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

Kui Jia, Tsung-Han Chan, Zinan Zeng, and Yi Ma

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 inliers and establish *consistent* feature correspondences across the set of images. This is a challenging combinatorial problem, and the problem complexity grows exponentially with the image number. To this end, we propose a novel framework, called ROML, to address this problem. ROML optimizes simultaneously a partial permutation matrix (PPM) for each image, and feature correspondences are established using the obtained PPMs. Our contributions are summarized as follows: (1) We formulate the problem as rank and sparsity minimization for optimizing PPMs. (2) We use the ADMM method to solve the so formulated problem, in which a subproblem associated with PPM optimization is a difficult integer quadratic program (IQP). We prove that under wildly applicable conditions, this IQP is equivalent to a linear program over a compact convex set, for which an exact solution is able to be attained. (3) Our framework is independent of the domain of applications and type of features. (4) Extensive experiments on the applications of robust object matching and common object localization demonstrate the superiority of our method over existing ones.

Index Terms—Object matching, Feature correspondence, Common object localization, Low-rank, Sparsity.

1 Introduction

Object matching is a fundamental problem in computer vision. Given a pair or a set of images that contain common 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 variation 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 [26], [27]. 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 [28]. In [30], [31], [28], 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. The development of local invariant features [13], [16] for discriminative description of visual appearance has brought significant progress in object matching and recognition [17]. For example in [42], [45], 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 [2], [5] 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 existing 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 search problem. As the number of images increases, the complexity of finding such a global matching 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 information or constraints from other images that also contain the same object pattern of interest. It is unclear either how existing pair-wise matching methods can be generalized to address this global matching task. 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 appearing in each image. The inlier features describe local salient appearance of the object, and the rest of the features are outliers. Some inlier features could also be missing due to occlusion or mis-detection. Depending on applications, the types of features can be chosen as image coordinates, local region descriptors [13], [15], [14], or combination of them [23]. Under such a setting, each image is naturally represented as a set of feature vectors. If we can re-order the correctly identified inlier features from each image, and concatenate them as a (long) vector, ideally the matrix formed by horizontally arraying these vectors should be low-rank [1]. The underlying rationale is that inlier features, the corresponding ones of which are visually correlated to each other, repetitively appear in the image set, while outlier features just accidently do so. In situations when there are variations in inlier features of different images (e.g., due to illumination or pose changes), or when some inlier features are missing, we can decompose this matrix as a combination of low-rank term and sparse term, where the sparse term models those variations or missing inliers.

Motivated by these observations, we propose in this paper a novel and principled framework, termed ROML, for robustly matching objects in a set of images. ROML is formulated to optimize a partial permutation matrix (PPM) (cf. (1)) for each image by rank and sparsity minimization. The PPM simultaneously selects inlier features from each image and re-order them, so that after selection and re-ordering, the matrix formed by these feature vectors can be decomposed into a low-rank matrix and a sparse matrix. We solve ROML using the ADMM method [25], in which the subproblem associated with PPM optimization is a difficult integer quadratic programming (IQP) problem. We prove that under widely applicable conditions, this IQP is *equivalent* to a linear program over a compact convex set, for which an exact solution is able to be attained. Our method is based on a unified framework, independent of the domain of applications and type of features. Extensive experiments on robust object matching and common object localization show the superiority of our method over existing ones.

A preliminary work of this paper has appeared in [53]. In the present paper, we have made significant improvement over [53] in the following aspects. In addition, we have also completely rewritten the paper to present our ideas more clearly.

- Although [53] proposes to optimize a set of PPMs via rank and sparsity minimization for robust feature matching, however, its solution is attained by sequentially solving two costly sub-problems: a quadratic program over the continuous-domain relaxation of PPMs, followed by a binary integer programming that projects each relaxed PPM into its feasible set. In fact, the second subproblem is irrelevant to the original objective function, and consequently, the thus obtained PPMs and feature correspondences are only suboptimal solutions. In contrast, we propose in the present paper a new method to solve the PPM optimization and prove that under wildly applicable conditions, the PPM optimization step is equivalent to a linear program over a compact convex set, which is much easier to be solved to an exact solution.
- For matching features from multiple images, [53] considers similarity of feature descriptors only, neglecting the exploitation of spatial relations between feature points in each image, which could be very important for reliable matching when feature descriptors alone become less distinguishable between each other, thus limiting its effectiveness in many practical scenarios. In fact, only toy feature matching experiments on face images are presented in [53]. Instead, we show in this paper that our proposed formulation works with a variety of feature types including image coordinates, feature descriptors, or combination of them (cf. Section 4).
- We apply our method on diverse applications such as rigid/non-rigid object matching, matching object instances of a same category, and localizing common objects in a set of images in both weakly supervised and unsupervised settings. Compared to existing

[1] The form of low-rank matrix using image coordinates as features is different, which we will present in Section 4.1.

methods, we achieve the best performance in most of these applications.

The remainder of this paper is organized as follows.

2 Related Works

There is an intensive literature on object or shape matching between a pair of images. Representative works include shape context [28], graph matching [2], [5], and hyper-graph matching [8], [9], [6]. A comprehensive review of these methods can be found in [29]. In this section, we will focus on discussion of several existing methods that also use multiple images/point sets for object matching, and are thus more related to our proposed method.

Maciel and Costeira [18] first proposed to use PPM to model both feature correspondence 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 nonpolynomial. Matching criteria used in the cost function of [18] was based on pair-wise similarity of features in different images. Thus [18] is a local, unscalable method, inherently limiting its applicability in matching of a small number of images only. 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 our method is able to globally and robustly match corresponding features in a large set of images.

Rank constraints have been used in [20], [21] for matching of feature points in frames of a video sequence, for which they constructed a measurement matrix containing image coordinates of points extracted from a moving rigid object. Motivated by factorization model in shape-from-motion [22], they assumed this measurement matrix was low-rank, and used rank constraints to optimize PPMs for establishing point correspondences across frames. The method in [20] 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, i.e., partial occlusion of objects is not permitted; (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) in their rank minimization algorithm, an initial rough estimate of point correspondences for a new frame is assumed given, which may be only valid for slow motion objects. The aspects (1) and (3) have to some extent been alleviated in [21], but [21] cannot cope with the other limiting aspects. As a globally consistent and robust matching framework, our method has no such limitations. More importantly, we want to note that the machanism of rank constraints used in [20], [21] is different from that of our method. Methods [20], [21] 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 takes image coordinates as an instance, and also includes other features describing visual appearance information. 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 [23] for feature learning and correspondence. Given feature point sets extracted from a set of images, it can learn a unified feature space, which encodes information of both region descriptors and the geometric structure of each point set. [23] used k-means clustering in the embedded space for feature matching. However, clusters generated by k-means, i.e., groups of corresponded features, are only locally optimal matching solutions. And there is no explicit outlier rejection mechanism in [23] either. Compared to [23], our method uses the low-rank and sparse constraints to optimize PPMs, which integrates correspondence and outlier rejection in a single step. Due to the sparse constraint, our method is also more robust to data corruption and occlusion of inlier features, thus potentially able to get more accurate matching results.

