Robust Motion Segmentation with Unknown Correspondences

Pan Ji[1], Hongdong Li[1], Mathieu Salzmann[1,2], and Yuchao Dai[1]

[1] Australian National University, Canberra
[2] NICTA, Canberra

Abstract. Motion segmentation can be addressed as a subspace clustering problem, assuming that the trajectories of interest points are known. However, establishing point correspondences is in itself a challenging task. Most existing approaches tackle the correspondence estimation and motion segmentation problems separately. In this paper, we introduce an approach to performing motion segmentation without any prior knowledge of point correspondences. We formulate this problem in terms of Partial Permutation Matrices (PPMs) and aim to match feature descriptors while simultaneously encouraging point trajectories to satisfy subspace constraints. This lets us handle outliers in both point locations and feature appearance. The resulting optimization problem can be solved via the Alternating Direction Method of Multipliers (ADMM), where each subproblem has an efficient solution. Our experimental evaluation on synthetic and real sequences clearly evidences the benefits of our formulation over the traditional sequential approach that first estimates correspondences and then performs motion segmentation.

Keywords: Motion segmentation, point correspondence, subspace clustering, partial permutation matrix.

1 Introduction

Motion segmentation is a challenging problem whose outcome can positively impact many scene understanding techniques. It is well known that, under an affine camera model, the trajectories of independent motions lie in different linear (or affine) subspaces [26]. Thus, given the trajectories of points belonging to multiple moving objects, motion segmentation can be addressed as a subspace clustering problem.

Recently, there has been a surge of subspace clustering algorithms [8,13,30,15,9] reporting highly accurate results on benchmark datasets (e.g., [29]). However, motion segmentation is still far from being a solved problem. Indeed, most existing methods assume that complete point trajectories are available as input. For example, in the Hopkins155 dataset [29], perfect trajectories were obtained by manually labeling the points throughout the sequences. Such manual intervention is, of course, impractical in many realistic scenarios. While some methods are robust to small amounts of outliers (e.g., [8,13,9]), their performance quickly degrades

D. Fleet et al. (Eds.): ECCV 2014, Part VI, LNCS 8694, pp. 204–219, 2014.

as the number of mismatches increases. In practice, interest point detection and correspondence estimation are challenging tasks. Inevitable outliers and missing data make the problem even harder.

While research in the area of point correspondence estimation has also been progressing [19,33,10], existing methods are not being considered in the context of motion segmentation. Therefore, they cannot benefit from constraints associated with the problem. In particular, when observing multiple motions, the underlying point trajectories should lie in a union of subspaces.

In this paper, we introduce an approach to performing motion segmentation with unknown correspondences. In contrast to existing techniques that proceed in two stages (i.e., first correspondence estimation and then motion segmentation), this allows us to **(i)** benefit from the motion segmentation constraints throughout the entire process; and **(ii)** not require any pre-processing stage to clean up the trajectories used for motion segmentation, and thus be robust to outliers and missing observations. This, we believe, is a crucial step towards making motion segmentation applicable to more realistic scenarios.

More specifically, given interest points extracted independently in all the frames of a video sequence comprising both inliers and outliers, we exploit the constraint that data lying in the union of subspaces should be self-expressive. In other words, a trajectory lying in a subspace can be expressed as a linear combination of the other trajectories in the same subspace. We therefore search for Partial Permutation Matrices (PPMs) and combination coefficients that automatically select and reorder the inlier points so as to make them self-expressive. Furthermore, we make use of the fact that matched feature descriptors have a similar appearance across the frames and thus, when correctly arranged, should form a low-rank matrix. To obtain a solution to the resulting optimization problem, we employ the Alternating Direction Method of Multipliers (ADMM), and show that each subproblem can be solved efficiently. Given the combination coefficients, we can then separate the different motions by normalized cuts [21] or spectral clustering, as in [8,13,30,15,9].

We demonstrate the robustness and effectiveness of our method on several real sequences. Our experimental evaluation evidences the benefits of our formulation over sequentially solving correspondence estimation and motion segmentation, as is done by existing approaches. Importantly, on Hopkins155, our formulation with unknown correspondences achieves competitive results with the state-of-the-art motion segmentation methods that exploit perfect trajectories as input.

2 Related Work

Over the years, many techniques [1,5,12,25,24,18,8,13,15,9] have been proposed to tackle the problem of motion segmentation. These techniques can be roughly categorized into those working with dense observations, and those tackling the sparse points case.

In the dense scenario, the use of optical flow has been investigated to separate the motion of different objects observed in two frames [5,1], or in very

short sequences [25,24]. Dense point trajectories were also employed for motion segmentation in longer videos [18].

Our work is more directly related to methods that perform motion segmentation of sparse trajectories [26,32,31]. In particular, our approach draws inspiration from the recent subspace clustering literature [8,13,15,9]. Subspace clustering approaches exploit the fact that the trajectories of points belonging to multiple independent motions lie in a union of subspaces. As such, the data can be thought of as self-expressive in the sense that each trajectory can be represented as a linear combination of the other trajectories that lie in the same subspace. The motion segmentation problem is then typically recast as that of finding the coefficients of this linear combinations. Different penalty functions have been proposed to regularize these coefficients, such as sparsity in Sparse Subspace Clustering (SSC) [8], low-rank in Low Rank Representation (LRR) [13], or density in [15,9] of the coefficient matrix. The resulting coefficients are then used to build an affinity matrix, from which motion segmentation is achieved via spectral clustering or normalized cuts.

Whether dense or sparse, trajectory-based methods all assume that the correspondence problem has been solved beforehand, and that the trajectories are thus given as input. While some advances have been made towards handling outliers [8,13,9] and incomplete point tracks [23,4], the resulting techniques still require relatively clean data to yield good accuracy.

Ultimately, motion segmentation methods strongly rely on the accuracy of point correspondences. These correspondences can typically be obtained by independently matching local feature descriptors [14,16], or by making use of the temporal nature of the data to track sparse [22] or dense [2] image points. Rather than treating each point independently, several methods have been proposed to jointly find nonrigid correspondences between two sets of points [11,28,7]. In [27], this is achieved by combining point location and appearance.

In the case of a single rigid motion, it was shown that correspondence estimation can be expressed as a rank-minimization problem in terms of PPMs [19]. Indeed, when correctly organized, the trajectories of rigidly moving points form a rank 4 matrix. While, in [19], this was achieved by incrementally incorporating one frame at a time, which is subject to error propagation, this idea was pursued for Robust Object Matching using Low-rank and sparse constraints (ROML) [33,10], where a whole sequence was treated at once. Furthermore, in [33,10], this framework was extended to minimizing the rank of a matrix built from feature descriptors, thus making the approach applicable to more general correspondence problems.

