Dynamics Enhanced Multi-Camera Motion Segmentation from Unsynchronized Videos*

Xikang Zhang, Bengisu Ozbay, Mario Sznaier, Octavia Camps
Electrical and Computer Engineering
Northeastern University, Boston MA 02115, US
zhangxk@ece.neu.edu, ozbay.b@husky.neu.edu, {msznaier,camps}@coe.neu.edu

Abstract

This paper considers the multi-camera motion segmentation problem using unsynchronized videos: given two video clips containing several moving objects, captured by unregistered, unsynchronized cameras with different viewpoints, our goal is to assign features to moving objects in the scene. This problem challenges existing methods, due to the lack of registration information and correspondences across cameras. To solve it, we propose a method that combines shape and dynamical information and does not require spatio-temporal registration or shared features. As shown in the paper, this combination results in improved performance even in the single camera case, and allows for solving the multi-camera segmentation problem with a computational cost similar to that of existing single-view techniques.

1. Introduction

Motion segmentation using data collected with a single camera has been extensively studied in the past decade, [7, 10, 15, 17, 18]. A large portion of the recent work in the area is based on the fact, noted in [3], that trajectories from k (rigid) motions lie in a union of subspaces (each of dimension at most 4), embedded in $\mathbb{R}^{2F}$, where F denotes the number of frames, allowing for reducing the motion segmentation problem to subspace clustering. While most of the techniques for solving the later problem require solving a regularized optimization problem, [7] has shown that a very efficient, robust algorithm for subspace clustering can be obtained by simply considering a Robust Shape Interaction Matrix (RSIM), obtained by row-normalization and exponentiation of the Shape Interaction Matrix introduced in [3]. Alternatively, affinity based methods, apply spectral clustering to an affinity matrix directly computed from the point trajectories [1, 9, 14]. Finally, [13] proposed using a dynamics motivated similarity matrix, where the distance between two points was given by the order of the model needed to jointly explain the corresponding trajectories.

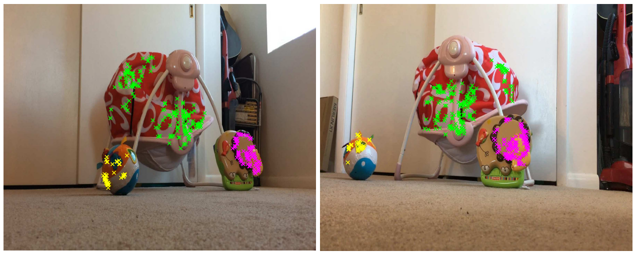

Figure 1: Multi-camera motion segmentation is challenging due to the lack of feature correspondences. We propose a new correspondence-free method to tackle this problem.

The methods above work well for the single camera case, even in the presence of outliers and missing data, On the other hand, these techniques fail in the unsynchronized multi-camera scenario of interest here, since in this case the underlying assumption that points from the same motion lie on a low order subspace no longer holds. In principle, this scenario could be handled by performing independent segmentations in each camera and then seeking correspondences across cameras using additional information. For instance, [19] exploits shape information. However, this requires viewing a set of point features in both cameras. In [5] cross-camera trajectory labeling is achieved by considering all possible assignments and selecting the one that minimizes the cost in a non-linear least squares problem. While this approach works well for scenes containing one or two moving objects, the cost of solving the non-linear least squares problem is not trivial, and overall complexity increases combinatorially with the number of objects.

Motivated by these difficulties, in this paper we propose a new approach that exploits both shape and dynamical information and does not require spatio-temporal registration

*This work was supported in part by NSF grants IIS–1318145, ECCS–1404163, CMMI–1638234; AFOSR grant FA9550-15-1-0392; and the Alert DHS COE under Award Number 2013-ST-061-ED0001.

