assumes initial matchings between each pair of the feature sets be available, and can only apply in the scenarios where there exist no outliers in each feature set, which make it less useful in the considered problem of feature-based object matching across a set of images.

3 Robust Object Matching Using Low-Rank and Sparse Constraints

Given a set of K images, we present in this section our problem formulation and algorithm for robust object matching. We consider the settings as stated in Problem 1. Assume n_k features $\{\mathbf{f}_i^k\}_{i=1}^{n_k}$ be extracted from the k th image, where the feature vector $\mathbf{f}_i^k \in \mathbb{R}^d$ can be either image coordinates of the feature point, or region descriptors such as SIFT (Lowe

1999) that characterize the local appearance. It can also be a combination of them by low-dimensional embedding (Torki and Elgammal 2010). In spite of these multiple choices, for now we generally refer to them as *features*. Discussion of different feature types and their applicable spectrums will be presented in Sect. 4. These n_k features are categorized as inliers or outliers. We assume at this moment that there are n inliers in each of the K images, where $n \leq n_k$ for $k = 1, \dots, K$. We will discuss the case of missing inliers shortly. In such a setting every k th image is represented as a set of n_k features, and the contained object instance is represented as the n inlier features.

3.1 Problem Formulation

Note that for inlier features in the K images, it is the feature similarity and geometric compatibility that determine they form an object *pattern* and this pattern repeats across the set of images. While similar outlier features may appear in multiple images, they just accidentally do so in a random, unstructured way. Our formulation for object matching is essentially motivated by these observations. Concretely, denote $\bar{\mathbf{F}}^k = [\mathbf{f}_1^k, \dots, \mathbf{f}_n^k] \in \mathbb{R}^{d \times n}$ as the matrix formed by inlier feature vectors in the k th image, so defined are the matrices $\{\bar{\mathbf{F}}^1, \dots, \bar{\mathbf{F}}^K\}$ for all the K images. Assume column vectors in each of these matrices are arrayed in the same order, i.e., inlier features in $\{\bar{\mathbf{F}}^1, \dots, \bar{\mathbf{F}}^K\}$ are respectively corresponded, then the matrix formed by $\bar{\mathbf{D}} = [\text{vec}(\bar{\mathbf{F}}^1), \dots, \text{vec}(\bar{\mathbf{F}}^K)] \in \mathbb{R}^{dn \times K}$ will be approximately low-rank, ideally rank one, where $\text{vec}(\cdot)$ is an operator that vectorizes a matrix by concatenating its column vectors.

Now consider the general case that there are outliers. Denote $\mathbf{F}^k = [\mathbf{f}_1^k, \dots, \mathbf{f}_{n_k}^k] \in \mathbb{R}^{d \times n_k}$ as the matrix having all n_k features of the k th image as its columns, where feature vectors are placed in a random order. The matrices $\{\mathbf{F}^1, \dots, \mathbf{F}^K\}$ for all K images are similarly defined. As aforementioned our interest for object matching is to identify the n inlier feature vectors from each matrix of $\{\mathbf{F}^1, \dots, \mathbf{F}^K\}$, and establish correspondences among them. For any k th image, this can be realized by the *partial permutation matrix* (PPM) defined by

$$\mathcal{P}^k = \{\mathbf{P}^k \in \mathbb{R}^{n_k \times n} \mid p_{i,j}^k \in \{0, 1\}, \sum_i p_{i,j}^k = 1 \forall j = 1, \dots, n, \sum_j p_{i,j}^k \leq 1 \forall i = 1, \dots, n_k\}, \quad (1)$$

where $p_{i,j}^k$ denotes an entry of the PPM \mathbf{P}^k at the i th row and j th column. Thus, there exist PPMs $\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K$ such that inlier feature vectors are selected and corresponded in $\{\mathbf{F}^k \mathbf{P}^k \in \mathbb{R}^{d \times n}\}_{k=1}^K$, i.e., the matrix $[\text{vec}(\mathbf{F}^1 \mathbf{P}^1), \dots, \text{vec}(\mathbf{F}^K \mathbf{P}^K)] \in \mathbb{R}^{dn \times K}$ is rank deficient. In the following of this paper, we also use

$$\mathbf{D}(\{\mathbf{P}^k\}_{k=1}^K) = [\text{vec}(\mathbf{F}^1 \mathbf{P}^1), \dots, \text{vec}(\mathbf{F}^K \mathbf{P}^K)], \quad (2)$$

to simplify writings of equations. In the context where the values of $\{\mathbf{P}^k\}_{k=1}^K$ are determined, we also write $\mathbf{D} = [\text{vec}(\mathbf{F}^1 \mathbf{P}^1), \dots, \text{vec}(\mathbf{F}^K \mathbf{P}^K)]$. Based on this low-rank assumption, feature correspondence can be formulated as the following problem to optimize $\{\mathbf{P}^k\}_{k=1}^K$

$$\min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K} \text{rank}(\mathbf{D}(\{\mathbf{P}^k\}_{k=1}^K)),$$

where $\text{rank}(\cdot)$ is a function to measure matrix rank. By introducing an auxiliary variable \mathbf{L} (to facilitate the development of a solving algorithm), the above problem can also be written as the following equivalent problem

$$\begin{aligned} & \min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K, \mathbf{L}} \text{rank}(\mathbf{L}) \\ \text{s.t. } & \mathbf{D}(\{\mathbf{P}^k\}_{k=1}^K) = \mathbf{L}. \end{aligned} \quad (3)$$

In practice, however, some inlier features characterizing the same local appearance information of object instances in different images could vary due to illumination change, object pose change, or other intra-category object variations. Some inlier features could also be missing due to partial occlusion of object instances. Thus the low-rank assumption used in (3) cannot be fully satisfied. To improve the robustness, we introduce a sparse error term into (3) to model all these contaminations, and modify the formulation (3) as

$$\begin{aligned} & \min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K, \mathbf{L}, \mathbf{E}} \text{rank}(\mathbf{L}) + \lambda \|\mathbf{E}\|_0 \\ \text{s.t. } & \mathbf{D}(\{\mathbf{P}^k\}_{k=1}^K) = \mathbf{L} + \mathbf{E}, \end{aligned} \quad (4)$$

where $\|\cdot\|_0$ is ℓ_0 -norm counting the number of nonzero entries, and $\lambda > 0$ is a parameter controlling the trade-off between rank of \mathbf{L} and sparsity of \mathbf{E} .

3.2 The Algorithm

The optimization problem (4) is not directly tractable due to the following aspects: (1) both $\text{rank}(\cdot)$ and $\|\cdot\|_0$ are non-convex, discrete-valued functions, minimization of which is NP-hard; (2) entries of $\{\mathbf{P}^k\}_{k=1}^K$ are constrained to be binary, resulting in a difficult nonlinear integer programming problem. To make it tractable, we first consider the recent convention of replacing $\text{rank}(\cdot)$ and $\|\cdot\|_0$ with their convex surrogates $\|\cdot\|_*$ and $\|\cdot\|_1$ respectively (Candès et al. 2011), where $\|\cdot\|_*$ denotes nuclear norm (sum of the singular values) and $\|\cdot\|_1$ is ℓ_1 -norm. Applying the same relaxation to (4) yields

$$\begin{aligned}
 & \min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K, \mathbf{L}, \mathbf{E}} \|\mathbf{L}\|_* + \lambda \|\mathbf{E}\|_1 \\
 & \text{s.t. } \mathbf{D}(\{\mathbf{P}^k\}_{k=1}^K) = \mathbf{L} + \mathbf{E}, \\
 & \mathcal{P}^k = \{\mathbf{P}^k \in \{0, 1\}^{n_k \times n} \mid \mathbf{1}_{n_k}^\top \mathbf{P}^k = \mathbf{1}_n^\top, \\
 & \mathbf{P}^k \mathbf{1}_n \leq \mathbf{1}_{n_k}\}, \forall k = 1, \dots, K,
 \end{aligned} \tag{5}$$

where we have written the constraints of $\{\mathcal{P}^k\}_{k=1}^K$ in matrix form, and $\mathbf{1}_{n_k}$ (or $\mathbf{1}_n$) denotes a column vector of length n_k (or n) with all entry values of 1. We refer to the problem (5) as our framework of *Robust Object Matching using Low-rank (and sparse) constraints* (ROML). In this paper, we set the parameter $\lambda = 5/\sqrt{dn}$ (except experiments in Sect. 5.1), which is suggested by the closely related work of Robust Principal Component Analysis (RPCA) (Candès et al. 2011).¹

The problem (5) involves jointly optimizing a set of K PPMs. As reviewed in Sect. 2, it is an instance of MiAP and proved to be NP-hard, for which approximate solution methods are practically used. To solve (5), note that it is a formulation of regularized consensus problem, where the local variables $\text{vec}(\mathbf{F}^k \mathbf{P}^k)$ (function of \mathbf{P}^k), $k = 1, \dots, K$, in $\mathbf{D}(\{\mathbf{P}^k\}_{k=1}^K)$ are constrained to be equal to components (column vectors) of the global variable $\mathbf{L} + \mathbf{E}$, which is further regularized in the objective function. In literature, consensus problems are popularly solved using ADMM method in the context of distributed optimization (Boyd et al. 2011; Bertsekas 1999; Bertsekas and Tsitsiklis 1989). The general ADMM method decomposes a global problem into local subproblems that can be readily solved. For consensus problems such as (5), ADMM decomposes optimization of \mathbf{L} , \mathbf{E} , and $\{\mathbf{P}^k\}_{k=1}^K$ into subproblems that update \mathbf{L} , \mathbf{E} , and each of $\{\mathbf{P}^k\}_{k=1}^K$ respectively. Thus joint optimization over $\{\mathbf{P}^k\}_{k=1}^K$ boils down as independent optimization of individual \mathbf{P}^k , $k = 1, \dots, K$, in each ADMM iteration. However, the subproblem to update each \mathbf{P}^k concerns with nonlinear integer programming. It is essential to understand the convergence property of ADMM under this condition, which we will discuss in Sect. 3.2.2 after presentation of our algorithmic procedure.

¹ Suppose we have a data matrix $\mathbf{D} \in \mathbb{R}^{m_1 \times m_2}$, which is formed by superposition of a low-rank matrix \mathbf{L} and a sparse matrix \mathbf{E} , i.e., $\mathbf{D} = \mathbf{L} + \mathbf{E}$. Assume the low-rank matrix \mathbf{L} is not sparse, and the sparse matrix \mathbf{E} is not low-rank (e.g., the support pattern of \mathbf{E} may be selected uniformly at random). RPCA proves that the matrices \mathbf{L} and \mathbf{E} can be recovered *exactly* via a convex program called Principal Component Pursuit: $\min_{\mathbf{L}_{rpca}, \mathbf{E}_{rpca}} \|\mathbf{L}_{rpca}\|_* + \lambda_{rpca} \|\mathbf{E}_{rpca}\|_1$ s.t. $\mathbf{D} = \mathbf{L}_{rpca} + \mathbf{E}_{rpca}$, provided that the rank of \mathbf{L} is not too large, and that \mathbf{E} is reasonably sparse. Under these assumptions, a *universal* choice of the parameter $\lambda_{rpca} = 1/\sqrt{\max(m_1, m_2)}$ is identified in the theoretical analysis of RPCA. In practical data, assumptions used in the theoretical proof of RPCA are not generally satisfied. RPCA suggests setting $\lambda_{rpca} = C/\sqrt{\max(m_1, m_2)}$, where C is a constant which can be adjusted properly to improve performance on practical data.

We first write the augmented Lagrangian of (5) as

$$\begin{aligned}
 \mathcal{L}_\rho(\mathbf{L}, \mathbf{E}, \{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K, \mathbf{Y}) &= \|\mathbf{L}\|_* + \lambda \|\mathbf{E}\|_1 \\
 &+ \frac{\rho}{2} \|\mathbf{L} + \mathbf{E} - \mathbf{D}(\{\mathbf{P}^k\}_{k=1}^K)\|_F^2 + \frac{1}{\rho} \mathbf{Y}^\top \mathbf{Y},
 \end{aligned} \tag{6}$$

where $\mathbf{Y} \in \mathbb{R}^{dn \times K}$ is a matrix of Lagrange multipliers, ρ is a positive scalar, and $\|\cdot\|_F$ denotes the Frobenius norm. The ADMM algorithm iteratively estimates one of the matrices \mathbf{L} , \mathbf{E} , $\{\mathbf{P}^k\}_{k=1}^K$, and the Lagrange multiplier \mathbf{Y} by minimizing (6), while keeping the others fixed. More specifically, our ADMM procedure consists of the following iterations

$$\mathbf{L}_{t+1} = \arg \min_{\mathbf{L}} \mathcal{L}_\rho(\mathbf{L}, \mathbf{E}_t, \{\mathbf{P}^k\}_{k=1}^K, \mathbf{Y}_t), \tag{7}$$

$$\mathbf{E}_{t+1} = \arg \min_{\mathbf{E}} \mathcal{L}_\rho(\mathbf{L}_{t+1}, \mathbf{E}, \{\mathbf{P}^k\}_{k=1}^K, \mathbf{Y}_t), \tag{8}$$

$$\{\mathbf{P}^k\}_{k=1}^K = \arg \min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K} \mathcal{L}_\rho(\mathbf{L}_{t+1}, \mathbf{E}_{t+1}, \{\mathbf{P}^k\}_{k=1}^K, \mathbf{Y}_t), \tag{9}$$

$$\mathbf{Y}_{t+1} = \mathbf{Y}_t + \rho(\mathbf{L}_{t+1} + \mathbf{E}_{t+1} - \mathbf{D}_{t+1}), \tag{10}$$

where t denotes the iteration number and we compute $\mathbf{D}_{t+1} = [\text{vec}(\mathbf{F}^1 \mathbf{P}_{t+1}^1), \dots, \text{vec}(\mathbf{F}^K \mathbf{P}_{t+1}^K)]$ after step (9).