However, while attractive, general solutions to the correspondence problem, such as [27,33,10] do not permit taking into account the specific constraints of the task at hand. Here, in contrast, we introduce an approach that jointly performs correspondence estimation and motion segmentation, and can thus incorporate the subspace constraints in motion segmentation throughout the whole process. As a result, not only does it yield high segmentation accuracy, but it also improves correspondence estimation.

3 Problem Formulation

We now present our approach to robust motion segmentation with unknown correspondences. Intuitively, we seek to select the inlier input points and reorder them such that they satisfy the subspace constraints (i.e., the point trajectories lie in a union of subspaces) and appearance constraints (i.e., the matched feature points have similar appearance across the images).

More specifically, let $\mathbf{w}_{fi} \in \mathbb{R}^{2\times 1}$ be the 2D location of point i detected in frame f of an F-frame sequence depicting multiple motions. Furthermore, let $\mathbf{t}_{fi} \in \mathbb{R}^{d\times 1}$ be the appearance descriptor of the same point in the same frame. The locations of all points in frame f can be concatenated in a $2 \times N_f$ position matrix $\mathbf{W}_f = [\mathbf{w}_{f1}, \cdots, \mathbf{w}_{fN_f}]$. Similarly, we can group all feature descriptors in frame f in a $d \times N_f$ matrix $\mathbf{T}_f = [\mathbf{t}_{f1}, \cdots, \mathbf{t}_{fN_f}]$.

In the absence of point correspondence across the frames, and even if we assume $N_f = N$, $\forall f$, simply stacking up of all the position matrices $\{\mathbf{W}_f\}_{f=1}^F$ does not yield valid point trajectories. However, there exists a reordering of the columns of the position matrix in each frame that yields coherent point trajectories[1]. Furthermore, when applied to the descriptor matrices $\{\mathbf{T}_f\}_{f=1}^F$, this ordering should also make the appearance of corresponding features coherent across the frames. In the presence of outliers, i.e., $N_f \neq N_{f'}$, $f \neq f'$, this process should not only reorder the points, but also select the inliers.

Following [19,4,33], we utilize Partial Permutation Matrices (PPMs) to model this reordering. Let $\mathbf{P}_f \in \{0,1\}^{N_f \times N}$ denote the PPM that selects and reorders the N inlier point coordinates in frame f. Given the F PPMs $\{\mathbf{P}_f\}_{f=1}^F$, we define the trajectory matrix as

$$\mathbf{D}_c = [(\mathbf{W}_1\mathbf{P}_1)^T | \cdots | (\mathbf{W}_F\mathbf{P}_F)^T]^T \ . \tag{1}$$

In an ideal, noise-free scenario, there exist PPMs such that the trajectory matrix $\mathbf{D}_c$ satisfies the subspace constraints. In practice, to account for noise of measurements, we decompose $\mathbf{D}_c$ into a clean measurement matrix $\mathbf{L}_c$ and a noise matrix $\mathbf{E}_c$. This can be written as

$$\mathbf{D}_c = \mathbf{L}_c + \mathbf{E}_c \ . \tag{2}$$

As was shown in [8,13], the fact that trajectories lie in a union of subspaces can equivalently be formulated in terms of self-expressiveness of the data. Note, however, that self-expressiveness only holds in the noise-free case. In our formulation, we therefore make use of the clean measurement matrix $\mathbf{L}_c$ to encode self-expressiveness. This yields

$$\mathbf{L}_c = \mathbf{L}_c\mathbf{C} \ , \tag{3}$$

[1] Note that this global reordering is subject to an ambiguity, since the order of the trajectories themselves is irrelevant. This, however, can easily be solved by fixing the order in the first frame of the sequence.

where each clean trajectory is represented as a linear combination of the other trajectories, with $\mathbf{C}$ storing the (unknown) combination coefficients. In the presence of affine subspaces, an additional constraint of the form $\mathbf{1}_N^T\mathbf{C} = \mathbf{1}_N^T$, where $\mathbf{1}_N$ is a column vector of 1s, can be further imposed.

For feature appearance, we define the $Nd \times F$ descriptor matrix obtained from the PPMs $\{\mathbf{P}_f\}_{f=1}^F$ as

$$\mathbf{D}_d = [\text{vec}(\mathbf{T}_1\mathbf{P}_1)| \cdots |\text{vec}(\mathbf{T}_F\mathbf{P}_F)] \,, \tag{4}$$

where $\text{vec}(\cdot)$ vectorizes its matrix argument in a columnwise manner.

Since, in a noise-free scenario, each specific feature point should have the same appearance in all the frames, $\mathbf{D}_d$ should have low rank (ideally rank one). To tackle the more realistic case of noisy measurements, however, we decompose $\mathbf{D}_d$ into a clean low rank component $\mathbf{L}_d$ and a noise component $\mathbf{E}_d$. Therefore, we have

$$\mathbf{D}_d = \mathbf{L}_d + \mathbf{E}_d. \tag{5}$$

In this formalism, our goal is to propose a formulation to motion segmentation with unknown correspondences, which satisfies the following requirements:

1. The matrix of clean inlier point trajectories $\mathbf{L}_c$ should be self-expressive.
2. The matrix of clean inlier feature descriptors $\mathbf{L}_d$ should have low-rank.
3. Noise and outliers in both point locations and feature descriptors must be accounted for.

While Point 1 is partially accounted for by the constraint in Eq. 3, it is crucial to prevent non-zero coefficients in $\mathbf{C}$ for any two trajectories belonging to different motions. Indeed, to perform motion segmentation, $\mathbf{C}$ needs to reflect the membership of the trajectories to their respective subspace. As was shown in [9], this can be achieved with any p-norm regularizer on $\mathbf{C}$. Here, in particular, we make use of the Frobenius norm, which is convex and easy to minimize. To address Point 2 in our requirements, we propose to search for the $\mathbf{L}_d$ with minimum rank. To this end, we employ a nuclear norm regularizer on $\mathbf{L}_d$, which is a convex surrogate to the rank function. Finally, Point 3 is addressed in two different ways. First, outliers in the point locations are accounted for by the PPMs. Second, to model further noise in the locations and in the descriptors, which we expect to be sparse, we make use of ℓ_1 regularizers on $\mathbf{E}_c$ and $\mathbf{E}_d$ as convex surrogates to the ℓ_0 norm.