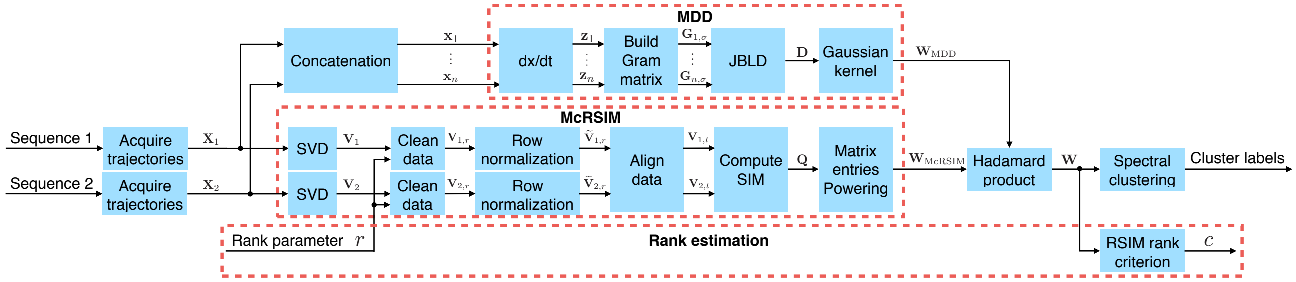

Figure 2: Diagram of the proposed method. The main idea is to combine information from a Multi-Camera Shape Interaction Matrix (McRSIM) and a Manifold Dynamic Distance (MDD) matrix, obtained using the Jensen-Bregman LogDet Divergence (JBLD), into a single affinity matrix $\mathbf{W}$. The cluster labels are obtained by performing spectral clustering on this matrix.

or shared features. Indeed, the only assumption used by the method is that the number of moving objects is invariant across cameras. The main observation motivating the proposed method is that, under affine camera models, the dynamical models underlying the motion of each rigid body are invariant under spatial rotation/translations and temporal offsets. Thus, a suitable defined distance between these models can be used, combined with a spectral clustering approach to assign points to objects, even if these points are observed by different cameras, not necessarily synchronized. As we show in the paper, such a distance can be efficiently computed by considering the manifold distance between Gram matrices corresponding to given trajectories, computed on the symmetric positive definite (SPD) matrix manifold. Since the "manifold dynamic distance" (MDD) feature is complementary to the shape feature used in RSIM (essentially based on geometric considerations, rather than dynamics), both can be combined to obtain an improved segmentation, even in the single camera case. While in principle the multi-camera case can be handled by considering this new feature alone, we show that a suitable multi-camera extension of RSIM (McRSIM) can be obtained by aligning the principal directions of the shape matrix across cameras. When combined with the new dynamics feature, McRSIM leads to a multi-camera robust motion segmentation algorithm that substantially outperforms approaches relying on dynamical or geometric information alone.

Paper contributions:

- **Single camera case:** We introduce a new feature, SPD manifold dynamic distance between trajectories (MDD), and show that, when combined with existing geometric-based features, the resulting algorithm outperforms the state of the art. This is illustrated using the Hopkins 155 data set.

- **Multiple camera case:**

  - We show that the MDD feature is invariant to affine transformations and time delays and thus can directly be used to perform multi-camera motion segmentation with unsynchronized cameras under a very mild condition: The same number of rigid motions must be observed across video sequences. There is no need for any number of points to be visible in both cameras or restrictions on the view-point difference or time offset.

  - We introduce a multi-camera generalization of the shape interaction matrix (McRSIM), obtained by considering velocities, rather than positions, and aligning the principal directions of the shape matrix. For a given set of points, McRSIM is also invariant to affine spatial transformations and time delays.

  - We propose a multi-camera motion segmentation algorithm based upon performing normalized spectral clustering on an affinity matrix obtained by combining the two features. Since its computational complexity is dominated by the complexity of performing a singular value decomposition on the data matrix, it can comfortably handle very large data sets. The advantages of this algorithm are illustrated both with a synthetic experiment using the Hopkins 155 data set, where, in each sequence half of the points are rotated/translated or delayed, and a new multi-camera data set, specifically created to benchmark this scenario.