The problems (7) and (8) for updating the global variables \mathbf{L} and \mathbf{E} are both convex programs. They can be explicitly written as the forms of the proximal operators associated with a nuclear norm or an ℓ_1 -norm respectively (Lin et al. 2010). To spell out the solutions, define the soft-thresholding operator for scalars as $\mathcal{T}_\tau[x] = \text{sign}(x) \cdot \max\{|x| - \tau, 0\}$, with $\tau > 0$. When applied to vectors/matrices, it operates element-wisely. The optimal solution to (7) and (8) can be written as

$$\begin{aligned}
 (\mathbf{U}, \mathbf{S}, \mathbf{V}) &= \text{svd}(\mathbf{D}_t - \mathbf{E}_t - \frac{1}{\rho} \mathbf{Y}_t), \\
 \mathbf{L}_{t+1} &= \mathbf{U} \mathcal{T}_{\frac{1}{\rho}}[\mathbf{S}] \mathbf{V}^\top,
 \end{aligned} \tag{11}$$

$$\mathbf{E}_{t+1} = \mathcal{T}_{\frac{\lambda}{\rho}}[\mathbf{D}_t - \mathbf{L}_{t+1} - \frac{1}{\rho} \mathbf{Y}_t]. \tag{12}$$

Derivations of the above solution to problems (7) and (8) can be found in Appendix 2.

Optimization of (9) is more involved than those of (7) and (8), mostly because of the binary constraints enforced on the entries of $\{\mathbf{P}^k\}_{k=1}^K$. Given updated variables \mathbf{L}_{t+1} , \mathbf{E}_{t+1} , and \mathbf{Y}_t , we explicitly write the problem (9) as

$$\begin{aligned}
 & \min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K} \frac{\rho}{2} \|\mathbf{L}_{t+1} + \mathbf{E}_{t+1} + \frac{1}{\rho} \mathbf{Y}_t \\
 & - [\text{vec}(\mathbf{F}^1 \mathbf{P}^1), \dots, \text{vec}(\mathbf{F}^K \mathbf{P}^K)]\|_F^2.
 \end{aligned} \tag{13}$$

We observe that (13) can be decoupled into K independent subproblems, each of which concerns with optimization of one of the local variables $\{\mathbf{P}^k\}_{k=1}^K$. The k th subproblem to update \mathbf{P}^k is written as

$$\min_{\mathbf{p}^k \in \mathcal{P}^k} \frac{\rho}{2} \left\| (\mathbf{L}_{t+1} + \mathbf{E}_{t+1} + \frac{1}{\rho} \mathbf{Y}_t) \mathbf{e}_k - \text{vec}(\mathbf{F}^k \mathbf{P}^k) \right\|_2^2, \quad (14)$$

where \mathbf{e}_k denotes a unit column vector with all entries set to 0 except the k th one, which is set to 1. Denote $\theta^k = \text{vec}(\mathbf{P}^k) \in \mathbb{R}^{n n_k}$, $\mathbf{G}^k = \mathbf{I}_n \otimes \mathbf{F}^k \in \mathbb{R}^{d n \times n n_k}$, $\mathbf{J}^k = \mathbf{I}_n \otimes \mathbf{1}_{n_k}^\top \in \mathbb{R}^{n \times n n_k}$, $\mathbf{H}^k = \mathbf{1}_n^\top \otimes \mathbf{I}_{n_k} \in \mathbb{R}^{n n_k \times n n_k}$, \otimes is the Kronecker product, and \mathbf{I}_n (or \mathbf{I}_{n_k}) is the identity matrix of size $n \times n$ (or $n_k \times n_k$). Using the fact $\text{vec}(\mathbf{XYZ}) = (\mathbf{Z}^\top \otimes \mathbf{X}) \text{vec}(\mathbf{Y})$, we can rewrite (14) as the following equivalent problem to update θ^k

$$\min_{\theta^k} \frac{\rho}{2} \theta^{k \top} \mathbf{G}^{k \top} \mathbf{G}^k \theta^k - \mathbf{e}_k^\top [\mathbf{Y}_t^\top + \rho(\mathbf{L}_{t+1} + \mathbf{E}_{t+1})^\top] \mathbf{G}^k \theta^k$$

s.t. $\mathbf{J}^k \theta^k = \mathbf{1}_n$, $\mathbf{H}^k \theta^k \leq \mathbf{1}_{n_k}$, $\theta^k \in \{0, 1\}^{n n_k}$. (15)

(15) appears to be a difficult integer constrained quadratic program. To solve it, a common approach is to relax the constraint set of (15) into its convex hull, and then project back the attained continuous-domain results by either thresholding or more complicated methods, which, however, cannot guarantee to get the optimal solution (Leordeanu et al. 2009). For the ROML problem (5), we assume that distinctive information of each column vector in any \mathbf{F}^k of $\{\mathbf{F}^k\}_{k=1}^K$ is represented by the relative values of its elements, rather than their absolute magnitude. In other words, multiplying each feature vector by a scaling factor does not change the pattern of each feature. Based on this assumption, we prove that (15) is equivalent to a linear sum assignment problem (Burkard et al. 2009).

Theorem 1 *For the ROML problem (5), assuming distinctive information of each column vector in any \mathbf{F}^k of $\{\mathbf{F}^k\}_{k=1}^K$ is represented by the relative values of its elements, (15) is always equivalent to the following formulation of linear sum assignment problem*

$$\min_{\theta^k} -\mathbf{e}_k^\top [\mathbf{Y}_t^\top + \rho(\mathbf{L}_{t+1} + \mathbf{E}_{t+1})^\top] \mathbf{G}^k \theta^k$$

s.t. $\mathbf{J}^k \theta^k = \mathbf{1}_n$, $\mathbf{H}^k \theta^k \leq \mathbf{1}_{n_k}$, $\theta^k \in \{0, 1\}^{n n_k}$. (16)

Proof We prove the equivalence by showing that, under the considered assumption for the ROML problem (5), the objective function of (15) is equivalent to a linear function, as written in (16), which together with the constraints of (16), turns out to be a formulation of LSAP. Denote $\mathbf{p}_j^k \in \mathbb{R}^{n_k}$, $j = 1, \dots, n$, as columns of PPM \mathbf{P}^k . From the definitions of \mathbf{G}^k and θ^k , it is straightforward to show that

$$\mathbf{G}^k \theta^k = \text{vec}(\mathbf{F}^k \mathbf{P}^k) = [(\mathbf{F}^k \mathbf{p}_1^k)^\top, \dots, (\mathbf{F}^k \mathbf{p}_n^k)^\top]^\top \quad (17)$$

Since $\mathbf{F}^k = [\mathbf{f}_1^k, \dots, \mathbf{f}_{n_k}^k] \in \mathbb{R}^{d \times n_k}$, and from the constraints of \mathbf{P}^k (explicitly stated in (5)), it is clear that each subvector $\mathbf{F}^k \mathbf{p}_j^k$, $j = 1, \dots, n$, of (17) selects one column feature vector

from \mathbf{F}^k , with a unique index from the set $\{1, \dots, n_k\}$. From (17) we also have

$$\theta^{k \top} \mathbf{G}^{k \top} \mathbf{G}^k \theta^k = \|\mathbf{G}^k \theta^k\|_2^2 = \sum_{j=1}^n \|\mathbf{F}^k \mathbf{p}_j^k\|_2^2. \quad (18)$$

In case of $n_k = n$, i.e., there exist no outliers in the considered feature-based object matching, (18) is equal to a constant value no matter what feasible \mathbf{P}^k or θ^k is used. In the more general case of $n_k > n$, since information of each of the feature vectors \mathbf{f}_i^k , $i = 1, \dots, n_k$, is preserved by relative values of its elements, we can always normalize them so that they have an equal Euclidean norm, i.e., $\|\mathbf{f}_i^k\|_2 = \dots = \|\mathbf{f}_{n_k}^k\|_2 = c_k$. And (18) is again equal to a constant value no matter what feasible \mathbf{P}^k or θ^k is used. We thus finish the proof. □

The LSAP (16) can be exactly and efficiently solved using a rectangular-matrix variant of the Hungarian algorithm (Burkard et al. 2009). After solving K (16)-like problems for $k = 1, \dots, K$, we get the updates of $\{\theta_{t+1}^k\}_{k=1}^K$ and compute $\mathbf{D}_{t+1} = [\mathbf{G}^1 \theta_{t+1}^1, \dots, \mathbf{G}^K \theta_{t+1}^K]$. The Lagrange multiplier matrix \mathbf{Y}_{t+1} is then updated using (10). Our ADMM procedure iteratively performs the steps (7), (8), (9), and (10), until a specified stopping condition is satisfied. Normally, the primal and dual residuals can be used as the stopping criteria.² To improve the convergence, a common practice is to use a monotonically increasing sequence of $\{\rho_t\}$. We also adopt this strategy. The pseudocode of our algorithm is summarized in Algorithm 1.³

3.2.1 Discussion of Solving ROML Using ADMM

Solving the ROML formulation (5) establishes n sets of consistent feature correspondences across the given K images. In other words, it aims to find n “good” ones out of the total $\frac{1}{n!} \binom{n_k}{n_k-n}^K$ feasible solutions, assuming $n_1 = \dots = n_k = \dots = n_K$. As reviewed in Sect. 2, ROML belongs to the more general class of MiAP. To see how ROML relates to MiAP, we write the standard MiAP formulation (Burkard

² For the ROML problem (5), the primal residual is $\mathcal{R}_{pri}^{t+1} = \mathbf{L}^{t+1} + \mathbf{E}^{t+1} - \mathbf{D}^{t+1}$, and the dual residuals are $\mathcal{R}_{dual, \mathbf{L}}^{t+1} = \rho(\mathbf{E}^t + \mathbf{D}^t - \mathbf{E}^{t+1} - \mathbf{D}^{t+1})$ (w.r.t. the variable \mathbf{L}) and $\mathcal{R}_{dual, \mathbf{E}}^{t+1} = \rho(\mathbf{D}^t - \mathbf{D}^{t+1})$ (w.r.t. the variable \mathbf{E}).

³ We note that the obtained solution $\{\mathbf{P}^k\}_{k=1}^K$ by solving the ROML problem (5) using Algorithm 1 belongs to a group of equivalent solutions, since there is no constraint on the order of columns in any of $\{\mathbf{P}^k\}_{k=1}^K$. It is always easy to transform $\{\mathbf{P}^k\}_{k=1}^K$ to some canonical form, e.g., by permuting columns of each of $\{\mathbf{P}^k\}_{k=1}^K$ according to sorted image coordinates of feature points in any of the K images. Without loss of generality we assume the solution $\{\mathbf{P}^k\}_{k=1}^K$ given by Algorithm 1 has been transformed to some canonical form for ease of evaluation.

Algorithm 1: Solving ROML by ADMM

input : Feature vectors $\mathbf{F}^k = [\mathbf{f}_1^k, \dots, \mathbf{f}_{n_k}^k] \in \mathbb{R}^{d \times n_k}$ (normalized as $\|\mathbf{f}_1^k\|_2 = \dots = \|\mathbf{f}_{n_k}^k\|_2 = c_k$ when there exist outliers), $k = 1, \dots, K$, the number n of inliers, weight $\lambda > 0$, and initialization of $\{\mathbf{P}_0^k \in \mathcal{P}^k\}_{k=1}^K, \mathbf{L}_0 = 0, \mathbf{E}_0 = 0, \mathbf{Y}_0 = 0$, and $\rho_0 > 0$.

1 **while** *not converged* **do**
 2 $(\mathbf{U}, \mathbf{S}, \mathbf{V}) = \text{svd}(\mathbf{D}_t - \mathbf{E}_t - \frac{1}{\rho_t} \mathbf{Y}_t)$.
 3 $\mathbf{L}_{t+1} = \mathbf{U} \mathcal{T}_{\frac{\lambda}{\rho_t}}[\mathbf{S}] \mathbf{V}^\top$.
 4 $\mathbf{E}_{t+1} = \mathcal{T}_{\frac{\lambda}{\rho_t}}[\mathbf{D}_t - \mathbf{L}_{t+1} - \frac{1}{\rho_t} \mathbf{Y}_t]$.
 5 **for each** k **do**
 6 let $\theta_t^k = \text{vec}(\mathbf{P}_t^k), \mathbf{G}^k = \mathbf{I}_n \otimes \mathbf{F}^k, \mathbf{J}^k = \mathbf{I}_n \otimes \mathbf{1}_{n_k}^\top$,
 $\mathbf{H}^k = \mathbf{1}_{n_k}^\top \otimes \mathbf{I}_{n_k}$, solve the LSAP problem (16) to get the update θ_{t+1}^k .
 7 **end**
 8 $\mathbf{D}_{t+1} = [\mathbf{G}^1 \theta_{t+1}^1, \dots, \mathbf{G}^K \theta_{t+1}^K]$.
 9 $\mathbf{Y}_{t+1} = \mathbf{Y}_t + \rho_t (\mathbf{L}_{t+1} + \mathbf{E}_{t+1} - \mathbf{D}_{t+1})$.
 10 $\rho_{t+1} \leftarrow \rho_t$.
 11 $t \leftarrow t + 1$.
 12 **end**

output: solution $\{\mathbf{P}_t^k\}_{k=1}^K, \mathbf{L}_t$, and \mathbf{E}_t .

et al. 2009) for the considered feature-based object matching problem as

$$\begin{aligned}
 & \min_{\{z_{i_1, i_2, \dots, i_K}\}} \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_K=1}^{n_K} a_{i_1, i_2, \dots, i_K} z_{i_1, i_2, \dots, i_K} \\
 & \text{s.t.} \quad \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} \dots \sum_{i_K=1}^{n_K} z_{i_1, i_2, \dots, i_K} \leq 1, \quad i_1 = 1, \dots, n_1 \\
 & \quad \sum_{i_3=1}^{n_3} \sum_{i_4=1}^{n_4} \dots \sum_{i_K=1}^{n_K} z_{i_1, i_2, \dots, i_K} \leq 1, \quad i_2 = 1, \dots, n_2 \\
 & \quad \vdots \\
 & \quad \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} \dots \sum_{i_{K-1}=1}^{n_{K-1}} z_{i_1, i_2, \dots, i_K} \leq 1, \quad i_K = 1, \dots, n_K \\
 & \quad \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} \dots \sum_{i_K=1}^{n_K} z_{i_1, i_2, \dots, i_K} = n, \quad z_{i_1, i_2, \dots, i_K} \in \{0, 1\}, \quad (19)
 \end{aligned}$$

where $i_k \in \{1, \dots, n_k\}$ indexes the n_k feature vectors extracted from the k th image, $k = 1, \dots, K$. The global decision variable z_{i_1, i_2, \dots, i_K} is equal to 1 when the corresponding feature points are matched across the K images, with each feature from one of the K images, and a_{i_1, i_2, \dots, i_K} denotes the cost of this matching. By factoring/reformulating the set of global decision variables $\{z_{i_1, i_2, \dots, i_K}\}$ as PPMs $\{\mathbf{P}^k\}_{k=1}^K$ defined by (1), we get the following equivalent problems

$$\min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K} \sum_{j=1}^n \sum_{i_1=1}^{n_1} \dots \sum_{i_K=1}^{n_K} a_{i_1, \dots, i_K} \prod_{k=1}^K p_{i_k, j}^k, \quad (20)$$

$$\min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K} \sum_{j=1}^n \sum_{i_1=1}^{n_1} \dots \sum_{i_K=1}^{n_K} \|\mathbf{[f}_{i_1}^1, \dots, \mathbf{f}_{i_K}^K]\|_* \prod_{k=1}^K p_{i_k, j}^k, \quad (21)$$

$$\min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K} \sum_{j=1}^n \|\mathbf{[F}^1 \mathbf{p}_j^1, \dots, \mathbf{F}^K \mathbf{p}_j^K]\|_*, \quad (22)$$

where $j \in \{1, \dots, n\}$ indexes the n inlier matches, and $p_{i_k, j}^k$ and \mathbf{p}_j^k denote the (i_k, j) entry and j th column of \mathbf{P}^k respectively. In (21) and (22), we have used nuclear norm of the matrix formed by a candidate match of K feature vectors as the cost coefficient a_{i_1, \dots, i_K} . As an instance of MiAP, jointly optimizing the set of PPMs in the above equivalent problems is NP-hard. Approximate methods are thus important to get practically meaningful solutions. In fact, due to inevitable noise in cost coefficients, e.g., that generated by various variations of object instances in different images, it is often sufficient to find suboptimal solutions that are within the noise level of the optimal one.