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 either be image coordinates of the feature point, or region descriptors such as SIFT [13] that describe local appearance information. It can also be some form of low-dimensional embedding that encodes both local appearance and spatially structural information of any feature point relative to other points in the image [23]. In spite of these multiple choices, for now we just generally refer to them as *features*. Discussion of different feature types and their applicable spectrums will be presented in Section 4. These n_k features can be categorized as inliers or outliers. Without loss of generality we assume at this moment that there exist an equal number of n inlier features in each of the K images, where $n \leq n_k$ for $k \in \{1, \ldots, K\}$. We will discuss the cases of missing inlier features 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 accidently do so in a random, unstructured way. Our formulation for object matching is essentially motivated by these observations. Concretely, denote $\overline{\mathbf{F}}^k = [\mathbf{f}_1^k, \dots, \mathbf{f}_n^k] \in \mathbb{R}^{d \times n}$ as the matrix formed by inlier feature vectors in the k^{th} image, so defined are the matrices $\{\overline{\mathbf{F}}^1, \dots, \overline{\mathbf{F}}^K\}$ for all the K images. Assume column vectors in each of these matrices are arrayed in the same order, i.e., inlier features in $\{\overline{\mathbf{F}}^1, \dots, \overline{\mathbf{F}}^K\}$ are respectively corresponded, then the matrix formed by $\overline{\mathbf{D}} = [\mathrm{vec}(\overline{\mathbf{F}}^1)| \cdots |\mathrm{vec}(\overline{\mathbf{F}}^K)] \in \mathbb{R}^{dn \times K}$ will be approximately low-rank, ideally rank one, where $\mathrm{vec}(\cdot)$ 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

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

Thus, there exist (*but not unique*) 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

$$\mathbf{D} = [\mathrm{vec}(\mathbf{F}^1\mathbf{P}^1)| \cdots |\mathrm{vec}(\mathbf{F}^K\mathbf{P}^K)] \in \mathbb{R}^{dn \times K} \quad (2)$$

is rank deficient. Based on this low-rank assumption, feature correspondence can thus be formulated as the following problem to optimize $\{\mathbf{P}^k\}_{k=1}^K$

$$\min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K, \mathbf{L}} \mathrm{rank}(\mathbf{L}) \\ \text{s.t. } [\mathrm{vec}(\mathbf{F}^1\mathbf{P}^1)| \cdots |\mathrm{vec}(\mathbf{F}^K\mathbf{P}^K)] = \mathbf{L}. \quad (3)$$

In practice, however, inlier features characterizing the same local appearance information of object instances could be rather different 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 of the data matrix $\mathbf{D}$, where we assume these errors only appear in a small fraction of $\mathbf{D}$. The formulation (3) can then be modified as

$$\min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K, \mathbf{L}, \mathbf{E}} \mathrm{rank}(\mathbf{L}) + \lambda \|\mathbf{E}\|_0 \\ \text{s.t. } [\mathrm{vec}(\mathbf{F}^1\mathbf{P}^1)| \cdots |\mathrm{vec}(\mathbf{F}^K\mathbf{P}^K)] = \mathbf{L} + \mathbf{E}, \quad (4)$$

where $\|\cdot\|_0$ denotes the 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 $\mathrm{rank}(\cdot)$ 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 $\mathrm{rank}(\cdot)$ and $\|\cdot\|_0$ with their convex surrogates $\|\cdot\|_*$ and $\|\cdot\|_1$ respectively, where $\|\cdot\|_*$ denotes the nuclear norm or sum of the singular values, and $\|\cdot\|_1$ is the ℓ_1-norm. It has been shown in the related RPCA problem [54] [2] that when the rank of the matrix $\mathbf{L}$ to be recovered is not too high and the number of nonzero entries in $\mathbf{E}$ is not too large, this convention can exactly recover $\mathbf{L}$ in RPCA. Applying the same relaxation to (4) yields

$$\min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K, \mathbf{L}, \mathbf{E}} \|\mathbf{L}\|_* + \lambda \|\mathbf{E}\|_1 \\ \text{s.t. } [\mathrm{vec}(\mathbf{F}^1\mathbf{P}^1)| \cdots |\mathrm{vec}(\mathbf{F}^K\mathbf{P}^K)] = \mathbf{L} + \mathbf{E}, \\ \mathcal{P}^k = \{\mathbf{P}^k \in \{0,1\}^{n_k \times n} | \mathbf{1}_{n_k}^\top \mathbf{P}^k = \mathbf{1}_n^\top, \\ \mathbf{P}^k \mathbf{1}_n \leq \mathbf{1}_{n_k}\}, \; \forall \, k = 1, \dots, K, \quad (5)$$

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).

Optimization of (5) involves a large number of variables in matrices $\mathbf{L}$, $\mathbf{E}$, and $\{\mathbf{P}^k\}_{k=1}^K$, for which a distributed, scalable algorithm is desired. We thus consider using the Alternating Direction Method of Multipliers (ADMM) method [25], [55]. ADMM decomposes a large-scale global problem into small local subproblems that can be readily solved. It has shown its efficiency in related problems [56], [57] with a general form of $\min \sum_i \lambda_i \|\mathbf{X}_i\|_{(i)}$ s.t. $\sum_i \mathbf{X}_i = \mathbf{M}$, where $\mathbf{M}$ is a data matrix and $\|\cdot\|_{(i)}$ are some proper norms including $\|\cdot\|_*$ and $\|\cdot\|_1$ that encourage low-complexity structures. However, the difference here is that the subproblem associated with $\{\mathbf{P}^k\}_{k=1}^K$ concerns with nonlinear integer programming. It is essential to understand the efficacy and convergence properties of ADMM under these conditions, which we will discuss in Section 3.2.1 after presentation of our algorithmic procedure.

With $\mathbf{D} = [\mathrm{vec}(\mathbf{F}^1\mathbf{P}^1)| \cdots |\mathrm{vec}(\mathbf{F}^K\mathbf{P}^K)]$, the augmented Lagrangian for (5) can be written as

$$\mathcal{L}_\rho(\mathbf{L}, \mathbf{E}, \{\mathbf{P}^k\}_{k=1}^K, \mathbf{Y}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{E}\|_1 + \\ \langle \mathbf{Y}, \mathbf{L} + \mathbf{E} - \mathbf{D} \rangle + \frac{\rho}{2} \|\mathbf{L} + \mathbf{E} - \mathbf{D}\|_F^2, \quad (6)$$

[2] RPCA addresses the problem $\min \|\mathbf{L}\|_* + \lambda \|\mathbf{E}\|_1$ s.t. $\mathbf{D} = \mathbf{L} + \mathbf{E}$, given the data matrix $\mathbf{D}$.

where $\mathbf{Y} \in \mathbb{R}^{dn \times K}$ is a matrix of Lagrange multipliers, ρ is a positive scalar, $\langle \cdot, \cdot \rangle$ denotes the matrix inner product, 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. The constraints of $\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K$ will be enforced in the subproblem associated with updating of $\{\mathbf{P}^k\}_{k=1}^K$. More specifically, our ADMM procedure consists of the following iterations

$$\begin{aligned}
\mathbf{L}_{t+1} &= \arg\min_{\mathbf{L}} \mathcal{L}_\rho\big(\mathbf{L}, \mathbf{E}_t, \{\mathbf{P}_t^k\}_{k=1}^K, \mathbf{Y}_t\big), && (7)\\
\mathbf{E}_{t+1} &= \arg\min_{\mathbf{E}} \mathcal{L}_\rho\big(\mathbf{L}_{t+1}, \mathbf{E}, \{\mathbf{P}_t^k\}_{k=1}^K, \mathbf{Y}_t\big), && (8)\\
\{\mathbf{P}_{t+1}^k\}_{k=1}^K &= \arg \min_{\{\mathbf{P}^k \in \mathcal{P}^k\}_{k=1}^K} \mathcal{L}_\rho\big(\mathbf{L}_{t+1}, \mathbf{E}_{t+1}, \\
&\qquad\qquad \{\mathbf{P}^k\}_{k=1}^K, \mathbf{Y}_t\big), && (9)\\
\mathbf{Y}_{t+1} &= \mathbf{Y}_t + \rho\big(\mathbf{L}_{t+1} + \mathbf{E}_{t+1} - \mathbf{D}_{t+1}\big), && (10)
\end{aligned}$$

where t indicates the iteration number and we compute $\mathbf{D}_{t+1} = [\mathrm{vec}(\mathbf{F}^1\mathbf{P}_{t+1}^1)| \cdots |\mathrm{vec}(\mathbf{F}^K\mathbf{P}_{t+1}^K)]$ immediately after the step (9).