2. Preliminaries

2.1. Notation

$\mathcal{S}^n$	set of symmetric matrices in $\mathbb{R}^{n\times n}$
$\mathcal{S}^n_+(\mathcal{S}^n_{++})$	set of positive-semidefinite (-definite) matrices in $\mathbb{R}^{n\times n}$
$\mathbf{x}(\mathbf{X})$	a vector (matrix) in $\mathbb{R}^n(\mathbb{R}^{n\times m})$
$\mathbf{X}(\succeq)\succ 0$	$\mathbf{X}$ is positive-(semi)definite
$\lvert\mathbf{X}\rvert$	determinant of the matrix $\mathbf{X}$
$\circ$	Hadamard matrix product $\mathbf{M}=\mathbf{X}\circ\mathbf{Y}$ has entries $m_{ij}=x_{ij}y_{ij}$.

2.2. Modeling 3D Rigid Motion

As shown in [13], the 3D coordinates at time t (in the camera frame), $\mathbf{P}(t) = [X(t)\ Y(t)\ Z(t)]^T$ of a point P on an object are related to their past values by an ARMA model of the form $\mathbf{P}(t) = \sum_{i=1}^{n} a_i \mathbf{P}(t-i)$, for some n large enough. Here a_i are scalars that depend only on the motion, and can be estimated, together with n, from the experimental data by considering the Hankel matrix associated with the trajectories of the point defined by:

$$\mathbf{H}_n(\mathbf{P}) = \begin{bmatrix} \mathbf{P}(1) & \mathbf{P}(2) & \cdots & \mathbf{P}(n+1) \\ \mathbf{P}(2) & \mathbf{P}(3) & \cdots & \mathbf{P}(n+2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}(n) & \mathbf{P}(n+1) & \cdots & \mathbf{P}(2n) \end{bmatrix} \qquad (1)$$

Specifically, it can be shown (see for instance [16], Chapter 10) that n is the smallest integer such that $\mathbf{H}_n$ is rank deficient, and in that case its null space is spanned by a vector of the form $\mathbf{r} = \begin{bmatrix}\mathbf{a}^T & -1\end{bmatrix}^T$, where $\mathbf{a} \doteq \begin{bmatrix}a_n & a_{n-1} & \dots & a_1\end{bmatrix}^T$. This observation was exploited in [13] to perform single-camera motion segmentation by grouping points according to rank of the Hankel matrix of point-wise differences of time trajectories $\mathbf{P}_i(t) - \mathbf{P}_j(t)$.

2.3. A Dynamic Feature Invariant under Affine Transformations and Time Shifts

Suppose that each point in the object undergoes a transformation of the form $\mathbf{P} \to \tilde{\mathbf{P}} \doteq \mathbf{A}\mathbf{P} + \mathbf{t}$, where $\mathbf{A}$ and $\mathbf{t}$ are, a given affinity matrix and translation vector, respectively. Assume that the 3D points are viewed using an orthographic camera, that is the 2D coordinates of the points $\mathbf{p} = \begin{bmatrix}x & y\end{bmatrix}^T$ are given by $x(t) = X(t), y(t) = Y(t)$. Let $\mathbf{v}$ and $\tilde{\mathbf{v}}$ denote the corresponding 2D velocities, that is, $\mathbf{v}(t) \doteq \mathbf{p}(t) - \mathbf{p}(t-1)$ and $\tilde{\mathbf{v}}(t) \doteq \tilde{\mathbf{p}}(t) - \tilde{\mathbf{p}}(t-1)$ Since $\mathbf{v}(t) = \mathbf{p}(t) - \mathbf{p}(t-1) = \mathbf{\Pi}\left[\mathbf{P}(t) - \mathbf{P}(t-1)\right]$, and $\tilde{\mathbf{v}}(t) = \mathbf{\Pi}\mathbf{A}\left[\mathbf{P}(t) - \mathbf{P}(t-1)\right]$, where $\mathbf{\Pi} \doteq \begin{bmatrix}1 & 0 & 0 \\ 0 & 1 & 0\end{bmatrix}$, it follows that $\mathbf{v}$ and $\tilde{\mathbf{v}}$ satisfy:

$$\mathbf{v}(t) = \sum_{i=1}^{n} a_i \mathbf{v}(t-i), \quad \tilde{\mathbf{v}}(t) = \sum_{i=1}^{n} a_i \tilde{\mathbf{v}}(t-i) \qquad (2)$$