Integrating all these constraints, we express motion segmentation with unknown correspondences as the optimization problem

$$\begin{aligned} \min_{\{\mathbf{P}_f\}_{f=1}^F, \mathbf{C}, \mathbf{L}_{c,d}, \mathbf{E}_{c,d}} & \frac{1}{2}\|\mathbf{C}\|_F^2 + \lambda_1\|\mathbf{L}_d\|_* + \lambda_2\|\mathbf{E}_c\|_1 + \lambda_3\|\mathbf{E}_d\|_1 \\ \text{s.t.} \quad & \mathbf{L}_c = \mathbf{L}_c\mathbf{C} \,,\ \left(\mathbf{1}_N^T\mathbf{C} = \mathbf{1}_N^T\right) \,, \\ & \mathbf{D}_c = \mathbf{L}_c + \mathbf{E}_c,\ \mathbf{D}_d = \mathbf{L}_d + \mathbf{E}_d \,, \\ & \mathbf{1}_{N_f}^T\mathbf{P}_f = \mathbf{1}_N^T,\ \mathbf{P}_f\mathbf{1}_N \leq \mathbf{1}_{N_f},\ \mathbf{P}_f \in \{0,1\}^{N_f \times N} \,, \end{aligned} \tag{6}$$

where λ_1, λ_2, and λ_3 balance the different terms in the objective function, and where the constraints on $\{\mathbf{P}_f\}_{f=1}^F$ enforce these matrices to be PPMs. Note that these PPMs appear in (6) via the matrices $\mathbf{D}_c$ and $\mathbf{D}_d$ as can be seen from their definitions in Eqs. 1 and 4, respectively.

Note that, for motion segmentation with *known* correspondences, the use of the nuclear norm on the coefficients $\mathbf{C}$ has been advocated in the past [13]. This was motivated by the fact that a low-rank $\mathbf{C}$ would inherently incur a low-rank $\mathbf{L}$, thus reflecting the fact that the trajectories of K motion form a rank $4K$ matrix. However, as was shown in [9], under the self-expressiveness constraint $\mathbf{L} = \mathbf{LC}$, $\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{C}\|_F^2$ and $\min_{\mathbf{C}} \|\mathbf{C}\|_*$ are equivalent. With this observation, the Frobenius norm comes as a natural choice over the nuclear norm in (6) due to its computational simplicity.

Importantly, note that, in (6), the trajectory matrix $\mathbf{D}_c$ and the descriptor matrix $\mathbf{D}_d$ share the same PPMs $\{\mathbf{P}_f\}_{f=1}^F$. This induces a connection between motion segmentation and point correspondence, and thus makes the two problems work in a cooperative manner and help each other during the optimization procedure. As will be shown in our experiments, this collaboration not only yields accurate motion segmentation, but also improves the point correspondence results over methods dedicated to this task only.

3.1 Solving (6)

Due to the discrete nature of PPMs, (6) is non-convex. Here, we propose to solve it via the ADMM, which has proven effective for many non-convex problems such as matrix separation [20], non-negative matrix factorization [34], and correspondence estimation [33]. The ADMM works by decomposing the original optimization problem into several smaller subproblems, each of which can be solved efficiently.

We therefore seek to decompose (6) into several subproblems. With the ADMM, this is achieved by first deriving the augmented Lagrangian of (6), which can be expressed as

$$\begin{aligned}\mathcal{L}_\rho(\{\mathbf{P}_f\}_{f=1}^F, \mathbf{C}, \mathbf{L}_{c,d}, \mathbf{E}_{c,d}, \{\mathbf{Y}_i\}_{i=1}^4) = \frac{1}{2}\|\mathbf{C}\|_F^2 + \lambda_1\|\mathbf{L}_d\|_* + \lambda_2\|\mathbf{E}_c\|_1 + \lambda_3\|\mathbf{E}_d\|_1 \\ +\mathrm{tr}\left(\mathbf{Y}_1^T(\mathbf{L}_c - \mathbf{L}_c\mathbf{C})\right) + \mathrm{tr}\left(\mathbf{Y}_2^T(\mathbf{1}^T\mathbf{C} - \mathbf{1}^T)\right) \\ +\mathrm{tr}\left(\mathbf{Y}_3^T(\mathbf{D}_c - \mathbf{L}_c - \mathbf{E}_c)\right) + \mathrm{tr}\left(\mathbf{Y}_4^T(\mathbf{D}_d - \mathbf{L}_d - \mathbf{E}_d)\right) \\ +\frac{\rho}{2}(\|\mathbf{D}_c - \mathbf{L}_c - \mathbf{E}_c\|_F^2 + \|\mathbf{L}_c - \mathbf{L}_c\mathbf{C}\|_F^2 + \|\mathbf{D}_d - \mathbf{L}_d - \mathbf{E}_d\|_F^2 + \|\mathbf{1}^T\mathbf{C} - \mathbf{1}^T\|_2^2)\,,\end{aligned} \tag{7}$$

where $\{\mathbf{Y}_i\}_{i=1}^4$ are the matrices of Lagrange multipliers corresponding to the four constraints in (6), and ρ is the penalty parameter[2]. Note that, although not explicitly written here, the constraints on the PPMs are maintained.

[2] Note that, for ease of notation, we have omitted explicitly writing the dimension of the vectors of all 1s, now all denoted by $\mathbf{1}$.

The ADMM then consists of iteratively updating the individual variables so as to minimize $\mathcal{L}_\rho$ while the other variables are fixed. In the following, we derive the update for each of our variables. We denote by a superscript t the current value of the variables and by a superscript $t+1$ the new values.