The steps (7) and (8) involving updating of $\mathbf{L}$ and $\mathbf{E}$ are both convex programs. In particular, they can be explicitly written as the forms of the proximal operator associated with a nuclear norm or the proximal operator associated with an ℓ_1-norm respectively. Each of them has a simple closed-form solution, and can thus be efficiently solved. To spell out the solutions, define the soft-thresholding or shrinkage operator for scalars as $\mathcal{T}_\tau[x] = \mathrm{sign}(x) \cdot \max\{|x| - \tau, 0\}$, with $\tau > 0$. When applied to vectors or matrices, it operates element-wisely. With the definition of shrinkage operator we can write the optimal solution to the problem (7) as

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

where $\mathrm{svd}(\cdot)$ denotes the Singular Value Decomposition operator. The optimal solution to the problem (8) is

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

Optimization of the problem (9) is more involved than (7) and (8), mostly because of the binary constraints enforced on the entries of PPMs $\{\mathbf{P}^k\}_{k=1}^K$. To solve (9), we first observe that (9) can be decoupled into K independent subproblems, each of which concerns with optimization of one of the matrices $\{\mathbf{P}^k\}_{k=1}^K$. For the k^{th} subproblem, denote $\theta^k = \mathrm{vec}(\mathbf{P}^k) \in \mathbb{R}^{nn_k}$, $\mathbf{G}^k = \mathbf{I}_n \otimes \mathbf{F}^k \in \mathbb{R}^{dn \times nn_k}$, $\mathbf{J}^k = \mathbf{I}_n \otimes \mathbf{1}_{n_k}^\top \in \mathbb{R}^{n \times nn_k}$, $\mathbf{H}^k = \mathbf{1}_n^\top \otimes \mathbf{I}_{n_k} \in \mathbb{R}^{n_k \times nn_k}$, $\otimes$ is the Kronecker product, and $\mathbf{I}_n$ (or $\mathbf{I}_{n_k}$) is the identity matrix of size $n \times n$ (or $n_k \times n_k$). The k^{th} subproblem can then be written, for updating of θ_{t+1}^k, as

$$\begin{aligned}
\min_{\theta^k} \frac{\rho}{2}\theta^{k\top}\mathbf{G}^{k\top}\mathbf{G}^k\theta^k - \mathbf{e}_k^\top\big[\mathbf{Y}_t^\top + \rho\big(\mathbf{L}_{t+1} + \mathbf{E}_{t+1}\big)^\top\big]\mathbf{G}^k\theta^k\\
\text{s.t. } \mathbf{J}^k\theta^k = \mathbf{1}_n,\ \mathbf{H}^k\theta^k \leq \mathbf{1}_{n_k},\ \theta^k \in \{0,1\}^{nn_k}, \quad (13)
\end{aligned}$$

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, and we have used the fact $\mathrm{vec}(\mathbf{XYZ}) = (\mathbf{Z}^\top \otimes \mathbf{X})\mathrm{vec}(\mathbf{Y})$ [19]. (13) appears to be a difficult integer constrained quadratic programming problem. To solve it, a common approach is to relax the constraint set of (13) into its convex hull, i.e., to constrain $\mathbf{P}^k$ as a row-wise substochastic matrix [18], and then project back the attained continuous-domain results, by either thresholding or other more complicated methods. However, the projection operation is not related to the original objective function of (13), and the thus obtained solution cannot guarantee to be optimal. For the present work of robust object matching, we prove that under widely applicable conditions, (13) is equivalent to a linear programming (LP) problem over a compact convex constraint set.

Theorem 1: Under the settings of object matching as stated in Problem 1, (13) is always equivalent to the following linear programming problem over a compact convex set

$$\begin{aligned}
\min_{\theta^k} -\mathbf{e}_k^\top\big[\mathbf{Y}_t^\top + \rho\big(\mathbf{L}_{t+1} + \mathbf{E}_{t+1}\big)^\top\big]\mathbf{G}^k\theta^k\\
\text{s.t. } \mathbf{J}^k\theta^k = \mathbf{1}_n,\ \mathbf{H}^k\theta^k \leq \mathbf{1}_{n_k},\ \theta^k \geq 0. \qquad (14)
\end{aligned}$$

Proof: We first show that under the settings of object matching considered in this paper, the objective function of (13) is equivalent to a linear function, as written in (14). Denote $\mathbf{p}_i^k \in \mathbb{R}^{n_k}$, $i = 1, \ldots, n$, as columns of PPM $\mathbf{P}^k$. From the definitions of $\mathbf{G}^k$ and θ^k, it is straightforward to show that

$$\mathbf{G}^k\theta^k = \mathrm{vec}(\mathbf{F}^k\mathbf{P}^k) = \begin{bmatrix} \mathbf{F}^k\mathbf{p}_1^k \\ \vdots \\ \mathbf{F}^k\mathbf{p}_n^k \end{bmatrix}. \qquad (15)$$

Since $\mathbf{F}^k = [\mathbf{f}_1^k, \ldots, \mathbf{f}_{n_k}^k] \in \mathbb{R}^{d \times n_k}$, and from the constraints of $\mathbf{P}^k$ (explicitly stated in (5)), it is clear that each subvector $\mathbf{F}^k\mathbf{p}_i^k$, $i = 1, \ldots, n$, of (15) selects one column feature vector from $\mathbf{F}^k$, with a unique index from the set $\{1, \ldots, n_k\}$. From (15) we also have

$$\theta^{k\top}\mathbf{G}^{k\top}\mathbf{G}^k\theta^k = \|\mathbf{G}^k\theta^k\|_2^2 = \sum_{i=1}^n \|\mathbf{F}^k\mathbf{p}_i^k\|_2^2. \qquad (16)$$

In case that there are no outliers, i.e., $n_k = n$, (16) is equal to a constant value no matter what feasible $\mathbf{P}^k$ or θ^k is used. In the more general case that there exist outliers, we can always normalize the column vectors $\mathbf{f}_i^k$, $i = 1, \ldots, n_k$, to let them have an equal Euclidean norm, i.e., $\|\mathbf{f}_1^k\|_2 = \cdots = \|\mathbf{f}_{n_k}^k\| = c_k$. And (16) is again equal to a constant value no matter what feasible $\mathbf{P}^k$ or θ^k is used. We thus finish the first part of the proof.

Denote Θ as the constraint set of vectors of (13), and $\widehat{\Theta}$ as the constraint set of vectors of (14), which is compact and convex. It has been proved that $\widehat{\Theta}$ is the convex hull of Θ, and the set of vertices of $\widehat{\Theta}$ is exactly Θ (see Appendix A.2 in [18]). It is also known that the minimum of a linear function over a compact convex set is located at an extreme point of the set [58]. Consequently, by relaxing Θ to its convex hull $\widehat{\Theta}$, we get the equivalent LP problem (14) of (13). □

Theorem 1 shows that the relaxation of (13) to (14) is exact: solving (14) gives rise to a solution θ^k satisfying the constraints in (13), i.e., the attained optimal $\mathbf{P}^k$ is still a PPM. The LP problem (14) can be easily solved by any off-the-shelf LP solver [59].

After solving K (14)-like LP problems for $k = 1, \ldots, K$, we get the updates of $\{\theta^k_{t+1}\}_{k=1}^K$ and compute $\mathbf{D}_{t+1} = \big[\mathbf{G}^1\theta^1_{t+1}, \ldots, \mathbf{G}^K\theta^K_{t+1}\big]$. 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 [3]. To improve the convergence, a common practice is use a monotonically increasing sequence of $\{\rho_t\}$. We also adopt this strategy. The pseudocode of our algorithm is summarized in Algorithm 1. Finally, we want to note that the so obtained solution $\{\mathbf{P}^k\}_{k=1}^K$ by solving (5) 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 obtained solution $\{\mathbf{P}^k\}_{k=1}^K$ have been transformed to some canonical form for ease of evaluation.