From the discussion above, it follows that $\mathbf{H}_n(\mathbf{v})$ and $\mathbf{H}_n(\tilde{\mathbf{v}})$ share the same (right) null space, spanned by the vector $\mathbf{r} = \begin{bmatrix}\mathbf{a}^T & -1\end{bmatrix}^T$, that is, the (right) null space of the (velocity) Hankel matrix is invariant to affine + translational transformations. Similarly, since the sequence $\mathbf{v}(t)$ and its time delayed version $\mathbf{v}(t-\tau)$ both satisfy (2), it follows that the corresponding Hankel matrices share the same right null space. These observations are summarized next:

Theorem 1. *Consider the 3D trajectory of a moving point $\mathbf{P}(t) \doteq \begin{bmatrix}X(t) & Y(t) & Z(t)\end{bmatrix}^T$ and let $\mathbf{p}(t)$ denote its 2D orthographic projection. Then, the right null space of the Hankel matrix $\mathbf{H}_n(\mathbf{v})$ obtained from the 2D velocities $\mathbf{v}(t) \doteq \mathbf{p}(t) - \mathbf{p}(t-1)$ is invariant under 3D affine transformations, translations and time delays.*

2.4. Comparing Trajectories

From the discussion above, it follows that points undergoing the same motion will lead to Hankel matrices with the same null space, even if the trajectories are observed by cameras with different viewpoints and a temporal offset. Thus, in principle, 2D points could be assigned to "motions"[1] comparing the subspace angles between the null spaces of the corresponding (velocity) Hankel matrices. However, a potential difficulty here is that computing these angles requires estimating the rank of these matrices, a difficult task in the presence of noise. To circumvent this difficulty, motivated by the work of [20], in this paper, we will use Gram (rather than Hankel) matrices defined as $\mathbf{G}_n(\mathbf{p}) \doteq \mathbf{H}_n(\mathbf{p})^T \mathbf{H}_n(\mathbf{p})$[2]. The advantage of using Gram matrices, is that, with a suitable regularization, they can be embedded in the SPD manifold and compared using a number of manifold metrics, without the need for rank estimation, as shown in the following theorem, adapted from [20]:

Theorem 2. *Given Gram matrices $\mathbf{G}_1, \mathbf{G}_2$, define regularized matrices $\mathbf{G}_{1,\sigma} \doteq \mathbf{G}_1/\|\mathbf{G}_1\|_F + \sigma\mathbf{I}$, $\mathbf{G}_{2,\sigma} \doteq \mathbf{G}_2/\|\mathbf{G}_2\|_F + \sigma\mathbf{I}$, where $\sigma > 0$. Then $\lim_{\sigma\to 0} \delta(\mathbf{G}_{1,\sigma}, \mathbf{G}_{2,\sigma}) \neq \infty$ if and only if the corresponding Hankel matrices $\mathbf{H}_1$, $\mathbf{H}_2$ have the same (right) null space, where $\delta(.,.)$ denotes a suitable metric in $\mathcal{S}_{++}$.*

Motivated by computational efficiency considerations, in this paper we will use as manifold metric the Jensen-Bregman LogDet Divergence (JBLD) defined by:

$$\delta^2_{\text{ld}}(\mathbf{X}, \mathbf{Y}) = \log\left|\frac{\mathbf{X}+\mathbf{Y}}{2}\right| - \frac{1}{2}\log|\mathbf{X}\mathbf{Y}| \qquad (3)$$

This metric offers a good compromise between efficiency (since computing it does not require performing singular value decompositions) and approximating the Affine Invariant Riemannian Metric, a true geodesic metric in $\mathcal{S}_{++}$.