From Eq. 7, it can be seen that $\mathbf{L}_c$ only appears in linear and least-squares terms. Therefore, its update can easily be obtained in closed-form. More specifically, it is given by

$$\mathbf{L}_c^{t+1} = \left[\left(\mathbf{Y}_3^t + \mathbf{Y}_1^t(\mathbf{C}^{t^T} - \mathbf{I})\right)/\rho^t + \mathbf{D}_c^t - \mathbf{E}_c^t\right]\left[\mathbf{I} + (\mathbf{I} - \mathbf{C}^t)(\mathbf{I} - \mathbf{C}^{t^T})\right]^{-1} . \quad (8)$$

Similarly, all the terms involving $\mathbf{C}$ are simple linear and quadratic terms. This yields the closed-form update

$$\mathbf{C}^{t+1} = \left[\mathbf{I} + \rho^t(\mathbf{L}_c^{t+1^T}\mathbf{L}_c^{t+1} + \mathbf{1}\mathbf{1}^T)\right]^{-1}\left(\mathbf{L}_c^{t+1^T}\mathbf{Y}_1^t - \mathbf{1}\mathbf{Y}_2^t + \rho^t(\mathbf{L}_c^{t+1^T}\mathbf{L}_c^{t+1} + \mathbf{1}\mathbf{1}^T)\right). \quad (9)$$

Although not as straightforward, the updates for $\mathbf{L}_d$, $\mathbf{E}_c$ and $\mathbf{E}_d$ can still be computed efficiently. To this end, we note that these updates correspond to the solutions of the following optimization problems:

$$\mathbf{L}_d^{t+1} = \underset{\mathbf{L}_d}{\operatorname{argmin}}\, \lambda_1/\rho^t\|\mathbf{L}_d\|_* + 1/2\|\mathbf{L}_d - (\mathbf{D}_d^t - \mathbf{E}_d^t + \mathbf{Y}_4^t/\rho^t)\|_F^2 , \quad (10)$$

$$\mathbf{E}_c^{t+1} = \underset{\mathbf{E}_c}{\operatorname{argmin}}\, \lambda_2/\rho^t\|\mathbf{E}_c\|_1 + 1/2\|\mathbf{E}_c - (\mathbf{D}_c^t - \mathbf{L}_c^{t+1} + \mathbf{Y}_3^t/\rho^t)\|_F^2 , \quad (11)$$

$$\mathbf{E}_d^{t+1} = \underset{\mathbf{E}_d}{\operatorname{argmin}}\, \lambda_3/\rho^t\|\mathbf{E}_d\|_1 + 1/2\|\mathbf{E}_d - (\mathbf{D}_d^t - \mathbf{L}_d^{t+1} + \mathbf{Y}_4^t/\rho^t)\|_F^2 . \quad (12)$$

Problems (10), (11) and (12) are convex programs whose solutions can be obtained in closed-form. To this end, let us define the soft-thresholding operator [3] $\mathcal{T}_\tau[x] = \text{sign}(x)\cdot\max(|x| - \tau, 0)$, which operates elementwise on scalars or matrices. The optimal solution to (10) can then be obtained as

$$\mathbf{L}_d^{t+1} = \mathbf{U}\mathcal{T}_{\lambda_1/\rho^t}(\Sigma)\mathbf{V}^T, \quad (13)$$

where $[\mathbf{U}, \Sigma, \mathbf{V}] = \text{svd}(\mathbf{D}_d^t - \mathbf{E}_d^t + \mathbf{Y}_4^t/\rho^t)$. The updates for $\mathbf{E}_c$ and $\mathbf{E}_d$ can be written as

$$\mathbf{E}_c^{t+1} = \mathcal{T}_{\lambda_2/\rho^t}(\mathbf{D}_c^t - \mathbf{L}_c^{t+1} + \mathbf{Y}_3^t/\rho^t) . \quad (14)$$

$$\mathbf{E}_d^{t+1} = \mathcal{T}_{\lambda_3/\rho^t}(\mathbf{D}_d^t - \mathbf{L}_d^{t+1} + \mathbf{Y}_4^t/\rho^t) . \quad (15)$$

The PPMs $\{\mathbf{P}_f\}_{f=1}^F$ are binary matrices, and thus updating them is non-trivial. Recall that, in Eq. 7, the PPMs appear via $\mathbf{D}_c$ and $\mathbf{D}_d$ only. Therefore, $\{\mathbf{P}_f\}_{f=1}^F$ can be updated by solving the problem

$$\min_{\{\mathbf{P}_f\}_{f=1}^F} \|\mathbf{D}_c - (\mathbf{L}_c^{t+1} + \mathbf{E}_c^{t+1} - \mathbf{Y}_3^t/\rho^t)\|_F^2 + \|\mathbf{D}_d - (\mathbf{L}_d^{t+1} + \mathbf{E}_d^{t+1} - \mathbf{Y}_4^t/\rho^t)\|_F^2$$
$$\text{s.t. } \mathbf{1}^T\mathbf{P}_f = \mathbf{1},\ \mathbf{P}_f\mathbf{1} \leq \mathbf{1},\ \mathbf{P}_f \in \{0,1\}^{N_f \times N} , \quad (16)$$

where $\mathbf{D}_c = [(\mathbf{W}_1\mathbf{P}_1)^T|\cdots|(\mathbf{W}_F\mathbf{P}_F)^T]^T$, $\mathbf{D}_d = [\text{vec}(\mathbf{T}_1\mathbf{P}_1)|\cdots|\text{vec}(\mathbf{T}_F\mathbf{P}_F)]$ (as defined in Eqs. 1 and 4, respectively).

Problem (16) can then be decomposed into F independent subproblems, each of which only involves one PPM. The subproblem for frame f can be written as

$$\begin{aligned}\min_{\mathbf{P}_f} \; & \|\mathbf{W}_f\mathbf{P}_f - \alpha_f^T(\mathbf{L}_c^{t+1} + \mathbf{E}_c^{t+1} - \mathbf{Y}_3^t/\rho^t)\|_F^2 \\ & + \|\text{vec}(\mathbf{T}_f\mathbf{P}_f) - (\mathbf{L}_d^{t+1} + \mathbf{E}_d^{t+1} - \mathbf{Y}_4^t/\rho^t)\mathbf{e}_f\|_2^2 \\ \text{s.t. } & \mathbf{1}^T\mathbf{P}_f = \mathbf{1},\ \mathbf{P}_f\mathbf{1} \le \mathbf{1},\ \mathbf{P}_f \in \{0,1\}^{N_f \times N} \quad ,\end{aligned} \tag{17}$$

where $\mathbf{e}_f$ is a binary column vector with only the f^{th} element set to 1, and $\alpha_f = [\mathbf{e}_{2f-1}|\mathbf{e}_{2f}]$. Problem (17) turns out to be a binary assignment problem, which can be solved by the Hungarian algorithm in polynomial time [17]. The details of the solution of Problem (17) via the Hungarian algorithm are given in appendix. Once we have computed the updates $\{\mathbf{P}_f^{t+1}\}_{f=1}^F$, $\mathbf{D}_c$ and $\mathbf{D}_d$ can be updated accordingly.