Algorithm 1: Solving ROML by ADMM

```
input : Feature vectors $\mathbf{F}^k = [\mathbf{f}^k_1, \ldots, \mathbf{f}^k_{n_k}] \in \mathbb{R}^{d \times n_k}$
        (normalized to be $\|\mathbf{f}^k_1\|_2 = \cdots = \|\mathbf{f}^k_{n_k}\| = c_k$
        when there exist outliers), $k = 1, \ldots, K$, the
        number $n$ of inliers, weight $\lambda > 0$, and
        initialization of $\{\mathbf{P}^k_0 \in \mathcal{P}^k\}_{k=1}^K$, $\mathbf{L}_0 = 0$,
        $\mathbf{E}_0 = 0$, $\mathbf{Y}_0 = 0$, and $\rho_0 > 0$.
while not converged do
    $(\mathbf{U}, \mathbf{S}, \mathbf{V}) = \mathrm{svd}\big(\mathbf{D}_t - \mathbf{E}_t - \frac{1}{\rho_t}\mathbf{Y}_t\big)$.
    $\mathbf{L}_{t+1} = \mathbf{U}\mathcal{T}_{\frac{1}{\rho_t}}\big[\mathbf{S}\big]\mathbf{V}^\top$.
    $\mathbf{E}_{t+1} = \mathcal{T}_{\frac{\lambda}{\rho_t}}\big[\mathbf{D}_t - \mathbf{L}_{t+1} - \frac{1}{\rho_t}\mathbf{Y}_t\big]$.
    for each $k$ do
        let $\theta^k_t = \mathrm{vec}(\mathbf{P}^k_t)$, $\mathbf{G}^k = \mathbf{I}_n \otimes \mathbf{F}^k$,
        $\mathbf{J}^k = \mathbf{I}_n \otimes \mathbf{1}^\top_{n_k}$, $\mathbf{H}^k = \mathbf{1}^\top_n \otimes \mathbf{I}_{n_k}$, update $\theta^k_t$ via
        solving the LP problem (14):
        $\theta^k_{t+1} = \arg\min_{\theta^k} -\mathbf{e}^\top_k\big[\mathbf{Y}^\top_t + \rho_t\big(\mathbf{L}_{t+1} +$
        $\mathbf{E}_{t+1}\big)^\top\big]\mathbf{G}^k\theta^k \quad \text{s.t.} \quad \mathbf{J}^k\theta^k = \mathbf{1}_n,\ \mathbf{H}^k\theta^k \leq$
        $\mathbf{1}_{n_k},\ \theta^k \geq 0$.
    end
    $\mathbf{D}_{t+1} = \big[\mathbf{G}^1\theta^1_{t+1}, \ldots, \mathbf{G}^K\theta^K_{t+1}\big]$.
    $\mathbf{Y}_{t+1} = \mathbf{Y}_t + \rho_t\big(\mathbf{L}_{t+1} + \mathbf{E}_{t+1} - \mathbf{D}_{t+1}\big)$.
    $\rho_{t+1} \leftarrow \rho_t$.
    $t \leftarrow t + 1$.
end
output: solution $\{\mathbf{P}^k_t\}_{k=1}^K$, $\mathbf{L}_t$, $\mathbf{E}_t$ to the problem (5).
```

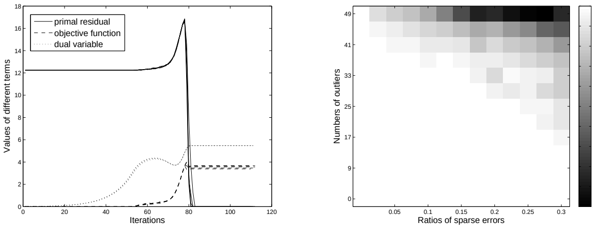
3.2.1 *Convergence Analysis*

The ADMM method has been 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 [60], [61], [62], [63] and references therein). For objective function with more than two separable convex functions, recent results show that under some error bound conditions, the linear convergence of ADMM method is guaranteed [64]. In our case, the ROML problem (5) is nonconvex, due to the binary constraint associated with $\{\mathbf{P}^k\}_{k=1}^K$. A natural question is why we do not relax it by its convex hull in the formulation of (5) so that the well-established convergence results can be readily applied. The reason is that the relaxation could lead to a convex-hull-relaxed version of problem (13), whose solution of $\{\theta^k\}_{k=1}^K$ (or $\{\mathbf{P}^k\}_{k=1}^K$) could be undesirable since it could arbitrarily synthesize new features $\{\mathbf{F}^k\mathbf{P}^k\}_{k=1}^K$ (by convex combination) that jointly form a low-rank matrix plus some small errors. As it turns out, we have to tackle binary integer optimization problem (13). Fortunately, with constant ℓ_2-norm property of feature vectors in each of $\{\mathbf{F}^k\}_{k=1}^K$, the solution of problem (13) can be exactly obtained by any efficient LP solver (cf. Theorem 1).

Regarding the ADMM method for the (nonconvex) ROML problem (5), the convergence property is still an open question in theory. However, it is not uncommon to see that the ADMM method has served a powerful heuristic for some nonconvex problems in practice, such as non-negative matrix factorization [65] and matrix separation [66]. One may anticipate that using ADMM for handling (5) will possibly have faster convergence or convergence to a point with lower minimum objective value, in comparison to other local optimization methods [25]. In the following, we present simulated experiments that demonstrate the excellent convergence property of the ADMM method for ROML.

Specifically, we generate synthetically $K = 30$ groups of vectors simulating extracted feature vectors from K images, and the dimension of each vector $\mathbf{f}$ is $d = 50$. There are 50 vectors including both inliers and outliers in each group. The inliers are produced by randomly generating d-dimensional vectors whose entries are drawn from i.i.d. normal distribution, and are shared in each of the K groups. The outliers are similarly produced by randomly generating d-dimensional vectors following i.i.d. normal distribution, but are independently generated for each group. We then add sparse errors of large magnitude to both inlier and outlier vectors. For each vector $\mathbf{f}$, the error values are uniformly drawn from the range $[-2\max(\mathrm{abs}(\mathbf{f})), 2\max(\mathrm{abs}(\mathbf{f}))]$. Finally, we normalize all vectors to constant ℓ_2-norm to fit with our algorithmic settings.

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



(a) (b)

Fig. 1. Simulation of Algorithm 1: (a) convergence plot in terms of the primal residual, objective function, and dual variable; (b) recovery precisions under varying numbers of outliers and ratios of sparse errors.

We investigate the convergence and recovery properties of our algorithm under varying numbers of outliers and ratios of sparse errors. The numbers of outliers in each group are ranged in $[0, 49]$, and the ratios of sparse errors in each vector are ranged in $[0, 0.3]$. We set $\lambda = 5/\sqrt{d}$ in these experiments. 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 precision 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 and $\| \cdot \|_0$ counts the number of nonzero (1-valued) matrix entries. For each setting of outlier number and sparse error ratio, we run 5 random tests and average the results. Figure 1-(b) reports the recovery precisions 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 45 and sparse error ratio is 0.2), we plot in Figure 1-(a) its convergence of 5 random tests in terms of the primal residual ($\|\mathbf{L} + \mathbf{E} - \mathbf{D}\|_F$), objective function ($\|\mathbf{L}\|_* + \lambda\|\mathbf{E}\|_1$), and dual variable ($\|\mathbf{Y}\|_F$). Convergence properties under other settings are similar to Figure 1-(a).

3.2.2 Computational Complexity

For ease of analysis we assume here $n_1 = \cdots = n_k = \cdots = n_K > n$. The main computation of Algorithm 1 comes from solving the K LP problems in each iteration, whose worst-case complexity is $\mathcal{O}\big(K(n_k^3 + n_k^2 n) n_k^{1/2} n^{1/2} + K d n^2 n_k\big)$. The overall complexity for each iteration is $\mathcal{O}\big(K(n_k^3 + n_k^2 n) n_k^{1/2} n^{1/2} + K d n^2 n_k + K^2 d n\big)$. 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. However, if ρ_t increases too fast, it has the risk of losing optimality [56]. In our experiments, we always set $\rho_0 = 1\mathrm{e}^{-6}$ and increase it with a factor of 1.01 after each iteration. Under this setting, it normally takes $500 \sim 1000$ iterations for Algorithm 1 to converge.

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

The simplest choice of feature representation is based on 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 [26], [27], [30], [31], [28].

For a moving rigid object in a video sequence or images of a rigid object captured from different viewpoints, denote $\mathbf{f}_i^k = [x_i^k, y_i^k]^\top \in \mathbb{R}^2$, $i = 1, \ldots, n_k$, as image coordinates based n_k features extracted from the k^{th} image. Let $\mathbf{F}^k = [\mathbf{f}_1^k, \ldots, \mathbf{f}_{n_k}^k] \in \mathbb{R}^{2 \times n_k}$. It has been shown in [22] that the matrix, defined by

$$\mathbf{D}' = \begin{bmatrix} \mathbf{F}^1\mathbf{P}^1 \\ \vdots \\ \mathbf{F}^K\mathbf{P}^K \end{bmatrix} \in \mathbb{R}^{2K \times n}, \tag{17}$$

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 $\{\mathbf{f}_i^k\}_{k=1}^K$, $i \in \{1, \ldots, n\}$, can be aligned in the same column of $\mathbf{D}'$. (17) is slightly different from the formation of $\mathbf{D}$ in (2). By applying the same low-rank and sparse constraints as in (5), we will show in the experiments in Section 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 [13], HOG [15], Geometric Blur [14], [1], GIST [51], 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, change of size, 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 [45], [44]. In Section 6, we present how our ROML framework can be applied to this application.