[1] Note that the null space of Hankel matrices alone cannot distinguish between objects undergoing exactly the same motion. This issue will be addressed in Section 3.

[2] Note that this is different from the definition used in [20], $\mathbf{G} \doteq \mathbf{H}\mathbf{H}^T$. Since $\mathbf{H} = \mathbf{K}\mathbf{X}$, where $\mathbf{K}$ depends only on the coefficients a_i and $\mathbf{X}$ depends on the initial conditions, the choice $\mathbf{H}^T\mathbf{H}$ leads to matrices whose null-space is independent of the initial conditions, a key feature when seeking to group points that share the same motion model but not necessarily the same initial conditions.

2.5. The Robust Shape Interaction Matrix

The seminal work in [3] showed that the motion segmentation problem can be solved by considering the shape interaction matrix $\mathbf{Q} = \mathbf{V}_r\mathbf{V}_r^T$, where $\mathbf{V}_r$ denotes the first r right singular vectors of the data matrix $\mathbf{X}$ and $r = 4\times$(number of independent motions). In the ideal case, $q_{i,j} \neq 0$ if and only if the pair of points (i,j) belong to the same motion. Thus, for clean data, motion segmentation can be solved by block-diagonalization. However, as proposed, the shape interaction matrix has an intrinsic bias [7], since the intra-class affinity depends on the magnitude of data points (e.g. points closer to the origin have a smaller value than those farther away), leading to relatively poor performance in the presence of noise and outliers. To address this issue, [7] proposed using a Robust Shape Interaction Matrix, obtained by row normalization of $\mathbf{V}_r$, to avoid the magnitude bias noted above, followed by element-wise powering. This last step denoises the affinity matrix, by suppressing small elements while leaving larger elements with values closer to 1 relatively unchanged. As shown in [7], combining the robustified SIM with spectral clustering leads to a subspace clustering algorithm that outperforms competing methods, with lower computational costs.

3. Dynamics Enhanced Single Camera Segmentation

From the discussion in Section 2.4 it follows that a suitable affinity matrix that takes into account dynamical information is given by $\mathbf{W}_{\text{ld}} = e^{-\mathbf{D}/d_{\max}}$, where the matrix $\mathbf{D} \in \mathbb{R}^{n_p\times n_p}$ has entries $d_{i,j} = \delta^2_{\text{ld}}(\mathbf{G}_{i,\sigma}, \mathbf{G}_{j,\sigma})$ and $d_{\max}$ is its largest entry. Here $\mathbf{G}_i$ denotes the Gram matrix obtained from the 2D velocities of the point $\mathbf{p}_i$ and σ is a design parameter. Note that, from Theorem 2, it follows that, as $\sigma \to 0$, ideally the matrix $\mathbf{W}$ will have $w_{i,j} = 0$ for points corresponding to different motions. However, on the other hand σ should not be taken too small, to avoid numerical problems. A good compromise is to take σ on the order of magnitude of the estimated noise covariance. Note that $\mathbf{W}_{\text{ld}}$ takes into account only dynamic information. An affinity matrix that considers both the dynamics and the geometry of the scene can be obtained by simply defining the combined affinity matrix $\mathbf{W} \doteq \mathbf{W}_{\text{ld}} \circ \mathbf{W}_{\text{RSIM}}$, where the later is the RSIM matrix defined in Section 2.5. The complete single camera algorithm is outlined in Algorithm 1.

4. The Multi-Camera Case

Next, we present the main result of the paper, an algorithm capable of handling multiple cameras and time offsets. Since the matrix $\mathbf{W}_{\text{ld}}$ is view-point and time-delay invariant, in principle one could just use it in combination with spectral clustering to segment the motions. However, as noted in section 2.4, such an approach cannot distinguish objects with the same dynamics. To circumvent this difficulty, next we introduce a modified RSIM matrix that takes into account multiple views and hence can be combined with $\mathbf{W}_{\text{ld}}$ proceeding in the spirit of Algorithm 1.