Finally, the Lagrange multipliers $\{\mathbf{Y}_i\}_{i=1}^4$ and ρ can be updated as

$$\mathbf{Y}_1^{t+1} = \mathbf{Y}_1^t + \rho(\mathbf{L}_c - \mathbf{L}_c\mathbf{C}) \; , \tag{18}$$

$$\mathbf{Y}_2^{t+1} = \mathbf{Y}_2^t + \rho(\mathbf{1}^T\mathbf{C} - \mathbf{1}^T) \; , \tag{19}$$

$$\mathbf{Y}_3^{t+1} = \mathbf{Y}_3^t + \rho(\mathbf{D}_c - \mathbf{L}_c - \mathbf{E}_c) \; , \tag{20}$$

$$\mathbf{Y}_4^{t+1} = \mathbf{Y}_4^t + \rho(\mathbf{D}_d - \mathbf{L}_d - \mathbf{E}_d) \; . \tag{21}$$

$$\rho^{t+1} = \min(\eta\rho^t, \rho_m) \; , \tag{22}$$

where $\eta > 1$ and ρ_m is the predefined maximum for ρ.

The process of iteratively updating all the variables is repeated until convergence, or until a maximum number of iterations is reached. Note that, while the ADMM does not have theoretical guarantee of global convergence for non-convex problems, in our simulation we find it always converges to the correct solution for our problem. The empirical convergence of our algorithm will be discussed in Section 4. Our algorithm for motion segmentation with unknown correspondences is summarized in Algorithm 1.

4 Experimental Evaluation

To evaluate the effectiveness of our approach for motion segmentation with unknown point correspondences, we conducted extensive experiments on both synthetic data and real images. In total, we performed four different sets of experiments, which we discuss below. To measure/compare the performances of different algorithms, we use the following criteria: (i) Accuracy in motion segmentation, expressed as

$$\text{ACC}_{\text{ms}} = \frac{\text{total number of correctly segmented trajectories}}{\text{total number of trajectories}},$$

Algorithm 1. Motion segmentation without correspondences via the ADMM

Input:
Position matrices $\{\mathbf{W}_f\}_{f=1}^F$, descriptor matrices $\{\mathbf{T}_f\}_{f=1}^F$, λ_1, λ_2, λ_3, $\eta > 1$, ρ_m, ϵ;

Initialize: $\mathbf{C}^0$, $\{\mathbf{P}_f^0\}_{f=1}^F$, $\mathbf{E}_c^0 = \mathbf{0}$, $\mathbf{E}_d^0 = \mathbf{0}$, $\{\mathbf{Y}_i^0\}_{i=1}^4 = \mathbf{0}$, ρ^0;

while not converged **do**
1. Update $(\mathbf{L}_c, \mathbf{C}, \mathbf{L}_d, \mathbf{E}_c, \mathbf{E}_d)$ by Eq. (8), Eq. (9), Eq. (13), Eq. (14) and Eq. (15);
2. Update $\{\mathbf{P_f}\}_{f=1}^F$ by solving F binary assignment problems (17) using the Hungarian algorithm, and then update $\mathbf{D}_c$ and $\mathbf{D}_d$ accordingly;
3. Update $\{\mathbf{Y}_i\}_{i=1}^4$ and ρ by Eq. (18)-Eq. (22);
4. Check the convergence conditions $\|\mathbf{L}_c - \mathbf{L}_c\mathbf{C}\|_\infty \leq \epsilon$, $\|\mathbf{1}^T\mathbf{C} - \mathbf{1}^T\|_\infty \leq \epsilon$, $\|\mathbf{D}_c - \mathbf{L}_c - \mathbf{E}_c\|_\infty \leq \epsilon$ and $\|\mathbf{D}_d - \mathbf{L}_d - \mathbf{E}_d\|_\infty \leq \epsilon$;

end while

Output: Coefficient matrix $\mathbf{C}$, PPMs $\{\mathbf{P}_f\}_{f=1}^F$.

and (ii) Accuracy in point correspondences, computed as

$$\mathrm{ACC}_{\mathrm{pc}} = \frac{1}{FN^2}\sum_{f=1}^{F}\|\mathbf{P}_f \circ \mathbf{P}_f^*\|_0 \ ,$$

where $\{\mathbf{P}_f^*\}_{f=1}^F$ are the ground truth correspondences (in PPM matrix), $\circ$ denotes the element-wise (Hadamard) product, and $\|\cdot\|_0$ is the ℓ_0 norm which counts the number of non-zeros entries.

4.1 Experiment-1: Synthetic Data, Noise-Free Case

In this first set of experiments, we aim to study the convergence of our algorithm. In other words, we want to understand, with perfectly controlled inputs, whether or not the proposed algorithm converges; and if so, whether it converges to the correct solution.

To this end, we synthesized two motion matrices $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{2F\times 4}$ by simulating F random rotation and translations, and two independently moving objects with shape matrices $\mathbf{S}_1, \mathbf{S}_2 \in \mathbb{R}^{4\times N/2}$. This yields a total of N 3D points in motion. Under an affine camera model, the measurement matrix of the synthesized sequence can be computed as $\mathbf{X} = [\mathbf{M}_1\mathbf{S}_1|\mathbf{M}_2\mathbf{S}_2] \in \mathbb{R}^{2F\times N}$. We chose $F = 25$ and $N = 40$ in all our synthetic experiments. In addition to $\mathbf{X}$, which contains point locations, we synthesized a 128-dimensional appearance vector (i.e., feature descriptor) for each feature point. The choice of 128D is only to conform to the convention of SIFT feature descriptors which will be used in all our real image experiments. We randomly generated 128-dimensional random vectors and required that the same feature point across multiple images has identical descriptors. Given this synthesized sequence, we randomly permuted the points, so that all the correspondence information is lost.

With this data, we tested our algorithm starting from two different initialization conditions (V1 and V2): (V1) Initialize the unknown permutation matrices $\{\mathbf{P}_f\}_{f=1}^F$ as the identity matrices (generally only 0-2% accuracy); (V2) Initialize the unknown permutation matrices $\{\mathbf{P}_f\}_{f=1}^F$ by a set of random permutations, under the constraint that 60% of the point correspondences are correct. The parameter of our method were set to (V1) $\rho_0 = 10^{-6}$, $\eta = 1.01$; (V2) $\rho_0 = 10^{-2}, \eta = 1.01$, which reflects the better initialization of (V2).