4.3 Combination of Image Coordinates and Region Descriptors

In some cases, local region descriptors alone are ambiguous for feature matching, for example, 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 feature 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 [2], [5], or linear-form constraints benefiting from a template image [11], [12]. In this work, we consider a simple method introduced in [23]. 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. For completeness of our presentation, we briefly summarize this method as follows.

Given a set of K images, denote $\mathbf{A}^k_{spa.} \in \mathbb{R}^{n_k \times n_k}$ 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 [23]. In this work, we compute $\mathbf{A}^k_{spa.}$ using Gaussian kernel as $\mathbf{A}^k_{spa.}(i,j) = \mathrm{e}^{-\|\mathbf{x}^k_i - \mathbf{x}^k_j\|^2/2\sigma^2_{spa.}}$, where $\mathbf{x}^k = [x^k, y^k]^\top$ denotes image coordinates in the k^{th} image, and $\sigma_{spa.}$ is a scaling parameter. Each feature point has an associated region descriptor, such as those used in the above subsection. Denote $\mathbf{A}^{pq}_{des.} \in \mathbb{R}^{n_p \times n_q}$ 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. $\mathbf{A}^{pq}_{des.}$ can be computed similar to $\mathbf{A}^k_{spa.}$ as $\mathbf{A}^{pq}_{des.}(i,j) = \mathrm{e}^{-\|\mathbf{f}^p_i - \mathbf{f}^q_j\|^2/2\sigma^2_{des.}}$, where $\mathbf{f}^p_i$ and $\mathbf{f}^q_j$ are feature descriptors from the p^{th} and q^{th} images respectively, and $\sigma_{des.}$ is a scaling parameter.

The method in [23] 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 $\{\mathbf{f}^k_i \in \mathbb{R}^d\}^{n_k}_{i=1}$, $k = 1, \ldots, K$, be the new features to be learned [4], the above objectives can be formalized as

$$\min \sum_{p,q} \sum_{i,j} \left\|\mathbf{f}^p_i - \mathbf{f}^q_j\right\|^2_2 \mathbf{A}^{pq}_{ij}, \tag{18}$$

where the matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is defined as: $\mathbf{A}^{pq} = \mathbf{A}^k_{spa.}$ when $p = q = k$, $\mathbf{A}^{pq} = \mathbf{A}^{pq}_{des.}$ 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 (18) turns to be a problem of Laplacian embedding [24]. Let $\tilde{\mathbf{F}} = [\mathbf{f}^1_1, \ldots, \mathbf{f}^1_{n_1}, \ldots, \mathbf{f}^K_1, \ldots, \mathbf{f}^K_{n_K}]^\top \in \mathbb{R}^{N \times d}$, (18) can be rewritten in matrix form as

$$\min_{\tilde{\mathbf{F}}} \operatorname{trace}\big(\tilde{\mathbf{F}}^\top \tilde{\mathbf{L}}_{\mathbf{A}} \tilde{\mathbf{F}}\big) \text{ s.t. } \tilde{\mathbf{F}}^\top \tilde{\mathbf{D}}_{\mathbf{A}} \tilde{\mathbf{F}} = \mathbf{I}, \tag{19}$$

where $\tilde{\mathbf{L}}_{\mathbf{A}} = \tilde{\mathbf{D}}_{\mathbf{A}} - \mathbf{A}$ is the Laplacian matrix of $\mathbf{A}$, and $\tilde{\mathbf{D}}_{\mathbf{A}}$ is a diagnal matrix with value of the i^{th} diagonal entry as $\sum_j \mathbf{A}_{ij}$. (19) is a generalized eigenvector problem: $\tilde{\mathbf{L}}_{\mathbf{A}}\mathbf{f} = \beta\tilde{\mathbf{D}}_{\mathbf{A}}\mathbf{f}$. Its optimal solution, i.e., the N new features in the d-dimensional embedded space, can be obtained by the bottom d nonzero eigenvectors.

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 robust matching of non-rigid, articulated objects, or instances of a same object category. Experiments in Section 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 relatively simple rigid object moving in a video sequence, to more challenging non-rigid object matching, and matching instances of a same object category. 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 Section 6 for the application of common object localization. In the following experiments, without mentioning 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 $1\mathrm{e}^{-6}$ and iteratively increased with a factor of 1.01. We conducted experiments on a xxx PC ... The running time ...

5.1 Rigid Object with 3D Motion: the Hotel sequence

The CMU "Hotel" sequence consists of 101 frames of a toy hotel building undergoing 3D motion [32]. Each frame has been manually labelled with the same set of 30 landmark points [4]. 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 15 frames out of the total 101 frames (every 7 frames), in order to simulate the wide baseline matching scenario. Image coordinates of landmark points in these 15 frames were arranged into a matrix $\mathbf{D}' \in \mathbb{R}^{30 \times 30}$ as defined in (17). We used Algorithm 1 to optimize a PPM

[4] For consistency we use the same notation $\mathbf{f}$ for different types of features.

TABLE 1
Results of different methods on the “Hotel” sequence. Accuracies are measured by the match ratio criteria.

Methods	DD [5]	COMPOSE [33]	SMAC [34]	LGM [4]	RankConstraints [20]	One-Shot [23]	Prev [53]	ROML
Accuracies	99.8%	96%	84%	90%		**100%**		**100%**

for each frame, where the penalty parameter was set as $\lambda = 5/\sqrt{30}$.

We compare our method with representative pair-wise (graph) matching methods including Dual Decomposition (DD) [5], COMPOSE [33], SMAC [34], and Learning Graph Matching (LGM) [4], which are based on either linear or quadratic assignment formulations, and also more related methods [20], [23] that are able to simultaneously match the set of 15 frames. For the former set of methods, matching between a total of 105 frame pairs needs to be established. Note that although all these methods are based on image coordinates, many of them have used the advanced shape context features [28]. And the method LGM [4] even has a separate learning stage that learns graph matching functions from another related sequence (the CMU “House” sequence).

To evaluate the performance of different methods, we use the match ratio criteria [5]. Table 1 reports the match ratios of different methods, where results of COMPOSE, SMAC, and LGM are from [5], [23]. From Table 1 we can see that ROML and One-Shot [23] achieve the best performance (no matching error), which clearly shows the advantage of simultaneous matching among the whole set of images. However, One-Shot [23] 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. The method [20] also exploits low-rank constraints, however, its performance is much worse than that of ROML. It is probably due to the nuclear norm relaxation and ADMM optimization used in our framework.

Compared to [20], [23], ROML has the additional advantage of being more robust against missing inliers. To verify, we performed another experiment by removing randomly selected landmark points on 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 $\lambda = 2/\sqrt{30}$ in Algorithm 1. The One-Shot method [23] 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 RankConstraints [20] has also been tuned to its best performance. Table 2 reports the match ratio results, where matching accuracies are computed over non-missing points only. Table 2 clearly shows that ROML is less influenced when there exist missing inlier points.

TABLE 2
Match ratios of different methods on the “Hotel” sequence with varying numbers of missing inlier points on each frame.

No. of missing points	RankCons. [20]	One-Shot [23]	ROML
1		76%	**95%**
3		64%	**79%**
5		59%	**71%**

5.2 Non-Rigid Object

We used the KTH dataset [52] to test the ability of ROML for non-rigid object matching. The task is to match corresponding points on human bodies undergoing articulated body motions. To this end, we chose the low-dimensional embedded feature representation [23], as explained in Section 4.3, which encodes information of both geometric structure and descriptor similarity. We used a “walking” sequence and a “hand waving” sequence from the KTH dataset, and sampled 13 and 17 frames respectively from them covering a half circle of each of their action periods. We extracted SIFT features from each frame of these two sets of images, where the numbers of feature points range from 22 to 43, out of which we manually labelled a consistent set of inlier points for each image set as matching ground truth. Feature matching was realized by optimizing (5). The parameters of affinity matrix construction for low-dimensional feature learning were set as $\sigma_{spa.} = 10$ and $\sigma_{des.} = 0.2$, and the dimensionality of learned features was set as $d = 60$.