To understand how we have developed an approximate method in preceding sections, we slightly modify (22) by vertically arraying the n matrices $\mathbf{[F}^1 \mathbf{p}_j^1, \dots, \mathbf{F}^K \mathbf{p}_j^K]$, $j = 1, \dots, n$, as a bigger matrix, resulting in the following problem to optimize $\{\mathbf{P}^k\}_{k=1}^K$

$$\min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K} \|\mathbf{[vec}(\mathbf{F}^1 \mathbf{P}^1), \dots, \text{vec}(\mathbf{F}^K \mathbf{P}^K)]\|_*, \quad (23)$$

which turns out to be equivalent to a nuclear norm relaxed version of (3). Indeed, by introducing the global variable \mathbf{L} in (3), and also the global variables \mathbf{L} and \mathbf{E} in (4) and (5) for a robust extension, we essentially formulate the multi-image object matching version of MiAP as a regularized consensus problem (Bertsekas and Tsitsiklis 1989), with $\|\mathbf{L}\|_* + \lambda \|\mathbf{E}\|_1$ as the regularization term. It becomes well suited to be solved using distributed optimization methods such as ADMM. As in the presented ADMM procedure (7)–(10), the “fusion” steps (7) and (8) collect information of the t th iteration $\{\mathbf{P}_t^k\}_{k=1}^K$ to update \mathbf{L} and \mathbf{E} , and the “broadcast” step (9) independently updates each \mathbf{P}^k of $\{\mathbf{P}^k\}_{k=1}^K$, using the updated fusion centers \mathbf{L}_{t+1} and \mathbf{E}_{t+1} . Our proposed ADMM method thus belongs to a strategy of “fusion-and-broadcast”, for ROML and the more general MiAP.

3.2.2 Convergence Analysis

The ADMM method is proven to converge to global optimum under some mild conditions for linearly constrained convex problem whose objective function is separable into two individual convex functions with non-overlapping variables (see Goldfarb et al. 2013 and references therein). In our case, the ROML problem (5) is nonconvex, due to the binary constraint associated with $\{\mathbf{P}^k\}_{k=1}^K$. The convergence

property of ADMM for nonconvex problems is still an open question in theory. However, it is not uncommon to see that ADMM has served a powerful heuristic for some nonconvex problems in practice (Zhang 2010; Shen et al. 2014). In the following, we present simulated experiments that demonstrate the excellent convergence property of ADMM for ROML.

Specifically, we generated synthetically $K = 30$ groups of vectors simulating extracted feature vectors from K images, with dimension of each vector \mathbf{f} as $d = 50$. There were both inlier and outlier feature vectors in each group. The inliers were produced by randomly generating d -dimensional vectors whose entries were drawn from i.i.d. normal distribution, and were shared in each of the K groups. The outliers were similarly produced by randomly generating d -dimensional vectors following i.i.d. normal distribution, but were independently generated for each group. We then added sparse errors of large magnitude to both inlier and outlier vectors. For each vector \mathbf{f} , the error values were uniformly drawn from the range $[-2 \max(\text{abs}(\mathbf{f})), 2 \max(\text{abs}(\mathbf{f}))]$. Finally, we normalized all vectors to constant ℓ_2 -norm to fit with our algorithmic settings. We fixed the number of inliers in each group as $n = 10$, and investigated the convergence and recovery properties of our algorithm under varying numbers of outliers and ratios of sparse errors. The number of outliers in each group was ranged in $[0, 40]$, and the ratio of sparse errors in each vector was ranged in $[0, 0.8]$. Denote the ground truth PPMs of any test setting as $\{\mathbf{P}^{k*}\}_{k=1}^K$, and the recovered PPMs as $\{\mathbf{P}^k\}_{k=1}^K$. The recovery rate is computed as $\sum_{k=1}^K \|\mathbf{P}^k \circ \mathbf{P}^{k*}\|_0 / \sum_{k=1}^K \|\mathbf{P}^{k*}\|_0$, where \circ is Hadamard product. For each setting of outlier number and sparse error ratio, we run 5 random trials and averaged the results. Figure 1b reports the recovery rates under different settings, which shows that our algorithm works perfectly in a

large range of outlier numbers and ratios of sparse errors. For one of them (the outlier number is 32 and sparse error ratio is 0.4), we plot in Fig. 1a its convergence curves of 5 random trials in terms of the primal residual ($\|\mathbf{L} + \mathbf{E} - \mathbf{D}\|_F$) and objective function ($\|\mathbf{L}\|_* + \lambda\|\mathbf{E}\|_1$). Convergence properties under other settings are similar to Fig. 1a.

3.2.3 Computational Complexity

For ease of analysis we assume here $n_1 = \dots = n_k = \dots = n_K > n$. Using an efficient implementation of the Hungarian algorithm (Burkard et al. 2009), the complexity for solving the LSAP is $\mathcal{O}(n_k^3)$. The overall complexity for each iteration of Algorithm 1 is $\mathcal{O}(Kn_k^3 + Kdn^2n_k + K^2dn)$. The number of iterations for Algorithm 1 to converge depends on the initial value of ρ_0 and the factor at which ρ_t increases after each iteration. If ρ_t increases too fast, it has the risk of converging to worse local optima (Lin et al. 2010). In this paper, except experiments reported in Sect. 5.1, we always set $\rho_0 = 1e^{-4}$ and increase it iteratively with a factor of 1.001. Under this setting, it normally takes about 3000 iterations for Algorithm 1 to converge.

The complexity analysis presented above suggests that the practical computation time of solving Algorithm 1 depends on parameter values of n, n_k, K , and d . The feature dimensionality d is usually application dependent, and can be reduced using dimensionality reduction techniques, e.g., PCA, when it is very large. To give an idea on the practical efficiency of Algorithm 1 under different settings of n, n_k , and K , especially when values of these parameters are increased with orders of magnitude, we present simulated experiments using synthetic data, which were generated similarly as in Sect. 3.2.2. More specifically, we fixed $d = 50$, and added sparse errors in each feature vector with a ratio of 0.2. We conducted these experiments on an Intel Xeon CPU running at 2.8GHz, using Matlab implementation of Algorithm 1. Table 1 reports the practical computation time. Table 1 suggests that the running time increases greatly when the number of inliers (value of n) increases with an order of magnitude from 10 to 100, and it increases (roughly) lin-

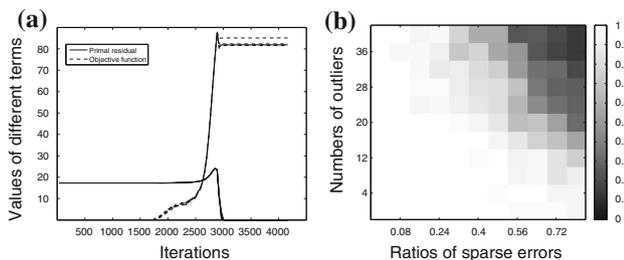


Fig. 1 Simulation of Algorithm 1, using 30 groups of synthetic feature vectors. The number of inliers in each group is fixed as 10. The number of outliers in each group is ranged in $[0, 40]$, and the ratio of sparse errors in each vector is ranged in $[0, 0.8]$. Each setting of outlier number and sparse error ratio is tested with 5 random trials. **a** Convergence plot in terms of the primal residual and objective function for 5 random trials of one test setting (the outlier number is 32 and sparse error ratio is 0.4); **b** recovery rates under different settings of outlier number and sparse error ratio, obtained by averaging over 5 random trials for each test setting

Table 1 Simulation on practical computation time (seconds) of Algorithm 1 under different parameter settings of n, n_k , and K

	$n_k = 100$	$n_k = 1000$	$n_k = 10,000$
$(n = 10, K = 10)$	99.62	95.10	483.27
$(n = 100, K = 10)$	9265.12	8495.03	9013.67
$(n = 10, K = 100)$	821.61	898.53	4669.64
$(n = 100, K = 100)$	75,802.31	83,114.28	91,263.45

All experiments are conducted on an Intel Xeon CPU running at 2.8GHz, using Matlab implementation of Algorithm 1

early with the number of images (value of K), and may not necessarily increase with the number of interest points per image (value of n_k). Notably, Table 1 tells that Algorithm 1 becomes prohibitively slow when the problem scales go up to the level of $n = 100$ and $K = 100$, suggesting the challenge of applying ROML to large-scale feature matching problems. In Sect. 7, we discuss this issue in more details and suggest promising future directions to pursue.

3.2.4 Estimating the Number of Inliers

Up to now we have assumed that the number of inliers is known for a given image set. This might be a strong assumption. In order to investigate how performance of ROML is influenced by this prior knowledge, we conducted simulated experiments that take different values of n , assuming different numbers of inliers, as inputs of Algorithm 1. More specifically, we generated $K = 30$ groups of synthetic feature vectors simulating extracted features from K images, similarly as did in Sect. 3.2.2. There were $n_k = 30$, $k = 1, \dots, K$, feature vectors including both inliers and outliers in every k th group, with dimension of each vector \mathbf{f} as $d = 50$. The ground truth number of inliers was 10 in each group. The inlier feature vectors were generated by randomly drawing as vector entries from i.i.d. normal distribution, and were shared in each of the K groups. The outliers were similarly produced, but were independently generated for each group. We then added sparse errors of large magnitude to both inlier and outlier vectors. For each vector \mathbf{f} , the error values were uniformly drawn from the range $[-2 \max(\text{abs}(\mathbf{f})), 2 \max(\text{abs}(\mathbf{f}))]$. We finally normalized all vectors to constant ℓ_2 -norm. In this investigation, we considered the test settings that the ratio of sparse errors in each feature vector was either 0.2 or 0.4.

By setting integer values of n increasingly from $n = 1$ to $n = n_k$, we compared in each of the n_k cases the identified and matched feature vectors from the K groups by ROML (via Algorithm 1), with the ground truth inlier features distributed in these K groups. We introduce two measures to quantify these comparisons, namely *Match Ratio* and *Identification Ratio*,⁴ which are similar to the measures of *precision*

⁴ From any given K groups of feature vectors, we can enumerate $\frac{K!}{(K-2)!2!}$ pairs of groups. For each group pair, assume ROML (or other feature matching methods) identify n pair-wise correspondences between feature vectors of these two groups. Denote the number of *ground truth correspondences* between inlier features of these two groups as n^* , and the number of *identified ground truth correspondences* by ROML (or other feature matching methods) as \tilde{n} . We define *Match Ratio* as $\sum \tilde{n} / \sum n$, and *Identification Ratio* as $\sum \tilde{n} / \sum n^*$, where \sum sums over all the $\frac{K!}{(K-2)!2!}$ group pairs. Note that when $n = n^*$, i.e., setting value of n as the ground truth number of inliers, the two introduced measures give equal results. Our definitions of the two measures using notions of pair-wise matching are to facilitate comparisons between ROML and traditional feature matching methods, such as graph

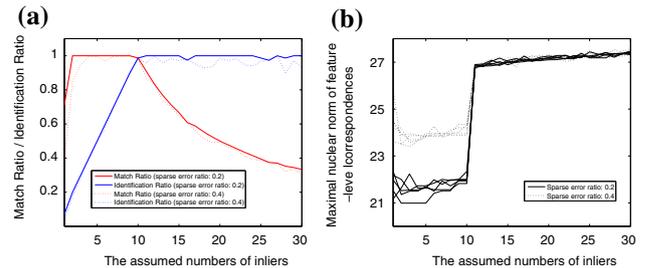


Fig. 2 Simulation for estimating the number of inliers, where two experiment settings are considered (sparse error ratios are 0.2 and 0.4 respectively), and each setting is tested with 5 random trials. The ground truth number of inliers is 10 in all these experiments. **a** Performance of ROML against varying assumed numbers of inliers (varying values of n as inputs of Algorithm 1), measured by Match Ratio and Identification Ratio. Each curve is obtained by averaging over 5 random trials. **b** Maximal nuclear norm of the established n feature-level correspondences by ROML (the γ_n in Eq. (24)) against varying assumed numbers of inliers (varying values of n as inputs of Algorithm 1). Each curve corresponds to one of the total 10 experiments

and *recall* in information retrieval. Figure 2a reports results of Match Ratio and Identification Ratio, which were obtained by averaging over 5 random trials for each setting of sparse error ratio. Figure 2a suggests that when the value of n is close to the ground truth number of inliers, ROML performs better in terms of giving a balanced result of Match Ratio and Identification Ratio, which verifies the importance of knowing the true number of inliers when applying ROML to feature-based object matching. Unfortunately, in many practical applications such as object recognition or 3D reconstruction, this information is usually unknown for a given image set. It becomes essential to develop a mechanism to estimate the number of inliers, in order to apply ROML to these practical problems.