Algorithm 1 RSIM-MDD: Single Camera Motion Segmentation.

```
Input: $\mathbf{X} \in \mathbb{R}^{3f\times n}$, a matrix containing all trajectories
    in a video sequence with homogeneous coordinates, $k$,
    the number of motions; $r_{\max}$ and $r_{\min}$, upper and lower
    bound of rank $r$; $\gamma$, the power parameter.
$s \leftarrow 0$
for $r := r_{\min}$ to $r_{\max}$ do
    $s \leftarrow s + 1$
    (Compute RSIM affinity matrix $\mathbf{W}_{\text{RSIM}}$)
    $[\mathbf{U},\mathbf{S},\mathbf{V}] \leftarrow \text{SVD}(\mathbf{X})$
    $\mathbf{V}_r \leftarrow \mathbf{V}(:, 1:r)$
    $\widetilde{\mathbf{V}}_r \leftarrow$ Normalize each row of $\mathbf{V}_r$
    $\mathbf{W}_{\text{RSIM}} \leftarrow \widetilde{\mathbf{V}}_r\widetilde{\mathbf{V}}_r^T$
    $\mathbf{W}_{\text{RSIM}}(p,q) \leftarrow (\mathbf{W}_{\text{RSIM}}(p,q))^{\gamma}, \forall p,q$
    (Compute MDD affinity matrix $\mathbf{W}_{\text{MDD}}$)
    for $j := 1$ to $n$ do
        $\mathbf{Y} \leftarrow$ reshape $\mathbf{X}(:,j)$ into $3\times f$ matrix
        $\mathbf{Y} \leftarrow$ remove 3rd row of $\mathbf{Y}$ which are all ones
        velocity $\mathbf{A} \leftarrow \mathbf{Y}(:, 2:\text{end}) - \mathbf{Y}(:, 1:\text{end}-1)$
        $\mathbf{H}_{\mathbf{A},j} \leftarrow$ Hankelize $\mathbf{A}$
        $\mathbf{G}_j \leftarrow \mathbf{H}_{\mathbf{A},j}\mathbf{H}_{\mathbf{A},j}^T$
        $\mathbf{G}_{j,\sigma} \leftarrow \mathbf{G}_j/\|\mathbf{G}_j\|_F + \sigma\mathbf{I}^{r,r}$
    end for
    $\mathbf{D}(p,q) \leftarrow \delta^2_{ld}(\hat{\mathbf{G}}_{p,\sigma}, \hat{\mathbf{G}}_{q,\sigma}), \forall p,q$
    $\mathbf{W}_{\text{MDD}}(p,q) = \exp(-\mathbf{D}(p,q)/\max(\mathbf{D})), \forall p,q$
    (Compute combined affinity matrix $\mathbf{W}$)
    $\mathbf{W}_s \leftarrow \mathbf{W}_{\text{RSIM}} \circ \mathbf{W}_{\text{MDD}}$
    Labels $\mathbf{z}_s \leftarrow$ spectral clustering on $\mathbf{W}_s$
    $c(s) \leftarrow \frac{\text{minCut}\{\mathbf{X}_1,\mathbf{X}_2,\cdots,\mathbf{X}_k\}}{\lambda_k - \lambda_{k+1}}$
end for
$\hat{s} \leftarrow \text{argmin}_s c(s)$
Output: $\mathbf{z}_{\hat{s}}$
```