We compare our method with One-Shot [23], and also several recent graph matching methods including RRWM [7] and SM [2], and hyper-graph matching methods including TM [6], RRWHM [8], and ProbHM [9]. For One-Shot, we chose its best-performing dimensionality of learned features, and run 10 trials of k-means clustering and averaged the results. For those pair-wise matching methods [7], [2], [6], [8], [9], we generated a total of 78 image pairs for the “walking” set and 136 image pairs for the “hand waving” set respectively. We used the codes available on their webpages to produce results on these two sets, where parameters were tuned to their respective best performance. Note that for these methods, $\min(n_p, n_q)$ correspondences are established between any two images with n_p and n_q feature points respectively, of which we count the number of correct

[5] Denote n as the number of ground truth feature correspondences between a pair of images, and $\bar{n}$ as the number of identified ground truth correspondences for this image pair. The match ratio is computed as $\frac{\sum \bar{n}}{\sum n}$, where $\sum$ stands for summation over all image pairs.

correspondences and compute the match ratio accordingly, while our method only optimizes for the best $n < \min(n_p, n_q)$ correspondences.

Table 3 reports the match ratios of different methods. Example feature correspondences among 4 images for RRWHM [8] and our method are also shown in Figure 2. Both One-Shot and our method can match multiple images simultaneously. However, matching accuracies of One-Shot are generally lower than 50%, which shows that One-Shot cannot perform well in the presence of outliers. Our results are much better than those of One-Shot, which shows that our ROML formulation optimized by the ADMM method is very effective for robust feature matching across a set of images. From Table 3 we can also see that results of both graph matching and hyper-graph matching methods are worse than ours. This demonstrates that leveraging more object pattern constraints (i.e., geometric and feature similarity constraints) from multiple images is very useful for feature matching. Moreover, Figure 2 suggests that our matching results across the 4 images are more consistent than those from graph matching methods: another desired property for many applications such as object recognition, tracking, and 3D reconstruction.

5.3 Object Instances of a Common Category

Lastly, we test how ROML performs to match object instances belonging to a same object category. We used 4 image sets of different categories from the Caltech101 [35] and MSRC [36] datasets. Numbers of images in these 4 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 26 to 127, out of which we manually labelled inlier points as matching ground truth. When some inlier points were not detected by SIFT on 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 again used the type of low-dimensional embedded features as explained in Section 4.3. We used Geometric Blur descriptors [1] to characterize local regions around interest points, and Euclidean distances between points in each image for measuring geometric relations. The parameter settings were same as those used in the preceding subsection.

We again compare our method with One-Shot [23] and recent graph and hyper-graph matching methods [7], [2], [6], [8], [9]. These methods were applied in the same way as they were used in the preceding subsection for matching non-rigid human bodies. However, parameters of these methods were tuned so that they can obtain their respective best results on this task of matching object instances. Table 4 reports results of different methods in terms of match ratio. Example feature correspondences for RRWHM [8] and our method are also shown in Figure 3. From Table 4 and Figure 3 we can see that for the relatively simple "Airplane" and "Motorbike" image sets, our method gives very good matching results, while One-Shot and pair-wise matching methods generally perform worse. The "Car" and "Face" image 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, and greatly outperforms all other methods. Note that graph and hyper-graph matching methods have searched a larger number of correspondences and counted correct ones out of them, and hyper-graph matching methods have exploited high-order geometric relations between feature points (normally relations between 3 points). These again demonstrate the effectiveness of ROML for robustly matching objects in a set of images.

Different choices of dimensionality d in low-dimensional feature learning may influence our method's performance. In Figure 4-(a), we plot our matching accuracies on the 6 test sets in Sections 5.2 and 5.3 with different choices of d. It shows that better results can generally be obtained when $d = 50 \sim 100$. It is expected that our method performs well only when the size of image sets (the K value) is relatively large. In Figure 4-(b), we plot results of our method 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.

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 [37] considers automatically discovering and learning object models from a collection of unlabelled images. In particular, given an image collection containing object instances belonging to unknown object categories, the task is to identify the object 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 instances of a single object 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 object variations and cluttered background.

Many methods have been proposed for this challenging task in either weakly supervised or unsupervised settings [38], [39], [40], [41], [42], [44], [45]. Here we pay more attention to those methods that explicitly take object (or its associated parts/features) localization

TABLE 3
Match ratios of different methods on the KTH "Walking" and "Hand waving" sequences.

Methods	RRWM [7]	SM [2]	TM [6]	RRWHM [8]	ProbHM [9]	One-Shot [23]	ROML
Walking	43%	24%	20%	32%	11%	44%	**87%**
Hand waving	46%	29%	24%	57%	20%	48%	**58%**

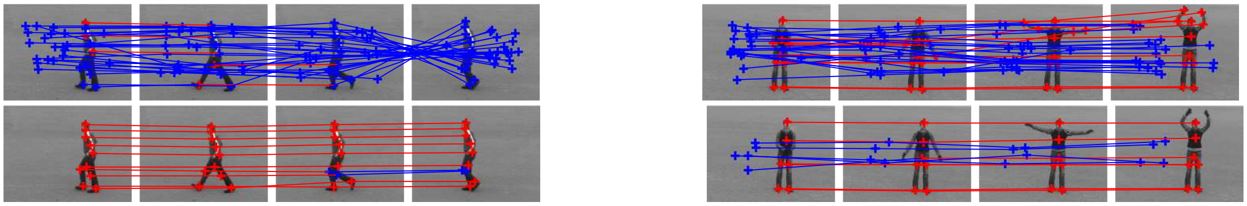


Fig. 2. Example feature correspondences among 4 images of the KTH "Walking" and "Hand waving" sequences respectively. Top row is from RRWHM [8], and bottom row is from our method. Red lines represent identified ground truth correspondences, and blue lines are for false correspondences.

TABLE 4
Match ratios of different methods on image sets of 4 classes from the Caltech101 and MSRC datasets.

Methods	RRWM [7]	SM [2]	TM [6]	RRWHM [8]	ProbHM [9]	One-Shot [23]	ROML
Airplanes	28%	19%	17%	54%	32%	65%	**95%**
Face	40%	22%	26%	54%	14%	61%	**87%**
Motorbike	50%	30%	23%	58%	28%	68%	**95%**
Car	26%	15%	12%	23%	12%	50%	**74%**

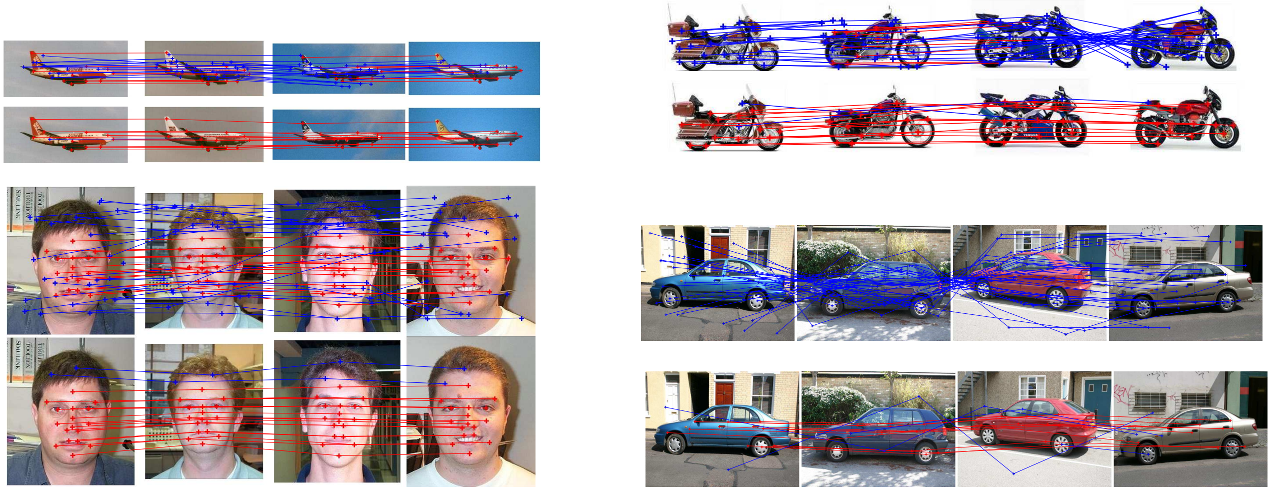
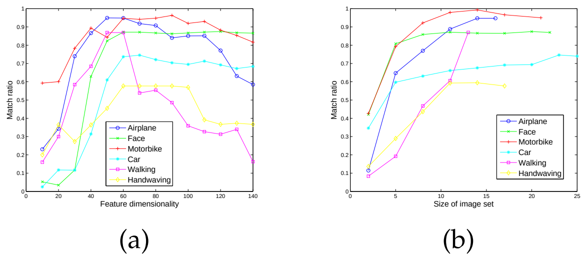


Fig. 3. Example feature correspondences among 4 images of the image sets "Airplanes", "Face", "Motorbike", and "Car" from the Caltech101 and MSRC datasets.. For every pair top is from RRWHM [8], and bottom is from our method. Red lines represent identified ground truth correspondences, and blue lines are for false correspondences.

into account. In particular, for object localization, the methods [39], [38], [41] exploit feature correspondences between each pair of images in the image collection as supporting information. For object class learning, they are based on either clustering algorithms [39], [38] or latent topic models [41]. These methods normally require the objects of interest covering a large portion of the images. More recently, saliency guided object learning techniques [44], [45] are proposed, which exploit generic knowledge of "objectness" [46], [47], [48] obtained from low-level image cues and/or learning from other irrelevant annotated images. Consequently, these methods can potentially locate object instances with large scale and appearance variations in cluttered background.