Our consideration of estimating the number of inliers is motivated by the exact low-rank property that ROML leverages for feature matching. Given a value of n , ROML establishes n feature-level correspondences across the K groups by solving Algorithm 1, and forms a matrix $\mathbf{D} \in \mathbb{R}^{dn \times K}$, defined as Eq. (2), using the obtained PPMs $\{\mathbf{P}^k\}_{k=1}^K$. Each of the n feature-level correspondences consists of K feature vectors respectively from the K groups, and corresponds to a submatrix of \mathbf{D} , which we denote as $\mathbf{D}_j \in \mathbb{R}^{d \times K}$ with $j \in \{1, \dots, n\}$. Equation (2) indicates $\mathbf{D} = [\mathbf{D}_1^\top, \dots, \mathbf{D}_n^\top]^\top$. When any j th correspondence is formed by inlier features, the corresponding \mathbf{D}_j would be rank deficient, and numerically would have lower nuclear norm. Conversely, the nuclear norm would be higher. We thus consider using the measure of $\|\mathbf{D}_j\|_*$, $j = 1, \dots, n$, for estimating the number of inliers.

Footnote 4 continued
 matching (Torresani et al. 2008; Cho et al. 2010; Leordeanu and Hebert 2005), which work in the setting of matching between a pair of images/groups. Section 5 presents such comparisons.

More specifically, given a data set of K groups of feature vectors, we solve a series of ROML problems (via Algorithm 1) with values of n set increasingly from $n = 1$ to $n = n_k$. For any specific value of n , a matrix $\mathbf{D} \in \mathbb{R}^{dn \times K}$ can be obtained when Algorithm 1 converges. Define *maximal nuclear norm of feature-level correspondences* as

$$\gamma_n = \max_{j=1, \dots, n} \|\mathbf{D}_j\|_*, \quad (24)$$

which computes maximum of the nuclear norms of the established n feature-level correspondences $\{\mathbf{D}_j\}_{j=1}^n$. Given the series $\{\gamma_1, \dots, \gamma_n\}$ obtained by increasingly setting $n = 1$ to the current n value, we further define $\bar{\gamma}_n = \sum_{i=1}^n \gamma_i / n$, which averages this series. Based on the notions of γ_n and $\bar{\gamma}_n$, we propose a simple scheme that estimates the number of inliers as the current value of n if

$$(\gamma_{n+1} - \bar{\gamma}_n) / \bar{\gamma}_n > \delta, \quad (25)$$

where δ is a scalar parameter. To investigate the efficacy of this proposed scheme, we plot in Fig. 2b the curves of γ_n values from $n = 1$ to $n = n_k$, for those simulated experiments reported in Fig. 2a. Figure 2b shows that for any curve corresponding to one of these simulated experiments, there is a clear stepping of γ_n values when n is set as the ground truth number of inliers, indicating that γ_n is a good measure for estimation of inlier numbers. Our use of $\bar{\gamma}_n$ in (25) is to improve the estimation robustness. We also observe from Fig. 2b that the gap of curve stepping becomes narrower when the ratio of sparse errors is increased in each feature vector, suggesting less efficacy of the proposed scheme (25). This is reasonable since when errors in feature vectors increase, it becomes less distinctive between inlier and outlier features.

Simply put, our proposed scheme starts from setting $n = 1$, and solves a series of ROML problems with increasing values of n , until the condition (25) is satisfied. The ratio computed in (25) is numerically stable w.r.t. varying values of d (feature dimension) and K (number of groups/images). We set the parameter $\delta = 0.05$ throughout this paper. It gives perfect estimation of the true numbers of inliers for all the simulated experiments reported in this section. In Sect. 5.2, we report experiments when applying this scheme to practical data.

3.2.5 Detection of the True Inliers

By solving the problem (5), ROML is able to identify n features from each of a set of K images and establish their correspondences. However, these nK features are not necessarily all the true inliers. This is particularly the case when some true inlier features are contaminated with noise, e.g., due to object appearance variations caused by illumination or pose changes, and when some inliers are missing, e.g., due

to partial occlusion of object instances. Detecting the true inliers out of the nK features is practically useful for applications such as 3D reconstruction and object recognition.

We have introduced an error term $\mathbf{E} \in \mathbb{R}^{dn \times K}$ in (5) to improve the robustness of ROML against the aforementioned contaminations of inlier features. One may think of using \mathbf{E} for detection of the true inliers, e.g., by thresholding ℓ_1 -norms of the nK d -dimensional subvectors in \mathbf{E} (n subvectors per column). However, the ROML formulation (5) is a non-convex problem. It decomposes out sparse errors into \mathbf{E} mainly for obtaining better PPMs $\{\mathbf{P}^k\}_{k=1}^K$, so that potential inliers from the K images can be identified and matched in $\mathbf{D} = [\text{vec}(\mathbf{F}^1 \mathbf{P}^1), \dots, \text{vec}(\mathbf{F}^K \mathbf{P}^K)]$. By the non-convex problem nature of (5), the obtained \mathbf{E} is not guaranteed to be consistent with those noise contaminating inlier features, especially when inlier features are heavily contaminated and become less distinctive from outliers. This is different from the RPCA problem (Candès et al. 2011) where an exact solution of the low-rank matrix and sparse error matrix can be obtained with theoretical guarantee. To detect the true inliers from the selected features by ROML, we propose an alternative scheme that first solves a RPCA problem: $\min_{\mathbf{L}_{rpca}, \mathbf{E}_{rpca}} \|\mathbf{L}_{rpca}\|_* + \lambda_{rpca} \|\mathbf{E}_{rpca}\|_1$ s.t. $\mathbf{D} = \mathbf{L}_{rpca} + \mathbf{E}_{rpca}$, using the obtained \mathbf{D} from ROML as the input data, where $\lambda_{rpca} = 1/\sqrt{dn}$ as theoretically derived in RPCA (Candès et al. 2011). The proposed scheme then thresholds ℓ_1 -norms of the nK d -dimensional error subvectors in $\mathbf{E}_{rpca} \in \mathbb{R}^{dn \times K}$, which are the decomposed error vectors respectively for the selected nK features by ROML. For any j th selected feature from the k th image, we determine it as a true inlier if

$$\|[\mathbf{E}_{rpca}]_{jd-d+1:jd,k}\|_1 < \xi, \quad (26)$$

where $[\mathbf{E}_{rpca}]_{jd-d+1:jd,k}$ denotes the error subvector with $j \in \{1, \dots, n\}$ and $k \in \{1, \dots, K\}$, and ξ is a scalar parameter. The scheme (26) is practically useful if magnitude of \mathbf{E}_{rpca} is stable when applying ROML (and the subsequent RPCA problem) to various data applications, so that ξ is less concerned with tuning. Note that magnitudes of entries in \mathbf{E}_{rpca} are controlled by two factors: magnitudes of entries in the feature vectors $\{\mathbf{F}^k\}_{k=1}^K$, and levels of noise contaminating these feature vectors. Since all the feature vectors in $\{\mathbf{F}^k\}_{k=1}^K$ have been normalized in Algorithm 1, magnitude of \mathbf{E}_{rpca} is less influenced by the first factor. To investigate how the scheme (26) performs w.r.t. the second factor, we conducted simulated experiments using varying levels of noise and ratios of missing true inliers. Given unit ℓ_2 -norm of normalized feature vectors in $\{\mathbf{F}^k\}_{k=1}^K$, we set $\xi = 4$ for all relevant experiments of inlier detection reported in this paper.

More specifically, we randomly generated synthetic data similarly as did in Sect. 3.2.4. We set $K = 30$ and $n_k = 30$ for $k = 1, \dots, K$. Dimension of each feature vector was set

Table 2 Simulation for detection of true inliers, using the proposed scheme (26)

Ratios of miss. inliers	5%	10%	30%	50%
Noise L. I	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00
Noise L. II	1.00/1.00	1.00/1.00	1.00/0.99	0.99/0.98
Noise L. III	1.00/0.99	1.00/0.96	1.00/0.95	0.99/0.92

Synthetic data are generated under 12 test settings: the ratios of sparse errors in each feature vector are 10% (Noise Level I), 30% (Noise Level II), or 50% (Noise Level III), and the ratios of missing inliers in each group are 5, 10, 30, or 50%. Results for each setting are obtained by averaging over 5 random trials and presented in the format of *Precision/Recall*

as $d = 50$. Sparse errors were added to feature vectors in the same way as in Sect. 3.2.4. The number of inliers was set as $n = 10$, which gives a total of $nK = 300$ true inliers from the K groups/images. After inlier features were generated, we randomly replaced a certain number of them with additionally generated outlier features, simulating the situation of missing inliers. In this investigation, we considered the test settings where the ratios of sparse errors in each feature vector were ranged in $[0.1, 0.5]$, simulating increased levels of noise, and the ratios of missing inliers for each group were ranged in $[0.05, 0.5]$. Under each setting, performance of the scheme (26) is measured by *precision* and *recall*. Precision is computed as the number of detected true inliers divided by the total number of detected features, and recall is computed as the number of detected true inliers divided by the total number of true inliers contained in the K groups. We run 5 random trials and averaged the results under each setting. Table 2 reports these simulated experiments, which suggests that precision scores of detecting the true inliers by the scheme (26) are very high for the considered test settings (in fact perfect precision for most of the settings). When the levels of noise (ratios of sparse errors) contaminating feature vectors increase, magnitudes of the decomposed error subvectors in \mathbf{E}_{rpca} for the true inliers also increase, and become less distinctive from those of outliers, resulting in slightly reduced recall scores. Nevertheless, results in Table 2 show that in a wide range of noise and missing inlier settings, the proposed scheme (26) is effective for detecting the true inliers. In Sect. 5.2, we also report experiments of applying (26) to practical data for detection of true inliers.

4 Choices of Feature Types and Their Applicable Spectrums

In the previous sections, we have represented an image as a set of features, where features generally refer to vectors characterizing image points and local regions centered on them. The task of object matching is then posed as Problem

1. Depending on different applications, these features can be chosen as either image coordinates, local region descriptors, or combination of them encoding both spatially structural and local appearance information. In the following, we present details of different choices of feature types and their applicable spectrums for robust object matching.

4.1 Image Coordinates

Given a set of points in an image, their coordinates can be directly used as features. In fact, coordinates of a set of inlier points in an image encode geometric relations among them, and it is the geometric structure of these points that determines the object pattern, and also provides a constraint for use in object matching. Image coordinates based features have been intensively used in early shape matching works (Scott and Longuet-Higgins 1991; Shapiro 1992; Besl and McKay 1992; Chui and Rangarajan 2003; Belongie et al. 2002).

For a moving rigid object in a video sequence or images of a rigid object captured from different viewpoints, denote $\mathbf{f}_i^k = [x_i^k, y_i^k]^\top \in \mathbb{R}^2$, $i = 1, \dots, n_k$, as image coordinates based n_k features extracted from the k th image. Let $\mathbf{F}^k = [\mathbf{f}_1^k, \dots, \mathbf{f}_{n_k}^k] \in \mathbb{R}^{2 \times n_k}$. It has been shown in Tomasi and Kanade (1992) that the matrix, defined by

$$\mathbf{D}'(\{\mathbf{P}^k\}_{k=1}^K) = [(\mathbf{F}^1 \mathbf{P}^1)^\top, \dots, (\mathbf{F}^K \mathbf{P}^K)^\top]^\top \in \mathbb{R}^{2K \times n}, \quad (27)$$

is highly rank deficient (at most rank 4 when considering translation and there is no measurement noise), if correct PPMs $\{\mathbf{P}^k\}_{k=1}^K$ are used so that n inlier points can be selected from each of $\{\mathbf{F}^k\}_{k=1}^K$ and corresponding points $\{\mathbf{f}_j^k\}_{k=1}^K$, $j \in \{1, \dots, n\}$, can be aligned in the same column of \mathbf{D}' . (27) is different from the formation of $\mathbf{D}(\{\mathbf{P}^k\}_{k=1}^K)$ in (2). By applying the same low-rank and sparse constraints as in (5), we will show in Sect. 5.1 that image coordinates based features are very useful for matching rigid objects.

4.2 Local Region Descriptors

It is also straightforward to use region descriptors characterizing locally visual appearance information as features. These include SIFT (Lowe 1999), HOG (Dalal and Triggs 2005), Geometric Blur (Berg and Malik 2001; Berg et al. 2005), GIST (Oliva and Torralba 2001), or even raw pixels of local patches. In general, these feature descriptors have the properties of invariance and distinctiveness. The invariance property makes it possible to match salient features extracted from images under geometric transformation or illumination change, while feature distinctiveness is important to differentiate between different salient regions. Features of such

kind can be used in scenarios where they are discriminative enough for matching, or geometric constraints between feature points are not available, such as common object localization (Deselaers et al. 2012; Zhu et al. 2012). In Sect. 6, we present how ROML can be applied to this application.

4.3 Combination of Image Coordinates and Region Descriptors

Local region descriptors alone could be ambiguous for feature matching when there exist repetitive textures or less discriminative local appearance in images. To improve the matching accuracy, it is necessary to exploit the geometric structure of inlier points that consistently appears in each of the set of images. In literature, there are many ways to exploit such geometric constraints, such as pair-wise compatibility of feature correspondences used in graph matching (Leordeanu and Hebert 2005; Torresani et al. 2008), or linear-form constraints benefiting from a template image (Li et al. 2011; Jiang and Yu 2009). In this work, we consider a simple method introduced in Torki and Elgammal (2010). For any interest point in each of the set of images, this method learns a low-dimensional embedded feature vector that combines information of both the local appearance and the spatial relations of this point relative to other points in the image. We present details of how to compute this type of learned embedded features in Appendix 1.

As suggested by Theorem 1, when there are no outliers, the thus learned features can be directly used in our ROML framework. When there exist outliers in any of the set of images, we can always normalize those features to let them have constant ℓ_2 -norm, and our method still applies. Since this type of learned features encode both appearance and spatial layout information, our method can potentially apply in more general settings, such as matching of non-rigid, articulated objects, or instances of a same object category. Experiments in Sect. 5 show the promise.

5 Experiments

In this section, we present experiments to show the effectiveness of ROML for robustly matching objects in a set of images. We consider different testing scenarios from the relatively simple rigid object matching, to the more challenging matching of object instances of a common category, and matching a non-rigid object moving in a video sequence. For these testing scenarios, we choose appropriate feature types of either image coordinates or combination of image coordinates and local region descriptors, while features of region descriptors alone will be used in Sect. 6 for the application of common object localization. In the following experiments (except those in Sect. 5.1), we always set the penalty parameter $\lambda = 5/\sqrt{dn}$ when solving the ROML problem (5) using

Algorithm 1, where ρ was initially set as $1e^{-4}$ and iteratively increased with a factor of 1.001.