4.1. A Multi-Camera Shape Interaction Matrix

Suppose that the same set of moving points is observed from two (fixed) cameras whose 3D positions are related by a rotation $\mathbf{R}_c$ and a translation $\mathbf{t}_c$. Let $\mathbf{s}_i$ denote the 3D homogeneous coordinates of the i^{th} point in the object reference frame and $\mathbf{P}_{i,j}(k),\ j = 1,2$ its 3D (homogeneous) coordinates in coordinate system of each camera at time k. Then

$$\mathbf{P}_{i,1}(k) = \begin{bmatrix} \mathbf{R}(k) & \mathbf{t}(k) \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{s}_i \tag{4}$$

and

$$\begin{aligned}\mathbf{P}_{i,2}(k) &= \begin{bmatrix}\mathbf{R}_c & \mathbf{t}_c\\ \mathbf{0} & 1\end{bmatrix}\mathbf{P}_{i,1}(k)\\ &= \begin{bmatrix}\mathbf{R}_c & \mathbf{t}_c\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}(k) & \mathbf{t}(k)\\ \mathbf{0} & 1\end{bmatrix}\mathbf{s}_i \doteq \begin{bmatrix}\tilde{\mathbf{R}}(k) & \tilde{\mathbf{t}}(k)\\ \mathbf{0} & 1\end{bmatrix}\mathbf{s}_i\end{aligned} \tag{5}$$

Thus, it follows that the corresponding 2D data matrices in coordinate frames of each camera satisfy

$$\mathbf{X}_1 = \mathbf{M}\mathbf{S},\ \mathbf{X}_2 = \tilde{\mathbf{M}}\mathbf{S} \tag{6}$$

where $\mathbf{S}$ is a matrix whose i^{th} column is $\mathbf{s}_i$ and the motion matrices $\mathbf{M}, \tilde{\mathbf{M}}$ have (block) rows of the form:

$$\mathbf{M}_k = \mathbf{\Pi}\begin{bmatrix}\mathbf{R}(k) & \mathbf{t}(k)\end{bmatrix} \text{ and } \tilde{\mathbf{M}}_k = \mathbf{\Pi}\begin{bmatrix}\tilde{\mathbf{R}}(k) & \tilde{\mathbf{t}}(k)\end{bmatrix}$$

Consider now the case of K moving objects, with the same set of points observed by the two cameras. From this discussion, it follows that as long as $rank(\mathbf{X}_1) = rank(\mathbf{X}_2) = 4K$, then their respective row spaces are spanned by the row space of $\mathbf{S}$, which, in turn is spanned by $\mathbf{V}_{r,1}^T$ and $\mathbf{V}_{r,2}^T$. Hence, given $\mathbf{V}_{r,1}$ and $\mathbf{V}_{r,2}$, there exists some orthogonal matrix $\mathbf{R}_v \in \mathbb{R}^{r\times r}$ such that $\mathbf{V}_{r,1} = \mathbf{V}_{r,2}\mathbf{R}_v$. It can be easily shown that one such matrix is given by

$$\mathbf{R}_v = \mathbf{V}_{r,2}^T\mathbf{V}_{r,1} \tag{7}$$

since

$$\begin{aligned}&\mathbf{V}_{r,1} = \mathbf{V}_{r,2}\mathbf{R}_v = \mathbf{V}_{r,2}\mathbf{V}_{r,2}^T\mathbf{V}_{r,1} =\\ &(\mathbf{I} - \mathbf{V}_{r,2}^{\perp}(\mathbf{V}_{r,2}^{\perp})^T)\mathbf{V}_{r,1} = \mathbf{V}_{r,1}\end{aligned} \tag{8}$$

where the last equality follows from the fact that $\mathbf{V}_{1,r}$ and $\mathbf{V}_{2,r}$ span the same subspace and hence $(\mathbf{V}_{2,r}^{\perp})^T\mathbf{V}_{1,r} = 0$. Note that this matrix is precisely the one that rotates the subspaces so that their corresponding angle becomes zero.