Fig. 1(a) depicts typical convergence curves (observed over 10 random trials) corresponding to the two initialization cases. We report the objective function value ($\frac{1}{2}\|\mathbf{C}\|_F^2 + \lambda_1\|\mathbf{L}_d\|_* + \lambda_2\|\mathbf{E}_c\|_1 + \lambda_3\|\mathbf{E}_d\|_1$) and the primal residuals[3] ($\|\mathbf{D}_c - \mathbf{L}_c - \mathbf{E}_c\|_F, \|\mathbf{L}_c - \mathbf{L}_c\mathbf{C}\|_F, \|\mathbf{D}_d - \mathbf{L}_d - \mathbf{E}_d\|_F, \|\mathbf{1}^T\mathbf{C} - \mathbf{1}^T\|_F$). Note that our algorithm converges to the same objective function value independently of initialization. Note also that better initialization leads to faster convergence. Here, the number of iterations reduces from 1800 to about 700. In terms of wall-clock time, this corresponds to a reduction from about 2 minutes to 46 seconds on a regular Core-i7 PC with 8GB of memory. In this experiment, the average $\mathrm{ACC}_{\mathrm{ms}}$ and $\mathrm{ACC}_{\mathrm{pc}}$ over the ten random tests were all 100% for both initializations. This shows that, starting from virtually no point correspondence information, our algorithm successfully recovers both the correct point matches (with 100% accuracy) and the correct motion segmentation results.

4.2 Experiment-2: Synthetic Data, with Noise and Outliers

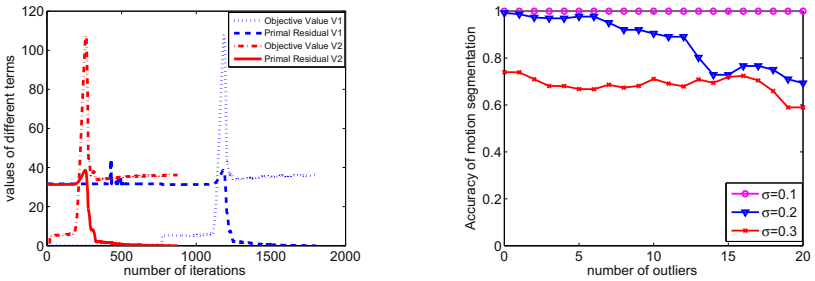
We then investigated the robustness of our algorithm to different amounts of outliers and noise, using synthetic data. In the same manner as above, we generated a 25-frame sequence of 40 points sampled from two independent motions, and drew feature vectors from i.i.d. Gaussian distributions. We then randomly permuted the point correspondences. We incrementally increased the number of outliers from 0 to 20 by adding gross errors to both the point coordinates and the feature descriptors. In addition to outliers, we also added fixed zero-mean Gaussian noise with standard deviation σ to the inliers.

Fig. 1(b) depicts the performance (averaged over 5 random trials) of our algorithm under different amounts of outliers and at fixed inlier noise level. In particular, we show the motion segmentation accuracy as a function of the number of outliers for 3 different levels of Gaussian noise. Note that the performance of our algorithm degrades gracefully as the amount of noise and outliers increases. Overall, our algorithm is rather robust to these adverse yet realistic conditions. For example, when the amount of measurement noise is moderate (i.e., σ from 0 to 0.1), our algorithm can almost achieve perfect motion segmentation results even for large number of outliers.

4.3 Experiment-3: Real Images, Hopkins155 Dataset

The Hopkins155 multibody dataset is a popular dataset for benchmarking motion segmentation methods. It contains 155 video sequences, which are however

[3] Here we plot the maximum value of the four primal residuals.



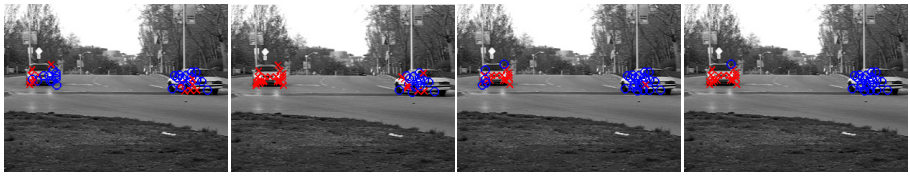
(a) (b)

Fig. 1. (a) Convergence curves for the objective function value and the primal residuals. Blue curves: initialization (V1); red curves: initialization (V2). (b) Accuracy of motion segmentation by adding different amounts of outliers and Gaussian noise.

all generated from 49 source video sequences. Among the 49 sequences, 28 are indoor scenes containing some checkerboard patterns, and the remaining 21 are outdoor natural scenes with no checkerboard pattern. While having a checkerboard pattern simplifies manual feature point matching, the repetitive pattern can actually confuse an appearance-based automatic feature matching algorithm. For example, SIFT descriptors computed at different corner points in a checkerboard are very similar. Since our algorithm makes use of appearance information (i.e., SIFT), we tested it on the 21 outdoor sequences first. More importantly, these 21 sequences are also much more realistic than the checkerboard ones. By dividing some of the sequences with three motions (g1g2g3) into subsequences with two motions(g1g2, g1g3, and g2g3), we obtained 27 sequences which cover most of the outdoor scenes in Hopkins155. The objective for this set of experiments on the real images of Hopkins155 is to verify the practical usefulness of our algorithm.

To create appearance observations, we computed a 128D SIFT descriptor at each one of the feature point locations provided with the Hopkins155 dataset. We then deliberately threw away all feature correspondence information. Given this input, our goal is to recover the missing point correspondences, and at the same time, to segment (cluster) all the feature points into correct motion groups.

Conventionally, when no point correspondences are given, motion segmentation is performed in a two-stage approach: (i) Point correspondences are estimated by, e.g., matching feature appearance or SIFT descriptors, and (ii) Subspace segmentation methods such as SSC [8] or LRR [13] are applied to the estimated point correspondences. Therefore, we employed the following methods as baselines: (1) SIFT matching followed by SSC, denoted by SIFT+SSC; (2) SIFT matching followed by LRR, denoted by SIFT+LRR; (3) ROML matching using embedded features [27,10] followed by SSC, denoted by RE+SSC; (4) ROML matching using embedded features [27,10] followed by LRR, denoted by RE+LRR; (5) ROML matching using SIFT feature [10,33] followed by SSC, denoted by RS+SSC; (6) ROML matching using SIFT feature followed by LRR, denoted by RS+LRR. Note that the embedded features combine the SIFT



(a) (b) (c) (d)

Fig. 2. Motion segmentation results of different algorithms for the cars2_07_g12 sequence. Points marked with the same color and marker ($\circ$ or $\times$) are from the same motion: (a) SIFT + SSC; (b) RE + SSC (RE: ROML with embedded feature); (c) RS + SSC (RS: ROML with SIFT feature); (d) Our algorithm. Best viewed in color.

features and the point coordinates by manifold learning [27], and were used for ROML in [10].