5.1 Rigid Object with 3D Motion

The CMU “Hotel” sequence consists of 101 frames of a toy hotel building undergoing 3D motion.⁵ Each frame has been manually labelled with the same set of 30 landmark points (Caetano et al. 2009), i.e., $n = 30$. We use the “Hotel” sequence to show that ROML can be applied using image coordinates as features for matching rigid objects. In particular, we sampled $K = 15$ frames out of the total 101 frames (every 7 frames), in order to simulate the wide baseline matching scenario. Given PPMs $\{\mathbf{P}^k\}_{k=1}^K$, image coordinates of landmark points in these 15 frames were arranged into a matrix $\mathbf{D}'(\{\mathbf{P}^k\}_{k=1}^K) \in \mathbb{R}^{2K \times n}$, as defined in (27). We used Algorithm 1 to optimize a PPM for each frame, where the penalty parameter was set as $\lambda = 5/\sqrt{2K}$, and ρ was initialized as $1e^{-6}$ and iteratively increased with a factor of 1.0001.

We compare our method with representative pair-wise graph matching methods including Dual Decomposition (DD) (Torresani et al. 2008), SMAC (Cour et al. 2006), and Learning Graph Matching (LGM) (Caetano et al. 2009), which are based on either linear or quadratic assignment formulations, and also with more related methods (Oliveira et al. 2005; Torki and Elgammal 2010) that are able to simultaneously match the set of 15 frames. For the former set of methods, matchings between a total of 105 frame pairs need to be established. Note that although all these methods are based on image coordinates, many of them have used the advanced shape context features (Belongie et al. 2002). To evaluate the performance of different methods, we use the Match Ratio criteria, which is defined in Sect. 3.2.4. Table 3 reports the Match Ratios of different methods, where results of SMAC and LGM are from Torresani et al. (2008), Torki and Elgammal (2010), and result of RankCon (Oliveira et al. 2005) is from our own code implementation of the method. Table 3 tells that ROML and One-Shot (Torki and Elgammal 2010) achieve the best performance (no matching error). However, One-Shot (Torki and Elgammal 2010) uses shape context feature to characterize each landmark point, and it performs a low-dimensional embedding combining information of both geometric structure and local descriptors of landmark points, while ROML just directly uses image coordinates. RankCon (Oliveira et al. 2005) also exploits low-rank constraints, however, its performance is much worse than that of ROML. This is due to the inherent error-propagation nature of the method. Matchings of landmark points in RankCon (Oliveira et al. 2005) are established in a frame-by-frame manner; once landmarks in the current frame are matched to those in the previous frames,

⁵ <http://vasc.ri.cmu.edu/idb/html/motion/hotel/>.

Table 3 Results of different methods on the “Hotel” sequence

Methods	DD (Torresani et al. 2008) (%)	SMAC (Cour et al. 2006) (%)	LGM (Caetano et al. 2009) (%)	RankCon (Oliveira et al. 2005) (%)	One-Shot (Torki and Elgammal 2010) (%)	Prev (Zeng et al. 2012) (%)	ROML-Pair (%)	ROML (%)
Accuracies	99.8	84	90	57	100	72	100	100

Accuracies are measured by the Match Ratio criteria
Best results are highlighted in bold

Table 4 Match Ratios of different methods on the “Hotel” sequence with missing landmark (inlier) points

No. of missing points	RankCon (Oliveira et al. 2005) (%)	One-Shot (Torki and Elgammal 2010) (%)	ROML
1	43	76	95
3	26	64	79
5	23	59	71

The cases of 1, 3, and 5 missing points in each frame are tested (the corresponding ratios of missing points over the total 30 ones are 3, 10, and 17 % respectively)

Best results are highlighted in bold

the matchings cannot be corrected in the later stages, and matching errors will inevitably propagate and accumulate. Thus for RankCon (Oliveira et al. 2005), it is critical to make the matchings of the first few frames accurate. Unfortunately, the low-rank property leveraged by RankCon (Oliveira et al. 2005) is weak given only landmark points in the first few frames. We also present result of our previous method (Zeng et al. 2012) (Prev Zeng et al. 2012) in Table 3, which corroborates our discussions on the advantage of ROML over (Zeng et al. 2012 in Sect. 1). It is interesting to observe that under the setting of pair-wise matching as for methods (Torresani et al. 2008; Cour et al. 2006; Caetano et al. 2009), ROML still performs perfectly on the “Hotel” sequence (“ROML-Pair” in Table 3). In fact, since the problem size of matching each frame pair is smaller, ROML converges faster with less iterations.

Compared to Oliveira et al. (2005), Torki and Elgammal (2010), ROML has the additional advantage of being more robust against missing inliers. To verify, we performed another experiment by removing randomly selected landmark points in each frame. For each removed landmark point, we also generated arbitrary image coordinates for it and made sure the generated coordinates were far enough away from the true ones, in order to fit with the algorithmic settings of these comparative methods. We set $\lambda = 2/\sqrt{2K}$ in Algorithm 1. One-Shot (Torki and Elgammal 2010) uses k-means clustering to obtain feature correspondences in the learned feature space. We chose its best-performing dimensionality of learned features, and run 10 trials of k-means clustering and averaged the results. Parameters of RankCon (Oliveira et al. 2005) has also been tuned to its best performance. Table 4 reports the Match Ratio results, which are computed over the non-missing points. Table 4 clearly shows that ROML is less influenced when there exist missing inlier points.

5.2 Object Instances of a Common Category

In this section, we test how ROML performs to match object instances belonging to a same object category. We used 6 image sets of different categories from Caltech101 (Fei-Fei

et al. 2007), MSRC,⁶ and the Internet. Numbers of images in these 6 sets ranged from 16 to 25. For each image, interest points were detected by SIFT: the numbers of detected interest points per image were from 27 to 174, out of which we manually labelled inlier points as matching ground truth. When some inlier points were not detected by SIFT in some images, we also manually labelled them in order to produce consistent sets of inlier points across the sets of images. In these experiments, we chose the low-dimensional embedded feature representation (Torki and Elgammal 2010), as explained in Sect. 4.3, which encodes information of both geometric structure and descriptor similarity. To learn the embedded features, we used Geometric Blur descriptors (Berg et al. 2005) to characterize local regions around interest points, and Euclidean distances between points in each image for measuring geometric relations. The parameters for embedded feature learning were set as $\sigma_{spa.} = 10$ and $\sigma_{des.} = 0.2$ (cf. Appendix 1 for the definition of $\sigma_{spa.}$ and $\sigma_{des.}$), and the dimensionality of learned features was set as $d = 60$. Feature matching was realized by solving (5) using Algorithm 1.

We compare our method with One-Shot (Torki and Elgammal 2010), and also with several recent graph matching methods including DD (Torresani et al. 2008), RRWM (Cho et al. 2010), and SM (Leordeanu and Hebert 2005), and hyper-graph matching methods including TM (Duchenne et al. 2011), RRWHM (Lee et al. 2011), and ProbHM (Zass and Shashua 2008). Codes of DD (Torresani et al. 2008), RRWM (Cho et al. 2010), TM (Duchenne et al. 2011), and RRWHM (Lee et al. 2011) are respectively from their authors' publicly available webpages. For ProbHM (Zass and Shashua 2008), we used a code implementation provided by the authors of RRWHM (Lee et al. 2011). The simple methods of One-Shot (Torki and Elgammal 2010) and SM (Leordeanu and Hebert 2005) are based on our own implementations. For One-Shot, we chose its best-performing dimensionality of the learned embedded features as our method used, and run 10 trials of k-means clustering and averaged the results. For pair-wise graph matching methods (Torresani et al. 2008; Cho et al. 2010; Leordeanu and Hebert 2005; Duchenne et al. 2011; Lee et al. 2011; Zass and Shashua 2008), we generated a total of $\frac{K!}{(K-2)!2!}$ image pairs for each test set with K images. These graph matching methods characterize interest points in each image by both their spatial relations and their respective local region descriptors. For a fair comparison, we used the same Geometric Blur region descriptors as our method used.⁷ Their parameters

⁶ The msrc dataset <http://research.microsoft.com/en-us/projects/objectclassrecognition/>.

⁷ We have also tried the SIFT descriptor (Lowe 1999) to characterize appearance of local regions around interest points. The matching

Table 5 Match Ratios of different methods on 6 image sets of different object categories

Methods	RRWM (Cho et al. 2010) (%)	SM (Leordeanu and Hebert 2005) (%)	TM (Duchenne et al. 2011) (%)	RRWHM (Lee et al. 2011) (%)	ProbHM (Zass and Shashua 2008) (%)	DD (Torresani et al. 2008) (%)	OneShot (Torki and Elgammal 2010) (%)	Prev (Zeng et al. 2012) (%)	ROML
Airplanes	28	54	17	54	32	70	65	87	95
Face	40	57	26	54	14	64	61	53	89
Motorbike	50	46	23	58	28	73	68	89	99
Car	26	39	12	23	12	51	50	59	81
Bus	13	25	24	43	18	52	44	64	79
BoA	7	12	6	15	7	12	16	35	75

Best results are highlighted in bold

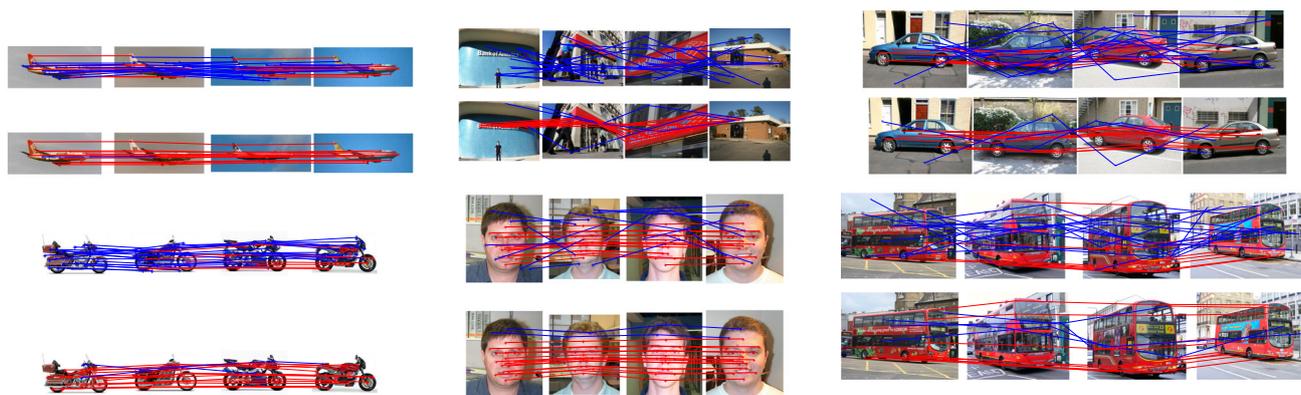


Fig. 3 Example feature correspondences among 4 images for different image sets. For every pair top is from DD (Torresani et al. 2008), and bottom is from our method. Red lines represent identified ground truth correspondences, and blue lines are for false ones (Color figure online)

were also tuned to their respective best performance on the 6 test sets.

Table 5 reports results of different methods in terms of Match Ratio. Example feature correspondences for DD (Torresani et al. 2008) and our method are shown in Fig. 3. Table 5 and Fig. 3 suggest that for the relatively simple “Airplane”, “Motorbike”, and “Face” image sets, our method gives very good matching results. The “Car”, “Bus”, and “Bank of America (BoA)” sets are more difficult due to the cluttered background, large viewpoint changes, or intra-category variations between different instances. Our method still gives reasonably good and consistent matching results. Both One-Shot and our method can match multiple images simultaneously. Our results are much better than those of One-Shot, which shows that One-Shot cannot perform well in the presence of outliers, and also that our ROML formulation optimized by the ADMM method is very effective for multi-image feature matching. Our method greatly outperforms graph and hyper-graph matching methods. It demonstrates that leveraging more object pattern constraints (i.e., geometric and feature similarity constraints) from multiple images is very useful for feature matching. Moreover, Fig. 3 suggests that our matching results across the 4 images are more consistent than those from graph matching methods: another desired property for many computer vision applications. In Table 5, we also compare with our previous method (Zeng et al. 2012) (Prev Zeng et al. 2012 in Table 5). Results of Prev (Zeng et al. 2012) are obtained using the same low-dimensional embedded features as ROML does. Table 5 tells that on all the 6 image sets, matching accuracies of ROML are much better than those of Prev (Zeng et al. 2012). The improved accuracy comes from the new way of PPM optimization, i.e., exactly solving an equivalent LSAP in the

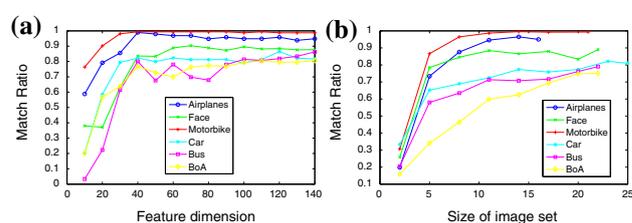


Fig. 4 Performance of ROML using **a** different choices of dimension d in low-dimensional feature learning and **b** different sizes of image sets (the K values)

present paper instead of sub-optimally solving two costly subproblems in Zeng et al. (2012), as we have discussed in Sect. 1.

Different choices of dimension d in low dimensional feature learning may influence our method’s performance. In Fig. 4a, we plot our matching accuracies with different choices of d on these 6 test sets. It shows that better results can generally be obtained when $d \geq 40$. It is expected that our method performs well only when the size of image sets (the K value) is relatively large. In Fig. 4b, we plot results of our method on the 6 test sets with different choices of K . It shows that when $K > 10$, our method can stably get good results, which confirms that simultaneously matching a set of images is very useful for robust object matching.

Except for matching accuracy, one may also be interested in comparing matching efficiency of different methods. We have analyzed the computational complexity of our proposed method in Sect. 3.2.3. In Table 6, we report practical computation time of different methods for those experiments reported in Table 5. These experiments were conducted on an Intel Xeon CPU running at 2.8GHz, using Matlab implementation of different methods. Table 6 suggests that ROML is much more efficient than the best-performing graph matching method DD (Torresani et al. 2008), and is slightly slower than other graph/hyper-graph matching methods. One-Shot (Torki and Elgammal 2010) is very fast, however, its accu-

Footnote 7 continued

accuracies using SIFT were slightly worse than those using Geometric Blur for both our method and these comparative methods.