In this section, we present how ROML can be applied to this COL task, in particular when using local region descriptors as features. Similar to [45], we also sample candidate bounding boxes from each image according to their objectness scores, and use appropriate feature descriptors to characterize the region appearance inside each bounding box. We then optimize (5) to select a bounding box from each image, i.e., $n = 1$ for the PPMs



(a) (b)

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

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. Our method can thus be considered as approximately learning a low-dimensional subspace model for the appearance of each object category discovered from the image collection. Compared to [39], [38], [41], which reply on pair-wise feature matching and iterative procedures for object discovery and class learning, our method is global, runs in a one-pass mode, and can potentially localize object instances more robustly and accurately.

We used the challenging PASCAL datasets [49], [50] to evaluate the capabilities of our method for COL in both weakly supervised and unsupervised settings. For the weakly supervised case, we followed the same experimental settings as in [45]. In particular, we used a subset of the PASCAL06 [49] 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 removed images in which all objects are marked as difficult or truncated in the ground truth annotation. The PASCAL07 dataset [50] is more challenging as objects vary grealy 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 [46] to randomly sample 100 bounding boxes proportionally to their probability of containing an object. To describe the region appearance inside each bounding box, we used the GIST descriptor with the default parameters as in [51], which gives a 512-dimensional feature vector for each region inside a bounding box. We also augmented this vector with aspect ratio ($width/height$) of the bounding box so that the selected bounding boxes from different images have more consistent shapes.

We measured COL performance by the percentage of correctly localized images out of all images in a class/viewpoint combination, where localization correctness in an image was based on PASCAL criteria, i.e., intersection of a bounding box with ground truth is more than half of their union. We compared with several baseline weakly supervised object localization and learning methods including MultiSeg [42] and Exemplar [43], and also with WSL-GK [45], which is saliency guided and performs EM-like alternation of localizing objects and learning a model for the object class.

Table 5 reports COL performance of different methods on the PASCAL06 and PASCAL07 datasets, in terms of percentage of correct localization in images over all class/viewpoint combinations. From Table 5 we can see that Objectness [46] gives very good initial candidates of object bounding boxes. Consequently, results of both our method and WSL-GK [45] on the PASCAL06 and PASCAL07 datasets compare favorably with those from MultiSeg [42] and Exemplar [43]. For the PASCAL07 dataset, our method is comparable to WSL-GK [45] when no iterative steps of class learning are performed in [45], and greatly outperforms [45] for the PASCAL06 dataset, for which our result in fact approaches final result of [45], 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 robust object matching and localization, we defer the possible 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 6 reports detailed results of different class/viewpoint combinations, where we also list results of ROML in the weakly supervised setting. Table 6 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 Figure 5, 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 Conclusion

References

[1] A. C. Berg, T. L. Berg, and J. Malik, Shape matching and object recognition using low distortion correspondences, *CVPR*, 2005.

TABLE 5
COL accuracies of different methods on the PASCAL06 and PASCAL07 datasets. For objectness [46], sampled bounding box with the highest score in each image is considered as the estimated localization.

	Objectness [46]	MultiSeg [42]	Exemplar [43]	WSL-GK [45] (No Learning)	ROML	WSL-GK [45] (With Learning)
PASCAL06	51%	28%	45%	55%	**64**%	64%
PASCAL07	28%	22%	33%	**37**%	36%	50%

TABLE 6
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%

[2] M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, *ICCV*, 2005.

[3] M. Cho, J. Lee, and K. M. Lee, Feature correspondence and deformable object matching via agglomerative correspondence clustering, *ICCV*, 2009.

[4] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola, Learning graph matching, *IEEE Trans. on PAMI*, Vol. 31, No. 6, pp. 1048-1058, 2009.

[5] L. Torresani, V. Kolmogorov, and C. Rother, Feature correspondence via graph matching: models and global optimization, *ECCV*, 2008.

[6] O. Duchenne, F. Bach, I. Kweon, and J. Ponce, A tensor-based algorithm for high-order graph matching, *CVPR*, 2009.

[7] M. Cho, J. Lee, and K. M. Lee, Reweighted random walks for graph matching, *ECCV*, 2010.

[8] J. Lee, M. Cho, K. M. Lee, Hyper-graph matching via reweighted random walks, *CVPR*, 2011.

[9] R. Zass and A. Shashua, Probabilistic graph and hypergraph matching, *CVPR*, 2008.

[10] M. Leordeanu, A. Zanfir1, C. Sminchisescu, Semi-supervised learning and optimization for hypergraph matching, *ICCV*, 2011.

[11] H. Li, E. Kim, X. Huang, and L. He, Object matching with a locally affine-invariant constraint, *Computer Vision and Pattern Recognition*, 2010.

[12] H. Jiang and S. X. Yu, Linear solution to scale and rotation invariant object matching, *Computer Vision and Pattern Recognition*, 2009.

[13] D. G. Lowe, Object recognition from local scale-invariant features, *ICCV*, pp. 11501157, 1999.

[14] A. C. Berg and J. Malik, Geometric blur for template matching, *CVPR*, pp. 607-614, 2001.

[15] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, *CVPR*, pp. 886-893, 2005.

[16] K. Mikolajczyk and C. Schmid, Scale and affine invariant interest point detectors. *IJCV*, 60(1):6386, 2004.

[17] T. Serre, L. Wolf, and T. Poggio, Object recognition with features inspired by visual cortex, *CVPR*, 2005.

[18] J. Maciel and J. P. Costeira, A global solution to sparse correspondence problems, *PAMI*, Vol. 25, No. 2, pp. 187 - 199, 2003.

[19] R. Horn and C. Johnson, Matrix Analysis, *Cambridge Univ. Press*, 1985.

[20] R. Oliveira , J. Costeira , and J. Xavier, Optimal point correspondence through the use of rank constraints, *CVPR*, 2005.

[21] R. Oliveira, R. Ferreira, and J. P. Costeira, Optimal multi-frame correspondence with assignment tensors, *ECCV*, 2006.

[22] C. Tomasi and T. Kanade, Shape from motion from image streams under orthography: a factorization method, *IJCV*, 9(2):137-154, 1992.

[23] M. Torki and A. Elgammal, One-shot multi-set non-rigid feature-spatial matching, *CVPR*, 2010.

[24] P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation*, 2003.

[25] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends in Machine Learning*, 3(1):1122, 2011.

[26] G. Scott and H. Longuet-Higgins, An algorithm for associating the features of two images, *The Royal Society of London*, 1991.

[27] L. Shapiro and J. Brady, Feature-based correspondence: an eigenvector approach, *Image and Vision Computing*, 1992.

[28] S. Belongie, J. Malik, and J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Trans. on PAMI*, 2002.

[29] D. Conte, P. Foggia, C. Sansone, and M. Vento, Thirty years of graph matching in pattern recognition, *Int. J. of Pattern Recognition and Artificial Intelligence*, 2004.

[30] P. J. Besl and N. D. McKay, A method for registration of 3-D shapes, *IEEE Trans. on PAMI*, 1992.

[31] H. Chui and A. Rangarajan, A new point matching algorithm for non-rigid registration, *Computer Vision and Image Understanding*, 2002.