For ROML-based methods ((3)–(6)) and our algorithm, we initialized $\{\mathbf{P}_f\}_{f=1}^{F}$ with the PPMs recovered from SIFT matching, and set $\rho_0 = 10^{-2}, \eta = 1.01$. We tuned the respective parameters of baselines to achieve the best results, and empirically set $\lambda_1 = 1, \lambda_2 = 0.05, \lambda_3 = 5/\sqrt{N}$ in our algorithm.

In Table 1, we summarize the results in terms of ACC_{ms} and ACC_{pc}. From Table 1(a), we can see that our algorithm outperforms the baselines in terms of both correspondence and motion segmentation accuracies. Moreover, when given 100% complete trajectories, SSC and LRR achieve ACC_{ms} of 99.31% and 97.16%, respectively. This means that our algorithm, while not requiring any point correspondence as input, achieves motion segmentation results comparable to SSC and better than LRR.

Furthermore, when looking at the sequences whose ACC_{pc} of SIFT matching is less than 75% (see Table 1(b)), we find that our algorithm outperforms the baselines significantly in terms of both motion segmentation and point correspondence. For visual comparison, in Fig 2 we show the motion segmentation results of different algorithms on one sequence of Hopkins155.

Out of curiosity, we also performed experiments on the Hopkins155 checkerboard sequences. Due to the algorithmic complexity of ROML and of our method (mainly in the step solving the binary assignment problems), we selected the sequences with at most 200 trajectories. From the results in Table 2, we can see that our approach also outperforms the baselines on the checkerboard sequences. This shows that jointly solving motion segmentation and feature correspondence indeed helps compensating for the lack of discriminative appearance.

4.4 Experiment-4: Real Images, Other Real Sequences

Here, we show that our algorithm can be applied to perform motion segmentation in more realistic scenarios, where neither interest points nor point correspondences are provided.

Given video sequences with multiple motions, we first ran a SIFT detector over the frames to get the locations and descriptors of the detected interest points. Note that these interest points contain both inliers and outliers. The

Table 1. Average motion segmentation and point correspondence accuracies (%) on the Hopkins155 27 non-checkerboard sequences

Methods	SIFT+SSC	SIFT+LRR	RE+SSC	RE+LRR	RS+SSC	RS+LRR	Ours
(a) All 27 non-checkerboard sequences							
ACC_{ms}	84.83	80.10	88.47	88.47	93.03	91.30	**97.29**
ACC_{pc}	84.86	84.86	87.86	87.86	95.46	95.46	**98.03**
(b) The 6 sequences whose ACC_{PC} of SIFT matching is less than 75%							
ACC_{ms}	75.01	75.57	82.35	79.47	91.60	84.34	**99.59**
ACC_{pc}	64.09	64.09	75.73	75.73	86.70	86.70	**95.35**

Table 2. Average motion segmentation and point correspondence accuracies (%) on the Hopkins155 checkerboard sequences with at most 200 trajectories

Methods	SIFT+SSC	SIFT+LRR	RE+SSC	RE+LRR	RS+SSC	RS+LRR	Ours
ACC_{ms}	60.86	60.25	75.68	63.14	78.68	66.07	**83.24**
ACC_{pc}	32.27	32.37	46.85	46.85	53.76	53.76	**65.02**

number of inlier points can be empirically approximated as $N = N_{\text{sift}} - 10$, where N_{sift} is the minimum number of SIFT matches from the first frame to any other frame. We then ran our algorithm to automatically select the inlier points, establish correspondences between them, and segment the trajectories into their respective motions.

We tested our algorithm on the airport sequence taken from the airport motion segmentation dataset [6]. Fig. 3 shows the results on three frames sampled from the 40-frame sequence. The inlier points are marked differently and each type of marker corresponds to one motion. Note that in each frame, 75-125 interest points were automatically extracted by the SIFT detector, and only 21 points are set as inliers, i.e., the number of outliers is 2-4 times as large as that of inliers. Moreover, the measurements of coordinates and descriptors are contaminated by noise due to the illumination variations across the sequence. As challenging as this sequence is for the task of motion segmentation, after manually labeling

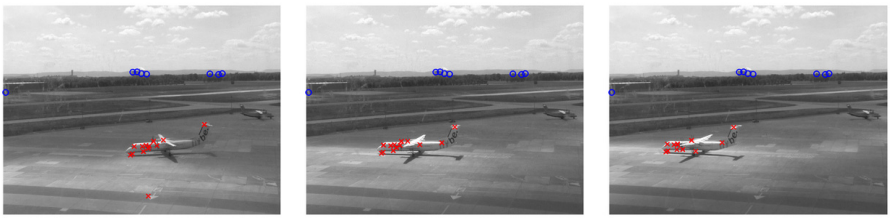


Fig. 3. Motion segmentation results of the airport sequence: Points marked with the same color and marker ($\circ$ or $\times$) are from the same motion. Best viewed in color.

the ground-truth correspondences for evaluation purpose, we found that our algorithm yields $\text{ACC}_{\text{ms}} = 96.43\%$ and $\text{ACC}_{\text{pc}} = 94.17\%$.

5 Conclusions

In this paper, we have proposed a unified framework to solve the problem of motion segmentation with unknown point correspondences. Our problem formulation is based on two important constraints: First, the recovered inlier point trajectories should satisfy the subspace constraints; Second, the matching feature descriptors should be low-rank, ideally rank one. With these two constraints, we have formulated our problem in terms of PPMs, which simultaneously select and reorder the inlier points. We have shown that our problem formulation can be solved via the ADMM. We have verified the effectiveness and robustness of our algorithm on both synthetic and real-world data and showed that it outperforms the existing two steps methods in terms of both motion segmentation and point correspondence accuracies. Our future work will focus on developing more efficient algorithms to solve the binary assignment problem using the inherent spatial constraints of motion segmentation.