Table 6 Computation time (seconds) of different methods on the 6 image sets used in Table 5

Methods	Airplanes	Face	Motorbike	Car	Bus	BoA
RRWM (Cho et al. 2010)	22.92	211.10	53.20	260.83	157.30	240.60
SM (Leordeanu and Hebert 2005)	5.44	61.71	17.61	39.31	20.61	43.15
TM (Duchenne et al. 2011)	44.01	418.30	145.38	282.92	187.39	315.04
RRWHM (Lee et al. 2011)	28.27	178.34	52.40	255.72	100.14	185.97
ProbHM (Zass and Shashua 2008)	39.58	375.48	128.45	252.04	168.70	269.31
DD (Torresani et al. 2008)	2145.56	11,714.50	3999.30	3688.99	3972.35	3827.95
One-Shot (Torki and Elgammal 2010)	0.50	1.65	1.15	1.73	1.44	1.40
Prev (Zeng et al. 2012)	874.19	6641.19	2497.25	2220.49	1665.51	2640.98
Prev (Zeng et al. 2012)-Parallel	219.79	1664.23	626.41	559.84	419.05	662.25
ROML	169.63	839.95	376.48	305.51	236.07	415.51
ROML-Parallel	52.09	236.24	110.90	106.36	74.74	124.55

All experiments were conducted on an Intel Xeon CPU running at 2.8GHz, using Matlab implementation of different methods

racy (reported in Table 5) is not satisfactory. As an improved method of our previous work (Zeng et al. 2012), ROML is much more efficient than (Zeng et al. 2012) as well. The improved efficiency is again due to the new way of PPM optimization in the present paper. In spite of this improved efficiency, most of ROML's computation is still on solving LSAPs for updating the set of K PPMs (steps 5–7 in Algorithm 1), which concerns with K independent subproblems and are fully parallelizable. When implementing the PPM optimization steps in parallel (ROML-Parallel in Table 6), efficiency of ROML is further improved.

Results of ROML in the above experiments are obtained by setting the value of n as the ground truth number of inliers for each image set. In practice, however, the true number of inliers is unknown. It is interesting to investigate how ROML performs when providing different values of n to Algorithm 1, assuming different numbers of inliers. We conducted such experiments using the same 6 image sets. Results are plotted in Figure 5 in terms of Match Ratio and Identification Ratio (defined in Sect. 3.2.4). Figure 5 tells that better results can be achieved when the values of n are close to the ground true ones, indicating the importance of knowing this prior knowledge. We have proposed in Sect. 3.2.4 a scheme for estimating the true number of inliers for a given image set, i.e., the condition (25). Applying this scheme to the 6 image sets gives the estimation results listed in Table 7. Compared with the true ones, results in Table 7 seem biased towards smaller estimations. This may be due to large variations among different object instances that exist in these practical data. As a result, some of the corresponded inlier features across an image set become less correlated, and thus more like outliers. Nevertheless, Table 7 and Fig. 5 suggest that in these practical problems, we can still use the estimated numbers of inliers (as inputs of Algorithm 1) to establish certain numbers of accurate feature matchings, as demonstrated by the Match Ratio results in Fig. 5.

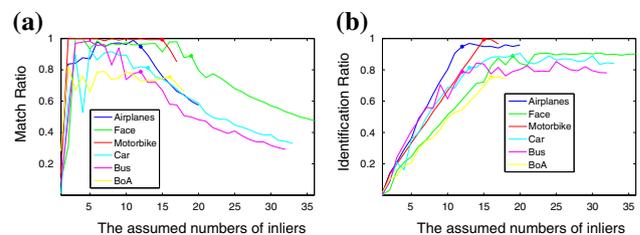


Fig. 5 Match Ratio **a** and Identification Ratio **b** of ROML on 6 image sets of different object categories, obtained by providing different values of n (the assumed number of inliers for each image set) to Algorithm 1. The mark on each curve corresponds to the true number of inliers

Table 7 Estimation of inlier numbers on 6 image sets of different object categories, using the proposed scheme (25)

	Ground truth inlier number	Estimated inlier number
Airplanes	12	9
Face	19	11
Motorbike	15	13
Car	13	10
Bus	12	5
BoA	16	11

In the above experiments, we evaluate ROML's performance by comparing the ground truth inlier features and their correspondences with those identified by ROML, e.g., the measure of Match Ratio. In practice, the ground truth information on which features are inliers is unavailable for a given image set. It is particularly interesting to detect the true inliers out of those features identified by ROML, so as to make ROML more useful for practical problems such as 3D reconstruction and object recognition. We have developed such a scheme (26) in Sect. 3.2.5. Applying our proposed scheme to the 6 image sets used in this section gives the results listed in Table 8, where performance of the scheme (26) is measured

Table 8 Detection of true inliers on 6 image sets of different object categories, using the proposed scheme (26)

Ratios of miss. inliers	5 %	10 %	30 %	50 %
Airplanes	1.00/0.81	1.00/0.76	0.84/0.72	0.69/0.63
Face	0.91/0.77	0.92/0.81	0.84/0.75	0.73/0.69
Motorbike	1.00/0.93	0.95/0.87	0.81/0.78	0.62/0.70
Car	0.94/0.78	0.93/0.77	0.80/0.67	0.62/0.60
Bus	0.94/0.62	0.97/0.63	0.83/0.63	0.64/0.48
BoA	1.00/0.73	0.90/0.60	0.63/0.42	0.53/0.39

For each image set, different portions of randomly chosen ground truth inliers are replaced with outliers, to simulate the scenarios of missing inliers. Results are presented in the format of *Precision/Recall*

Table 9 Match Ratios of ROML on 6 image sets of different object categories

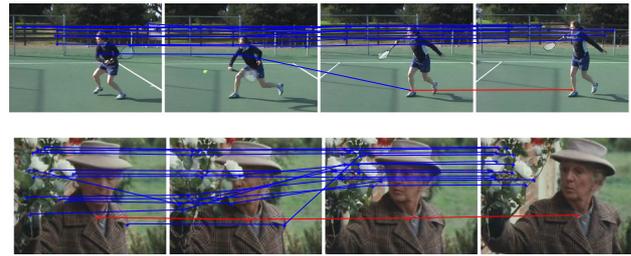
Ratios of missing inliers	5 %	10 %	30 %	50 %
Airplanes	95 %	94 %	74 %	69 %
Face	80 %	84 %	77 %	72 %
Motorbike	98 %	93 %	84 %	69 %
Car	82 %	79 %	71 %	57 %
Bus	68 %	72 %	66 %	55 %
BoA	76 %	59 %	38 %	30 %

For each image set, different portions of randomly chosen ground truth inliers are replaced with outliers, to simulate the scenarios of missing inliers

by precision and recall scores (defined in Sect. 3.2.5). These results were obtained by randomly replacing different portions of ground truth inliers with outliers for each image set, to simulate the scenarios of missing inliers. Table 8 shows that precision scores are generally high when not many true inliers are missing, and the corresponding recall scores are satisfactory to give enough numbers of true inliers for practical use. Corresponding Match Ratio results for experiments reported in Table 8 are also presented in Table 9, where Match Ratio is computed based on non-missing inliers. Table 9 suggests that ROML is less influenced when only a small portion of ground truth inliers are missing.

5.3 Non-rigid Object Moving in a Video Sequence

Lastly, we test how ROML performs to match a non-rigid object moving in a video sequence. This could be a much harder application scenario than that in Sect. 5.2. For example, in a video sequence with static background, some salient points in the background consistently appear across the video frames, with little change of appearance, and they can form their own “low-rank pattern”. The corresponding salient points (inliers) on the non-rigid foreground of interest may only appear in some of the frames, possibly with changing appearance. In this section, we used a 25-frame

**Fig. 6** Illustration of the failure of ROML on matching a non-rigid object moving in a video sequence. Most of the identified correspondences by ROML are from the background (*blue lines*), rather than from the foreground object of interest (*red lines*) (Color figure online)

“Tennis” sequence and a 50-frame “Marple” sequence (Brox and Malik 2010) to test how ROML performs in this challenging scenario. For the first sequence, we used KLT tracker (Lucas and Kanade 1981) to detect 100 interest points in each frame. For the second one, we detected 150 interest points in each frame. We again used the type of learned embedded features as explained in Sect. 4.3. The parameter settings were the same as those used in Sect. 5.2. Figure 6 illustrates the matching results of ROML among 4 frames of these two sequences respectively. Clearly, most of the identified correspondences by ROML are from the background, showing the failure when applying ROML to this challenging scenario. In fact, comparative graph and hyper-graph matching methods all failed on these two sequences. This difficulty of mining/matching foreground objects from video sequences with static background poses a common challenge for these feature-based object matching methods.

The above challenge can be largely alleviated when inliers from the foreground object can be specified in the first frame of a video sequence. This resembles an object tracking scenario. In the following, we report experiments that show how ROML performs in this alleviated non-rigid object matching scenario. More specifically, given a video sequence with interest points detected in each of the total K frames, we label inlier points from those detected in the first frame. The task is to match these inlier points, which are supposed to be on the object of interest, across the subsequent $K - 1$ video frames, simulating a tracking scenario. We again used the “Tennis” and “Marple” sequences. We labelled 11 inliers from the 100 detected interest points in the first frame of the “Tennis” sequence, and 14 inliers from the 150 detected interest points in the first frame of the “Marple” sequence. To adapt our method to this scenario, we simply fix \mathbf{P}^1 in steps 5–7 of Algorithm 1 so that it selects the n inliers labelled in the first frame, while optimizing the other $K - 1$ PPMs $\{\mathbf{P}^k\}_{k=2}^K$.⁸ We compare ROML with a baseline KLT tracker, and recent graph and hyper-graph matching meth-

⁸ To make this scheme effective, we *emphasize* the labelled first frame by normalizing feature vectors of its interest points to have a larger value

Table 10 Match Ratios of different methods on the “Tennis” and “Marple” sequences (Brox and Malik 2010)

Methods	KLT (Lucas and Kanade 1981) (%)	RRWM (Cho et al. 2010) (%)	SM (Leordeanu and Hebert 2005) (%)	TM (Duchenne et al. 2011) (%)	RRWHM (Lee et al. 2011) (%)	ProbHM (Zass and Shashua 2008) (%)	DD (Torresani et al. 2008) (%)	Prev (Zeng et al. 2012) (%)	ROML
Tennis	3	23	43	13	18	16	57	52	73
Marple	4	3	25	8	13	14	23	41	51

Best results are highlighted in bold

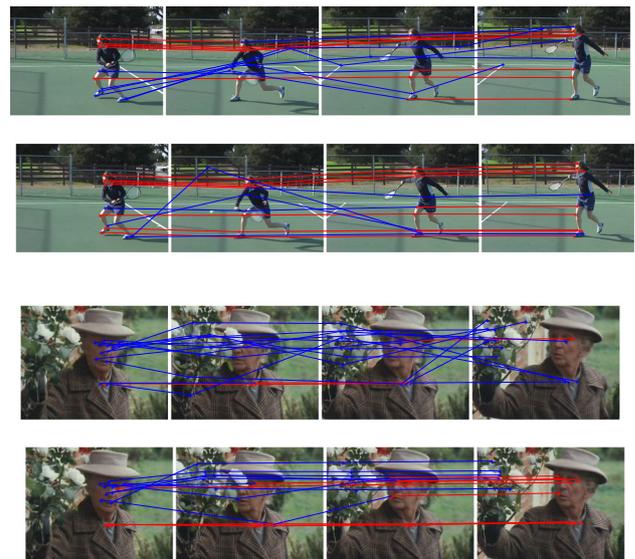


Fig. 7 Example correspondences of interest points among 4 frames of the “Tennis” and “Marple” sequences (Brox and Malik 2010) respectively. For every pair top is from DD (Torresani et al. 2008), and bottom is from our method. Red lines represent identified ground truth correspondences, and blue lines are for false ones (Color figure online)

ods (Torresani et al. 2008; Cho et al. 2010; Leordeanu and Hebert 2005; Duchenne et al. 2011; Lee et al. 2011; Zass and Shashua 2008). Since inlier points in the first frame are labelled ground truth, for graph and hyper-graph matching methods, we generated 24 and 49 frame pairs for the “Tennis” and “Marple” sequences respectively, i.e., between the first frame and each of the other frames, and used them for pair-wise matching. The other settings of these methods were the same as those used to produce Table 5 in Sect. 5.2. Parameters of these methods were also tuned to their respective best results on the two sequences. Table 10 reports the quantitative results of different methods in terms of Match Ratio. Example correspondences of interest points for DD (Torresani et al. 2008) and our method are also shown in Fig. 7. KLT tracker generally fails since there are abrupt motion and/or occlusion of inlier points in these two sequences. Compared to graph/hyper-graph matching methods, our method gives better results, which confirms the effectiveness of ROML for simultaneous multi-image object matching. We also compare with our previous method (Zeng et al. 2012) in Table 10. Con-

Footnote 8 continued

of ℓ_2 -norm, compared to those in the other $K - 1$ frames. This trick is important since when applying ROML to a video sequence with static background, it is possible that some of the interest points in the background are selected to form a “low-rank pattern”, while those of the true pattern labelled in the first frame are treated as outliers. Theoretical correctness of this scheme is pending for proof. In practice, we observed that it worked well, and we defer the proof of this scheme in future research.

sistent to those results in Sect. 5.2, ROML again improves over Prev (Zeng et al. 2012) on these two sequences.

6 Common Object Localization

Learning models of object categories typically requires manually labelling a large amount of training images (e.g., up to a bounding box of the object of interest), which however, are expensive to obtain and may also suffer from unintended biases by annotators. A recently emerging research topic (Tuytelaars et al. 2010) considers automatically discovering and learning object models from a collection of unlabelled images. Given an image collection containing object instances belonging to unknown categories, the task is to identify the categories, localize object instances in images, and learn models for them so that the learned models can be applied to novel images for object detection. This is a weakly supervised (or unsupervised) learning scenario when the image collection is known to contain object instances of a single category (or multiple categories), which is in general ill-posed. A critical component for success of learning is precise object localization inside each image. However, precise *common object localization* (COL) is extremely difficult given unknown object categories/models, and also large intra-category variations and cluttered background.