[32] The CMU "Hotel" data set, http://vasc.ri.cmu.edu/idb/html/motion/hotel/index.html.

[33] J. Duchi, D. Tarlow, G. Elidan, and D. Koller, Using combinatorial optimization within max-product belief propagation, *NIPS*, 2007.

[34] T. Cour, P. Srinivasan, and J. Shi, Balanced graph matching, *NIPS*, 2007.

[35] L. Fei-Fei, R. Fergus, and P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, *CVPR Workshop on Generative-Model Based Vision*, 2004.

[36] The MSRC dataset, http://research.microsoft.com/en-us/projects /objectclassrecognition/

[37] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine, Unsupervised object discovery: A comparison, *IJCV*, 2009.

[38] G. Kim, C. Faloutsos, and M. Hebert, Unsupervised modeling of object categories using link analysis techniques, *CVPR*, 2008.

[39] Y. J. Lee and K. Grauman, Foreground focus: Unsupervised learning from partially matching images, *IJCV*, 85:143166, 2009.

[40] Y. J. Lee and K. Grauman, Learning the easy things first: Self-paced visual category discovery, *CVPR*, 2011.

[41] D. Liu and T. Chen, Unsupervised image categorization and object localization using topic models and correspondences between images, *ICCV*, 2007.

[42] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, Using multiple segmentations to discover objects and their extent in image collections, *CVPR*, 2006.

[43] O. Chum and A. Zisserman, An exemplar model for learning object classes, *CVPR*, 2007.

[44] J.-Y. Zhu, J. Wu, Y. Wei, E. Chang, and Z. Tu, Unsupervised object class discovery via saliency-guided multiple class learning, *CVPR*, 2012.

[45] T. Deselaers, B. Alexe, and V. Ferrari, Weakly supervised localization and learning with generic knowledge, *IJCV*, 2012.

[46] B. Alexe, T. Deselares, and V. Ferrari, What is an object?, *CVPR*, 2010.

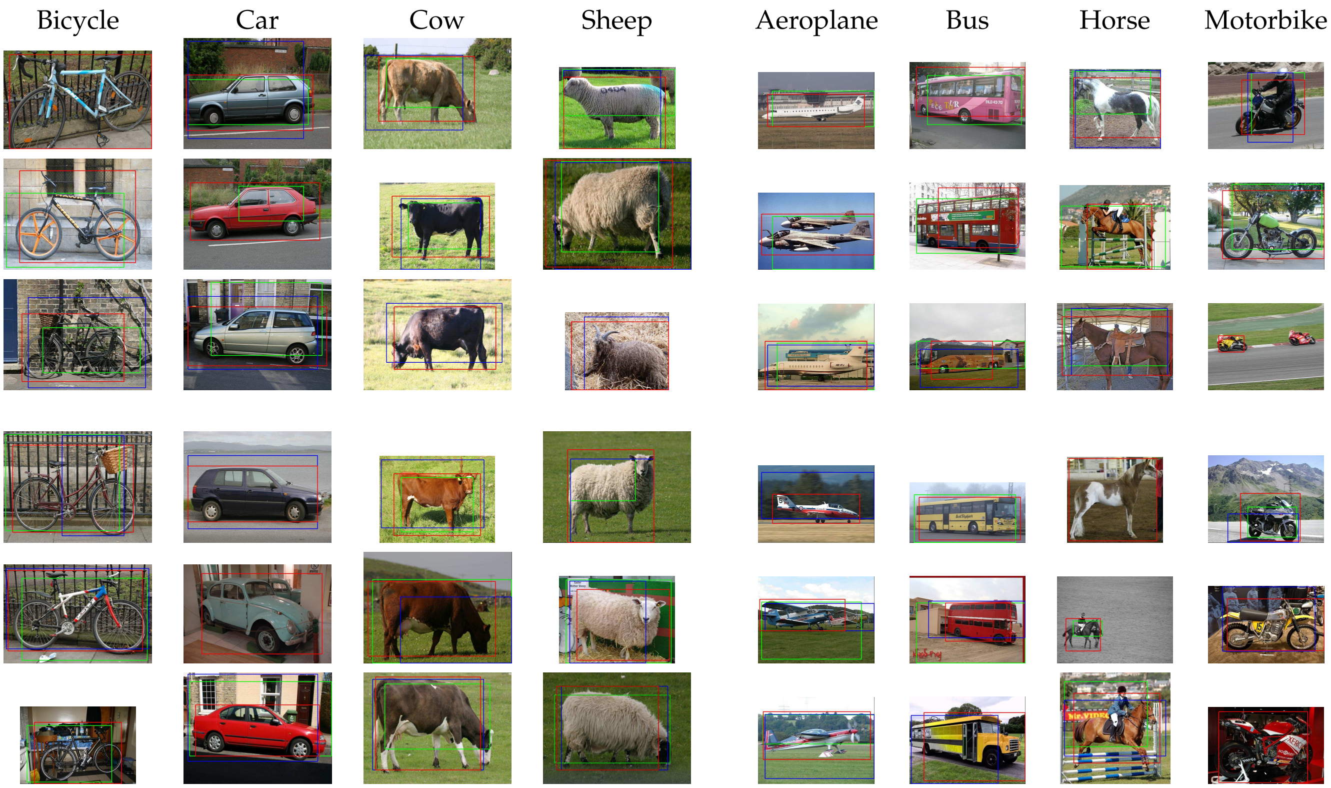


Fig. 5. Example images with estimated bounding boxes of different class/viewpoint combinations from the PASCAL06 and PASCAL07 datasets. In each image, results of Objectness [46] with the highest objectness score (green box), and ROML in unsupervised (blue box) and weakly supervised (red box) settings 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.

[47] I. Endres and D. Hoiem, Category independent object proposals, *ECCV*, 2010.

[48] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, Salient object detection by composition, *ICCV*, 2011.

[49] M. Everingham, L. Van Gool, C. Williams, and A. Zisserman, The PASCAL Visual Object Classes Challenge 2006 (VOC2006), 2006.

[50] M. Everingham, L. Van Gool, C. Williams, L. Winn, and A. Zisserman, The PASCAL Visual Object Classes Challenge 2007 Results, 2007.

[51] A. Oliva and A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, *IJCV* 42(3):145-175, 2001.

[52] C. Schuldt, I. Laptev, and B. Caputo, Recognizing human actions: a local SVM approach, *ICPR*, 2004.

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

[54] E. Candes, X. Li, Y. Ma, and J. Wright, Robust Principal Component Analysis?, *Journal of the ACM*, vol. 58, issue 3, 2011.

[55] D. P. Bertsekas, Nonlinear Programming, *Athena Scientific*, 1999.

[56] Z. Lin, M. Chen, L. Wu, and Y. Ma, The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices, *UIUC Tech. Report UILU-ENG-09-2215*, 2009.

[57] C. Chen, B. He, and X. Yuan, Matrix completion via alternating direction methods, *IMA Journal of Numerical Analysis*, 32, 227-245, 2012.

[58] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, Convex Analysis and Optimization, *Athena Scientific*, 2003.

[59] J. F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, *Optimiz. Methods Softw.*, vol. 11-12, pp. 625-653, 1999.

[60] J. Eckstein and D. P. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, *Mathematical Programming*, vol. 55, pp. 293-318, 1992.

[61] B. He, L.-Z. Liao, D. Han, and H. Yang, A new inexact alternating directions method for monotone variational inequalities, *Mathematical Programming*, vol. 92, no. 1, pp. 103-118, 2002.

[62] D. Goldfarb, S. Ma and K. Scheinberg, Fast alternating linearization methods for minimizing sum of two convex functions, *ArXiv preprint*, arXiv:0912.4571, 2009.

[63] W. Deng and W. Yin, On the global and linear convergence of the generalized alternating direction method of multipliers, *Rice University CAAM Technical Report*, 2012.

[64] M. Hong and Z.-Q. Luo, On the linear convergence of the alternating direction method of multipliers, *ArXiv preprint*, arXiv:1208.3922, 2012.

[65] Y. Zhang, An Alternating direction algorithm for nonnegative matrix factorization, *Rice University CAAM Technical Report*, TR10-03, 2010.

[66] Y. Shen, Z. Wen, and Y. Zhang, Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization, *Rice University CAAM Technical Report*, TR11-02, 2011.