Appendix

Solving (17) via the Hungarian Algorithm

From Problem (17), it is not straightforward to get the cost matrix for the Hungarian algorithm. Note, however, that we have $\text{vec}(\mathbf{W}_f\mathbf{P}_f) = (\mathbf{I}\otimes\mathbf{W}_f)\text{vec}(\mathbf{P}_f)$, $\text{vec}(\mathbf{T}_f\mathbf{P}_f) = (\mathbf{I}\otimes\mathbf{T}_f)\text{vec}(\mathbf{P}_f)$, where $\otimes$ is the Kronecker product. Let us define $\mathbf{G}_f = \mathbf{I}\otimes\mathbf{W}_f$, $\mathbf{a}_f = \text{vec}[\alpha_f^T(\mathbf{L}^{t+1} + \mathbf{E}_1^{t+1} - \mathbf{Y}_1^t/\rho^t)]$, $\mathbf{J}_f = \mathbf{I}\otimes\mathbf{T}_f$, and $\mathbf{b}_f = (\mathbf{M}^{t+1} + \mathbf{E}_2^{t+1} - \mathbf{Y}_3^t/\rho^t)\mathbf{e}_f$. Then Problem (17) becomes

$$\begin{aligned} &\min_{\mathbf{P}_f} \|\mathbf{G}_f\text{vec}(\mathbf{P}_f) - \mathbf{a}_f\|_2^2 + \|\mathbf{J}_f\text{vec}(\mathbf{P}_f) - \mathbf{b}_f\|_2^2 \\ &\text{s.t. } \mathbf{1}^T\mathbf{P}_f = \mathbf{1},\ \mathbf{P}_f\mathbf{1} \le \mathbf{1},\ \mathbf{P}_f \in \{0,1\}^{N_f\times N}\ . \end{aligned} \tag{23}$$

Therefore the assignment cost $\mathbf{Q}_f^1$ (or $\mathbf{Q}_f^2$) corresponding to the first (or second) term in (23) is the squared Euclidean distance between each column of $\mathbf{G}_f$ (or $\mathbf{J}_f$) and $\mathbf{a}_f$ (or $\mathbf{b}_f$). The total assignment cost is then $\mathbf{Q}_f = \mathbf{Q}_f^1 + \mathbf{Q}_f^2$. With this assignment cost, Problem (17) becomes solvable via the Hungarian algorithm.

Acknowledgements. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. The research is funded in part by ARC: DP120103896, DP130104567, LP100100588, DE140100180, CE140100016.

References

1. Amiaz, T., Kiryati, N.: Piecewise-smooth dense optical flow via level sets. IJCV 68(2), 111–124 (2006)
2. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. IEEE TPAMI 33(3), 500–513 (2011)
3. Cai, J., Candes, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optimization 20(4), 1956–1982 (2010)
4. Cheriyadat, A.M., Radke, R.J.: Non-negative matrix factorization of partial track data for motion segmentation. In: IEEE ICCV, pp. 865–872 (2009)
5. Cremers, D., Soatto, S.: Motion competition: A variational approach to piecewise parametric motion segmentation. IJCV 62(3), 249–265 (2005)
6. Dragon, R., Ostermann, J., Van Gool, L.: Robust realtime motion-split-and-merge for motion segmentation. In: Weickert, J., Hein, M., Schiele, B. (eds.) GCPR 2013. LNCS, vol. 8142, pp. 425–434. Springer, Heidelberg (2013)
7. Duchenne, O., Bach, F., Kweon, I., Ponce, J.: A tensor-based algorithm for high-order graph matching. In: IEEE CVPR, pp. 1980–1987 (2009)
8. Elhamifar, E., Vidal, R.: Sparse subspace clustering: Algorithm, theory, and applications. IEEE TPAMI 35(11), 2765–2781 (2013)
9. Ji, P., Salzmann, M., Li, H.: Efficient dense subspace clustering. In: IEEE WACV, pp. 461–468 (2014)
10. Jia, K., Chan, T.H., Zeng, Z., Ma, Y.: ROML: A robust feature correspondence approach for matching objects in a set of images. arXiv:1403.7877 (2014)
11. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: IEEE ICCV, pp. 1–8 (2005)
12. Li, H.: Two-view motion segmentation from linear programming relaxation. In: IEEE CVPR, pp. 1–8 (2007)
13. Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y.: Robust recovery of subspace structures by low-rank representation. IEEE TPAMI 35(1), 171–184 (2013)
14. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
15. Lu, C.-Y., Min, H., Zhao, Z.-Q., Zhu, L., Huang, D.-S., Yan, S.: Robust and efficient subspace segmentation via least squares regression. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 347–360. Springer, Heidelberg (2012)
16. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV 60(1), 63–86 (2004)
17. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of SIAM 5(1), 32–38 (1957)
18. Ochs, P., Malik, J., Brox, T.: Segmentation of moving objects by long term video analysis. IEEE TPAMI 36(6), 1187–1200 (2014)
19. Oliveira, R., Costeira, J., Xavier, J.: Optimal point correspondence through the use of rank constraints. In: IEEE CVPR, pp. 1–6 (2005)
20. Shen, Y., Wen, Z., Zhang, Y.: Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. Optimization Methods and Software 29(2), 239–263 (2014)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE TPAMI 22(8), 888–905 (2000)
22. Shi, J., Tomasi, C.: Good features to track. In: IEEE CVPR, pp. 593–600 (1994)

23. Sivic, J., Schaffalitzky, F., Zisserman, A.: Object level grouping for video shots. IJCV 67(2), 189–210 (2006)
24. Sun, D., Sudderth, E.B., Black, M.J.: Layered segmentation and optical flow estimation over time. In: IEEE CVPR, pp. 1768–1775 (2012)
25. Volz, S., Bruhn, A., Valgaerts, L., Zimmer, H.: Modeling temporal coherence for optical flow. In: IEEE ICCV, pp. 1116–1123 (2011)
26. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. IJCV 9(2), 137–154 (1992)
27. Torki, M., Elgammal, A.: One-shot multi-set non-rigid feature-spatial matching. In: IEEE CVPR, pp. 3058–3065 (2010)
28. Torresani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 596–609. Springer, Heidelberg (2008)
29. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: IEEE CVPR, pp. 1–8 (2007)
30. Vidal, R., Favaro, P.: Low rank subspace clustering (LRSC). Pattern Recognition Letters 43, 47–61 (2014)
31. Yan, J., Pollefeys, M.: A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 94–106. Springer, Heidelberg (2006)
32. Zelnik-Manor, L., Irani, M.: Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. In: IEEE CVPR (2003)
33. Zeng, Z., Chan, T.H., Jia, K., Xu, D.: Finding correspondence from multiple images via sparse and low-rank decomposition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 325–339. Springer, Heidelberg (2012)
34. Zhang, Y.: An alternating direction algorithm for nonnegative matrix factorization. Tech. rep., Rice University (2010)