Many methods have been proposed for this challenging task in either weakly supervised or unsupervised settings (Kim et al. 2008; Lee and Grauman 2009; Liu and Chen 2007; Russell et al. 2006). Among them the methods (Lee and Grauman 2009; Kim et al. 2008; Liu and Chen 2007) explicitly take object (or its associated parts/features) localization into account. These methods normally require the objects of interest covering a large portion of the images. More recently, saliency guided object learning techniques (Zhu et al. 2012; Deselaers et al. 2012) are proposed, which exploit generic knowledge of “objectness” (Alexe et al. 2010; Endres and Hoiem 2010; Feng et al. 2011) obtained from low-level image cues and/or learning from other irrelevant annotated images. Consequently, they can potentially locate object instances with large scale/appearance variations in cluttered background.

In this section, we present experiments to show how ROML can be applied to this COL task using local region descriptors as features. Similar to Deselaers et al. (2012), we also sample candidate bounding boxes from each image based on their objectness scores, and use appropriate region descriptors to characterize the appearance inside each bounding box. We then optimize (5) to select a bounding box from each image, i.e., $n = 1$ for the PPMs to be optimized. Ideally the selected bounding boxes should localize object instances deemed common in the given image collection, i.e., the matrix \mathbf{L} in (5) is rank deficient. We used the PAS-

CAL datasets (Everingham et al. 2006, 2007) for the COL experiments in both weakly supervised and unsupervised settings. For the weakly supervised case, we followed the same settings as in Deselaers et al. (2012). In particular, we used a subset of the PASCAL06 (Everingham et al. 2006) train + val dataset containing all images of 6 classes (bicycle, car, cow, horse, motorbike, sheep) from the left and right viewpoints. We conducted COL on all images of each class-viewpoint combination, which are assumed to contain object instances of the same class at a similar viewpoint. To make the problem better defined, we followed (Deselaers et al. 2012) and removed images in which all objects are marked as difficult or truncated in the ground truth annotation. The PASCAL07 dataset (Everingham et al. 2007) is more challenging as objects vary greatly in appearance, scale, and location. We also used 6 classes (aeroplane, bicycle, boat, bus, horse, and motorbike) of the PASCAL07 train + val dataset from the left and right viewpoints. The other settings were the same as for the PASCAL06 dataset. These classes of PASCAL06 and PASCAL07 datasets were chosen because they are the object classes on which fully supervised methods can perform reasonably well. For every image in one class-viewpoint combination, we used (Alexe et al. 2010) to sample 100 bounding boxes proportionally to their probability of containing an object (the objectness score). To describe the region appearance inside each bounding box, we used the GIST descriptor with the default parameters as in Oliva and Torralba (2001), which gives a 512-dimensional feature vector. As suggested in Deselaers et al. (2012), the shape and objectness score of a bounding box provide additional information that may help for COL. Let \mathbf{f} be the GIST descriptor vector for a bounding box. To use shape and objectness score, we first augmented \mathbf{f} with the aspect ratio (*width/height*) r of the bounding box, and then added perturbation noise $\mathbf{n} \in \mathbb{R}^{513}$ whose entries were drawn from normal distribution with standard deviation set as one minus objectness score of the bounding box. We used the thus produced vector $[\mathbf{f}^\top \kappa_r r]^\top + \kappa_n \mathbf{n}$ as the feature for each sampled bounding box, where κ_r and κ_n are weighting parameters. We set $\kappa_r = 0.08$ and $\kappa_n = 0.015$ for all the experiments reported in this section.

We measure COL performance by the percentage of correctly localized images out of all images in a class-viewpoint combination, where localization correctness in an image is based on PASCAL criteria, i.e., intersection of a bounding box with ground truth is more than half of their union. We compare with several baseline weakly supervised object localization and learning methods including MultiSeg (Russell et al. 2006) and Exemplar (Chum and Zisserman 2007), and also with WSL-GK (Deselaers et al. 2012), which is saliency guided and performs EM-like alternation of localizing objects and learning the object class model. In the preparation of this paper, we notice that a more recent work (Siva et al. 2013) gives better COL performance by using

Table 11 COL accuracies of different methods on the PASCAL06 and PASCAL07 datasets

	Objectness (Alexe et al. 2010) (%)	MultiSeg (Russell et al. 2006) (%)	Exemp. (Chum and Zisserman 2007) (%)	WSL-GK (Deselaers et al. 2012) (no learning) (%)	ROML (%)	WSL-GK (Deselaers et al. 2012) (with learning) (%)
PASCAL06	51	28	45	55	64	64
PASCAL07	28	22	33	37	36	50

For objectness (Alexe et al. 2010), sampled bounding box with the highest score in each image is considered as the estimated localization. Best results are highlighted in bold.

Table 12 COL accuracies of ROML for different class-viewpoint combinations of the PASCAL06 and PASCAL07 datasets in both weakly supervised and unsupervised settings

	PASCAL06				PASCAL07			
	Bicycle (%)	Car (%)	Cow (%)	Sheep (%)	Aeroplane (%)	Bus (%)	Horse (%)	Motorbike (%)
Weakly supervised—left	84	79	60	58	26	24	40	56
Unsupervised—left	80	79	60	52	30	29	35	56
Weakly supervised—right	69	70	66	52	38	61	35	65
Unsupervised—right	67	63	57	40	28	48	41	56

more advanced saliency estimation method. Since this section is mainly to show the usefulness of ROML for the COL task, we will not pursue adopting this new saliency method to further improve our results.

Table 11 reports COL accuracies of different methods on the PASCAL06 and PASCAL07 datasets, which are obtained by averaging over all class-viewpoint combinations. Results of MultiSeg (Russell et al. 2006) and Exemplar (Chum and Zisserman 2007) in Table 11 are from Deselaers et al. (2012). Table 11 suggests that Objectness (Alexe et al. 2010) gives very good initial candidates of object bounding boxes. Consequently, results of both our method and WSL-GK (Deselaers et al. 2012) on the PASCAL06 and PASCAL07 datasets compare favorably with those from MultiSeg (Russell et al. 2006) and Exemplar (Chum and Zisserman 2007). For the PASCAL07 dataset, our method is comparable to WSL-GK (Deselaers et al. 2012) when no iterative steps of class learning are performed in Deselaers et al. (2012), and greatly outperforms (Deselaers et al. 2012) for the PASCAL06 dataset, for which our result in fact approaches final result of Deselaers et al. (2012), which is obtained after full steps of class learning and using richer feature representation including GIST, color information, and HOG for object shapes. Since the present paper is focusing on object matching and localization, we defer extension of our method for object class learning as future research.

We also conducted COL experiments in the unsupervised setting using 4 classes from the PASCAL06 (bicycle, car, cow, and sheep) and PASCAL07 (aeroplane, bus, horse, and motorbike) datasets respectively. Other data setups were the same as those in the above weakly supervised COL experiments. For either of the PASCAL06 and PASCAL07 datasets, we put all images of different classes from one viewpoint as

an image collection, and applied ROML for object localization. Performance was again measured by the percentage of correctly localized images out of all images in a class-viewpoint combination. Table 12 reports detailed results of different class-viewpoint combinations, where we also list results of ROML in the weakly supervised setting. Table 12 tells that ROML performs consistently well in both weakly supervised and unsupervised object localization. Example images of these classes with localized bounding boxes are shown in Fig. 8, where we also show the bounding boxes with the highest objectness score in each image and those of ROML in weakly supervised setting for comparison.

7 Conclusions

In this paper, we propose a framework termed ROML, for robustly matching objects in a set of images. ROML is formulated as a rank and sparsity minimization problem to optimize a set of PPMs. The optimized PPMs identify inlier features from each image and establish their consistent correspondences across the image set. To solve ROML, we use the ADMM method, in which a subproblem associated with PPM updating is a difficult IQP. We prove that under widely applicable conditions, this IQP is equivalent to a formulation of LSAP, which can be efficiently solved by the Hungarian algorithm. Extensive experiments on rigid/non-rigid object matching, matching instances of a common object category, and common object localization show the efficacy of our proposed method.

In the present work, we have assumed for ROML that there is exactly one object instance contained in each of a set of images. This assumption is mainly to make an ideal

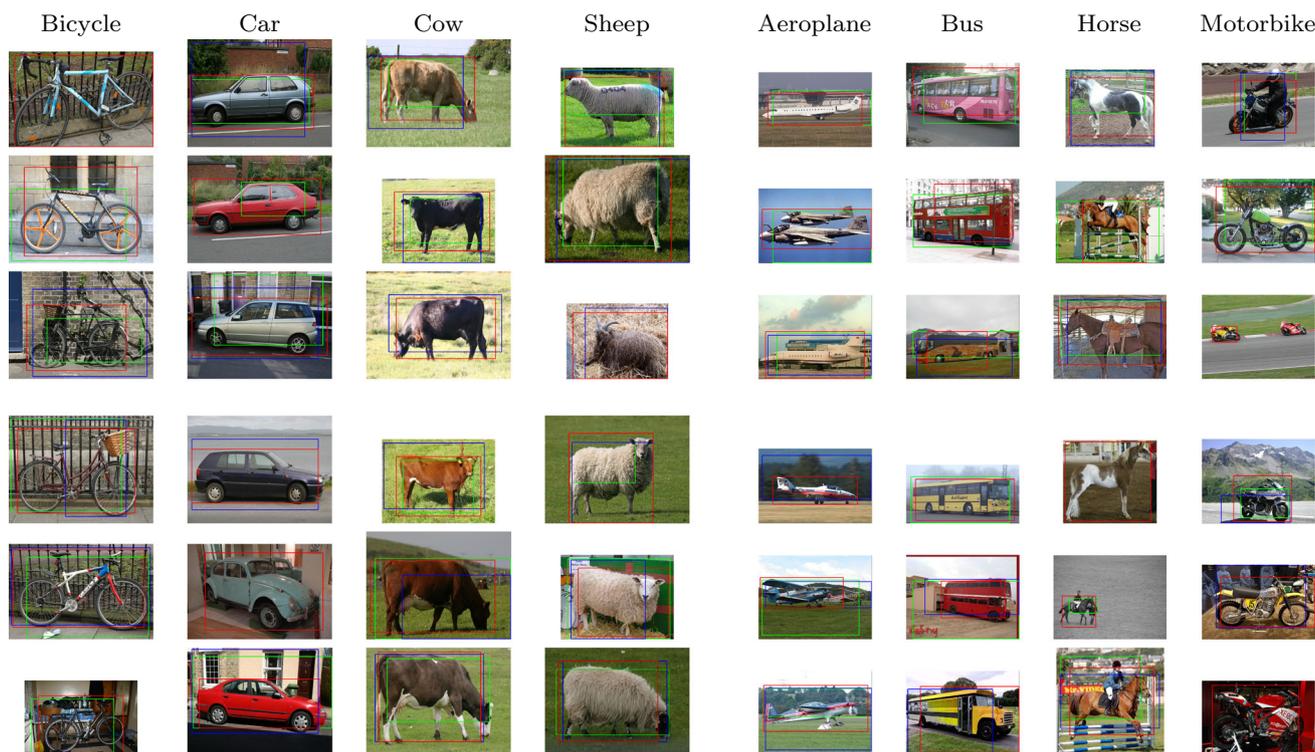


Fig. 8 Example images with estimated bounding boxes of different class-viewpoint combinations from the PASCAL06 and PASCAL07 datasets. In each image, results of Objectness (Alexe et al. 2010) with the highest objectness score (green box), and ROML in unsupervised

(blue box) and weakly supervised (red box) settings are shown (they may coincide in some images where only one or two boxes are shown). Top part is for left viewpoint, and bottom part is for right viewpoint (Color figure online)

problem setting for the difficult, combinatorial task of simultaneously matching inlier features of object instances across a set of images. However, in many practical problems such as unsupervised learning of object categories from an image collection, there could be multiple object instances, possibly of different categories, contained in one image. In these more challenging scenarios, ROML, by design, may at most identify and match one instance from each image, and ignore the other instances contained in the image collection. In order to learn object categories in these more challenging scenarios, one may need to extend the formulation of ROML so that multiple instances per image can be taken into account. For example, we have made an attempt of such kind in Zeng et al. (2013) for the task of learning object (face) categories from ambiguously labeled images, where each image in an image collection may contain multiple object instances of interest, and its associated caption has some labels of object category, with the true ones included, while the instance-label association is unknown. We take into account multiple instances per image in Zeng et al. (2013) by extending ROML to accommodate category-wise low-rank models and new constraints of PPMs. Nevertheless, extending ROML to unsupervised object learning remains an open question, and we are interested in pursuing this direction in future research.

Results of practical running time reported in Tables 1 and 6 suggest that it would be prohibitively slow to apply ROML (via Algorithm 1) to large-scale feature matching problems. Since most of the computation of Algorithm 1 is spent on solving matrix SVD (step 2) and K independent LSAPs (steps 5–7), one may revise Algorithm 1 accordingly, e.g., by using more efficient or approximate SVD solvers, to address the issue of scalability. Approximate SVD can be efficiently implemented using power method (Halko et al. 2011), which is based on a few iterations of QR factorization on a much reduced matrix. Using Hungarian algorithm to solve LSAP is already very efficient. To reduce the computation of steps 5–7, one may customize a stochastic version of ADMM algorithm for the proposed ROML formulation, and choose either approximate SVD or incremental SVD to alleviate the heavy computation of matrix SVD involved in each iteration of stochastic ADMM. While the per-iteration computation of stochastic ADMM becomes light, a remaining issue to be addressed is that the convergence rate of stochastic ADMM may not be comparable with its batch version, and consequently one may not necessarily get an improved overall efficiency by using stochastic ADMM to solve ROML. Alternatively, one may follow the low-rank matrix factorization based online RPCA algorithm (Feng et al. 2013), and

develop a corresponding online algorithm to solve ROML efficiently. By doing so one may need to address an additional issue of convergence, due to the discrete, non-convex problem nature of ROML. In future research, we are interested in pursuing these directions to develop new algorithms and applying ROML to large-scale feature matching problems.

Acknowledgments This work is partially supported by the National Natural Science Foundation of China (Grant No. 61202158), and the research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center from Singapore’s Agency for Science, Technology and Research (A*STAR).

Appendix 1: Learning Features of Coordinates-Descriptor Combination

Local region descriptors alone could be ambiguous for feature matching when there exist repetitive textures or less discriminative local appearance in images. To improve the matching accuracy, it is necessary to exploit the geometric structure of inlier points that consistently appears in each of the set of images. In this work, we consider a simple method introduced in [Torki and Elgammal \(2010\)](#) to exploit such geometric constraints. The method derives an embedded feature representation that combines information of both spatial arrangement of feature points inside each image, and similarity of feature descriptors across images. We briefly summarize this method as follows.

Given a set of K images, denote $\mathbf{A}_{spa}^k \in \mathbb{R}^{n_k \times n_k}$ as an affinity matrix that measures the spatial proximity of any two of the n_k extracted feature points in the k th image, where spatial proximity can be either measured based on Euclidean distances of image coordinates of feature points, which is invariant to translation and rotation, or made affine invariant ([Torki and Elgammal 2010](#)). In this work, we compute \mathbf{A}_{spa}^k using Gaussian kernel as $\mathbf{A}_{spa}^k(i, j) = e^{-\|\mathbf{x}_i^k - \mathbf{x}_j^k\|^2 / 2\sigma_{spa}^2}$, where $\mathbf{x}^k = [x^k, y^k]^T$ denotes image coordinates in the k th image, and σ_{spa} is a scaling parameter. Each feature point has an associated region descriptor. Denote $\mathbf{A}_{des}^{pq} \in \mathbb{R}^{n_p \times n_q}$ as another affinity matrix, each entry of which measures the similarity of region descriptors between a pair of features selected from the p th and q th images respectively. \mathbf{A}_{des}^{pq} can be computed similar to \mathbf{A}_{spa}^k as $\mathbf{A}_{des}^{pq}(i, j) = e^{-\|\mathbf{f}_i^p - \mathbf{f}_j^q\|^2 / 2\sigma_{des}^2}$, where \mathbf{f}_i^p and \mathbf{f}_j^q are feature descriptors from the p th and q th images respectively, and σ_{des} is a scaling parameter.

The method in [Torki and Elgammal \(2010\)](#) aims to learn embedded feature representations for all $N = \sum_{k=1}^K n_k$ points in the K images so that in the embedded space: (1) spatial structure of the point set in each image should be preserved; (2) features from different images with high descriptor similarity should be close to each other. Let $\{\mathbf{f}_i^k \in$

$\mathbb{R}^d\}_{i=1}^{n_k}, k = 1, \dots, K$, be the new features to be learned,⁹ the above objectives can be formalized as

$$\min \sum_{p,q} \sum_{i,j} \|\mathbf{f}_i^p - \mathbf{f}_j^q\|_2^2 \mathbf{A}_{ij}^{pq}, \tag{28}$$

where the matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is defined as: $\mathbf{A}^{pq} = \mathbf{A}_{spa}^k$ when $p = q = k$, $\mathbf{A}^{pq} = \mathbf{A}_{des}^{pq}$ when $p \neq q$, and $\mathbf{A}^{pq} \in \mathbb{R}^{n_p \times n_q}$ is the (p, q) block of all the $K \times K$ blocks of \mathbf{A} . The objective function (28) turns to be a problem of Laplacian embedding ([Belkin and Niyogi 2003](#)). Let $\tilde{\mathbf{F}} = [\mathbf{f}_1^1, \dots, \mathbf{f}_{n_1}^1, \dots, \mathbf{f}_1^K, \dots, \mathbf{f}_{n_K}^K]^T \in \mathbb{R}^{N \times d}$, (28) can be rewritten in matrix form as

$$\min_{\tilde{\mathbf{F}}} \text{trace}(\tilde{\mathbf{F}}^T \tilde{\mathbf{L}}_A \tilde{\mathbf{F}}) \text{ s.t. } \tilde{\mathbf{F}}^T \tilde{\mathbf{D}}_A \tilde{\mathbf{F}} = \mathbf{I}, \tag{29}$$

where $\tilde{\mathbf{L}}_A = \tilde{\mathbf{D}}_A - \mathbf{A}$ is the Laplacian matrix of \mathbf{A} , and $\tilde{\mathbf{D}}_A$ is a diagonal matrix with value of the i th diagonal entry as $\sum_j \mathbf{A}_{ij}$. (29) is a generalized eigenvector problem: $\tilde{\mathbf{L}}_A \mathbf{f} = \beta \tilde{\mathbf{D}}_A \mathbf{f}$. Its optimal solution, i.e., the N new features in the d -dimensional embedded space, can be obtained by the bottom d nonzero eigenvectors.

Appendix 2

We present derivations of the solutions (11) and (12) respectively for the problems (7) and (8) as follows.

Given updated variables \mathbf{E}_t , $\{\mathbf{P}_t^k\}_{k=1}^K$, and \mathbf{Y}_t , and write $\mathbf{D}_t = [\text{vec}(\mathbf{F}^1 \mathbf{P}_t^1), \dots, \text{vec}(\mathbf{F}^K \mathbf{P}_t^K)]$, the problem (7) can be written explicitly as

$$\min_{\mathbf{L}} \|\mathbf{L}\|_* + \frac{\rho}{2} \|\mathbf{L} - (\mathbf{D}_t - \mathbf{E}_t - \frac{1}{\rho} \mathbf{Y}_t)\|_F^2, \tag{30}$$

which appears to be the form of a proximal operator associated with the nuclear norm. According to [Lin et al. \(2010\)](#), optimal solution of (30) can be written as (11).

Given updated variables \mathbf{L}_{t+1} , \mathbf{Y}_t , and $\{\mathbf{P}_t^k\}_{k=1}^K$ with $\mathbf{D}_t = [\text{vec}(\mathbf{F}^1 \mathbf{P}_t^1), \dots, \text{vec}(\mathbf{F}^K \mathbf{P}_t^K)]$, the problem (8) can be written explicitly as

$$\min_{\mathbf{E}} \lambda \|\mathbf{E}\|_1 + \frac{\rho}{2} \|\mathbf{E} - (\mathbf{D}_t - \mathbf{L}_{t+1} - \frac{1}{\rho} \mathbf{Y}_t)\|_F^2, \tag{31}$$

which appears to be the form of a proximal operator associated with the ℓ_1 -norm. According to [Lin et al. \(2010\)](#), optimal solution of (31) can be written as (12).

⁹ For consistency we use the same \mathbf{f} for different feature types.

References

- Alexe, B., Deselaers, T., & Ferrari, V. (2010). What is an object? In *CVPR* (pp. 73–80).
- Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6), 1373–1396.
- Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transaction on PAMI*, 24(4), 509–522.
- Berg, A. C., & Malik, J. (2001). Geometric blur for template matching. In *CVPR* (Vol. 1, pp. 1–607).
- Berg, A. C., Berg, T. L., & Malik, J. (2005). Shape matching and object recognition using low distortion correspondences. In *CVPR* (Vol. 1, pp. 26–33).
- Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific.
- Bertsekas, D. P., & Tsitsiklis, J. N. (1989). *Parallel and distributed computation: Numerical methods*. Englewood Cliffs, NJ: Prentice-Hall, Inc.
- Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transaction on PAMI*, 14(2), 239–256.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 1–122.
- Brox, T., & Malik, J. (2010). Object segmentation by long term analysis of point trajectories. In: *ECCV* (pp. 282–295).
- Burkard, R. E., Dell'Amico, M., & Martello, S. (2009). *Assignment Problems*. Society for Industrial and Applied Mathematics.
- Caetano, T. S., McAuley, J. J., Cheng, L., Le, Q. V., & Smola, A. J. (2009). Learning graph matching. *IEEE Transaction on PAMI*, 31(6), 1048–1058.
- Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM*, 58(3), 11.
- Cho, M., Lee, J., & Lee, K. M. (2010). Reweighted random walks for graph matching. In *ECCV* (pp. 492–505).
- Chui, H., & Rangarajan, A. (2003). A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2), 114–141.
- Chum, O., & Zisserman, A. (2007). An exemplar model for learning object classes. In *CVPR* (pp. 1–8).
- Collins, R. T. (2012). Multitarget data association with higher-order motion models. In *CVPR* (pp. 1744–1751).
- Conte, D., Foggia, P., Sansone, C., & Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03), 265–298.
- Cour, T., Srinivasan, P., & Shi, J. (2006). Balanced graph matching. In *NIPS*.
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR* (Vol. 1, pp. 886–893).
- Deselaers, T., Alexe, B., & Ferrari, V. (2012). Weakly supervised localization and learning with generic knowledge. *IJCV*, 100(3), 275–293.
- Duchenne, O., Bach, F., Kweon, I. S., & Ponce, J. (2011). A tensor-based algorithm for high-order graph matching. *IEEE Transaction on PAMI*, 33(12), 2383–2395.
- Endres, I., & Hoiem, D. (2010). Category independent object proposals. In *ECCV* (pp. 575–588).
- Everingham, M., Van Gool, L., Williams, C., & Zisserman, A. (2006). The pascal visual object classes challenge.
- Everingham, M., Van Gool, L., Williams, C., Winn, L., & Zisserman, A. (2007). The pascal visual object classes challenge.
- Fei-Fei, L., Fergus, R., & Perona, P. (2007). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1), 59–70.
- Feng, J., Wei, Y., Tao, L., Zhang, C., & Sun, J. (2011). Salient object detection by composition. In *ICCV* (pp. 1028–1035).
- Feng, J., Xu, H., & Yan, S. (2013). Online robust pca via stochastic optimization. *Advances in Neural Information Processing Systems*, 26, 404–412.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Goldfarb, D., Ma, S., & Scheinberg, K. (2013). Fast alternating linearization methods for minimizing the sum of two convex functions. *Mathematical Programming*, 141(1–2), 349–382.
- Halko, N., Martinsson, P. G., & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2), 217–288.
- Jiang, H., & Yu, S. (2009). Linear solution to scale and rotation invariant object matching. In *CVPR* (pp. 2474–2481).
- Kim, G., Faloutsos, C., & Hebert, M. (2008). Unsupervised modeling of object categories using link analysis techniques. In *CVPR* (pp. 1–8).
- Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1–2), 83–97.
- Lee, J., Cho, M., & Lee, K. M. (2011). Hyper-graph matching via reweighted random walks. In *CVPR* (pp. 1633–1640).
- Lee, Y. J., & Grauman, K. (2009). Foreground focus: Unsupervised learning from partially matching images. *IJCV*, 85(2), 143–166.
- Leordeanu, M., & Hebert, M. (2005). A spectral technique for correspondence problems using pairwise constraints. In *ICCV* (Vol. 2, pp. 1482–1489).
- Leordeanu, M., Hebert, M., & Sukthankar, R. (2009). An integer projected fixed point method for graph matching and map inference. In *NIPS* (Vol. 2, p. 7).
- Li, H., Huang, J., Zhang, S., & Huang, X. (2011). Optimal object matching via convexification and composition. In *ICCV* (pp. 33–40).
- Lin, Z., Chen, M., & Ma, Y. (2010). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint [arXiv:1009.5055](https://arxiv.org/abs/1009.5055).
- Liu, D., & Chen, T. (2007). Unsupervised image categorization and object localization using topic models and correspondences between images. In *ICCV* (pp. 1–7).
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *ICCV* (Vol. 2, pp. 1150–1157).
- Lucas, B. D., Kanade, T., et al. (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI* (Vol. 81, pp. 674–679).
- Maciel, J., & Costeira, J. P. (2003). A global solution to sparse correspondence problems. *IEEE Transaction on PAMI*, 25(2), 187–199.
- Mikolajczyk, K., & Schmid, C. (2004). Scale & affine invariant interest point detectors. *IJCV*, 60(1), 63–86.
- Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3), 145–175.
- Oliveira, R., Costeira, J., & Xavier, J. (2005). Optimal point correspondence through the use of rank constraints. In *CVPR* (Vol. 2, pp. 1016–1021).
- Oliveira, R., Ferreira, R., & Costeira, J. P. (2006). Optimal multi-frame correspondence with assignment tensors. In *ECCV* (pp. 490–501).
- Pachauri, D., Kondor, R., & Singh, V. (2013). Solving the multi-way matching problem by permutation synchronization. In *NIPS* (pp. 1860–1868).
- Poore, A. B., & Robertson, A. J. I. I. (1997). A new lagrangian relaxation based algorithm for a class of multidimensional assignment

- problems. *Computational Optimization and Applications*, 8(2), 129–150.
- Robertson, A. J. (2001). A set of greedy randomized adaptive local search procedure (grasp) implementations for the multidimensional assignment problem. *Computational Optimization and Applications*, 19(2), 145–164.
- Russell, B. C., Freeman, W. T., Efros, A. A., Sivic, J., & Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *CVPR* (Vol. 2, pp. 1605–1614).
- Scott, G. L., & Longuet-Higgins, H. C. (1991). An algorithm for associating the features of two images. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 244(1309), 21–26.
- Serre, T., Wolf, L., & Poggio, T. (2005). Object recognition with features inspired by visual cortex. In *CVPR* (Vol. 2, pp. 994–1000).
- Shapiro, L. S., & Michael Brady, J. (1992). Feature-based correspondence: An eigenvector approach. *Image and Vision Computing*, 10(5), 283–288.
- Shen, Y., Wen, Z., & Zhang, Y. (2014). Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optimization Methods and Software*, 29(2), 239–263.
- Siva, P., Russell, C., Xiang, T., & Agapito, L. (2013). Looking beyond the image: Unsupervised learning for object saliency and detection. In *CVPR* (pp. 3238–3245).
- Tomasi, C., & Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2), 137–154.
- Torki, M. & Elgammal, A. (2010). One-shot multi-set non-rigid feature-spatial matching. In *CVPR* (pp. 3058–3065).
- Torresani, L., Kolmogorov, V., & Rother, C. (2008). Feature correspondence via graph matching: Models and global optimization. In *ECCV* (pp. 596–609).
- Tuytelaars, T., Lampert, C. H., Blaschko, M. B., & Buntine, W. (2010). Unsupervised object discovery: A comparison. *IJCV*, 88(2), 284–302.
- Zass, R. & Shashua, A. (2008). Probabilistic graph and hypergraph matching. In *CVPR* (pp. 1–8).
- Zeng, Z., Chan, T. H., Jia, K., & Xu, D. (2012). Finding correspondence from multiple images via sparse and low-rank decomposition. In *ECCV* (pp. 325–339).
- Zeng, Z., Xiao, S., Jia, K., Chan, T. H., Gao, S., Xu, D., & Ma, Y. (2013). Learning by associating ambiguously labeled images. In *CVPR* (pp. 708–715).
- Zhang, Y. (2010). *An alternating direction algorithm for nonnegative matrix factorization*. Rice Univ. CAAM Tech. Rep.
- Zhou, F. & De la Torre, F. (2012). Factorized graph matching. In *CVPR* (pp. 127–134).
- Zhu, J. Y., Wu, J., Wei, Y., Chang, E., & Tu, Z. (2012). Unsupervised object class discovery via saliency-guided multiple class learning. In *CVPR* (pp. 3218–3225).