In the case where sets of points observed by two cameras are different, the derivation above no longer holds. Nevertheless, one would expect that there exists a matrix $\mathbf{R}_v$ closely aligning the subspaces spanned by $\mathbf{V}_{1,r}$ and $\mathbf{V}_{2,r}$. Intuitively, if there is enough information on each camera to determine the shape of the object, then it should be possible to (approximately) align the principal directions of the shapes, as viewed by each camera. In this case, $\mathbf{R}_v$ can be found by solving the following optimization problem:

$$\min_{\mathbf{R}} \|\mathbf{V}_{1,r} - \mathbf{V}_{2,r}\mathbf{R}\|_F \text{ subject to } \mathbf{R}^T\mathbf{R} = \mathbf{I} \tag{9}$$

where, if needed, the matrices $\mathbf{V}_{i,r}$ are padded with zero columns, so that the dimensions are compatible. It is not hard to show that the explicit solution for this problem is precisely given by (7). This reasoning suggests using the following multi-camera shape interaction matrix:

$$\mathbf{Q}_v \doteq \begin{bmatrix}\mathbf{V}_{1,r}\\ \mathbf{V}_{2,r}\mathbf{R}_v\end{bmatrix}\begin{bmatrix}\mathbf{V}_{1,r}^T & \mathbf{R}_v^T\mathbf{V}_{2,r}^T\end{bmatrix} \tag{10}$$

where $\mathbf{R}_v$ is given by (7). Note that $(1,1)$ and $(2,2)$ blocks of $\mathbf{Q}_v$ are precisely $\mathbf{Q}_1$ and $\mathbf{Q}_2$, the shape interaction matrices in each camera, while (1,2) and (2,1) blocks provide cross-camera shape information. Finally, a robust version of $\mathbf{Q}_v$ can be obtained by row normalization as in [7].

4.2. Dynamics Enhanced Multi-Camera Segmentation

The discussion in the previous section suggests that an affinity matrix combining multi-camera dynamic and geometric information is given by $\mathbf{W} \doteq \mathbf{W}_{\text{ld}} \circ \mathbf{W}_{\text{McRSIM}}$, where $\mathbf{W}_{\text{McRSIM}}$ is obtained from $\mathbf{Q}_v$ defined in (10) by row normalization and exponentiation. This motivates the multi-camera motion segmentation Algorithm 2.

Algorithm 2 Multi camera motion segmentation with McRSIM-MDD

```
Input: Data matrices $\mathbf{X}_i \in \mathbb{R}^{3f\times n_i}, i = 1,2,\cdots,m$
    where $n_i$ is the number of trajectories in the $i$th camera;
    $k$ (number of motions); $r_{\text{max}}$ and $r_{\text{min}}$ (upper and lower
    bound of $\frac{\text{rank}}{\text{number of motions}}$); $\gamma$ (the power parameter).
$s \leftarrow 0$
for $r := r_{\text{min}} \cdot k$ to $r_{\text{max}} \cdot k$ do
    (Compute McRSIM affinity matrix $\mathbf{W}_{\text{McRSIM}}$)
    for $i := 1$ to $m$ do
        $[\mathbf{U}_i, \mathbf{S}_i, \mathbf{V}_i] \leftarrow \text{SVD}(\mathbf{X}_i)$
        $\mathbf{V}_{i,r} \leftarrow \mathbf{V}_i(:, 1:r)$
        $\widetilde{\mathbf{V}}_{i,r} \leftarrow$ Normalize each row of $\mathbf{V}_{i,r}$
    end for
    $n_{\text{max}} \leftarrow \max\{n_1, n_2, \cdots, n_m\}$
    Suppose $n_l = n_{\text{max}}$, then
    $\mathbf{V}_{l,t} \leftarrow \mathbf{V}_{l,r}$
    for $i := 1$ to $m$ do
        if i != l then
            $\mathbf{M} \leftarrow \widetilde{\mathbf{V}}_{i,r}^T\widetilde{\mathbf{V}}_{l,r}(1:n_i,:)$
            $[\mathbf{U}_{\mathbf{M}}, \mathbf{S}_{\mathbf{M}}, \mathbf{V}_{\mathbf{M}}] \leftarrow \text{SVD}(\mathbf{M})$
            $\mathbf{R} \leftarrow \mathbf{U}_{\mathbf{M}}\mathbf{V}_{\mathbf{M}}^T$
            $\mathbf{V}_{i,t} = \mathbf{V}_{i,r}\mathbf{R}$
        end if
    end for
    $\mathbf{V}_t \leftarrow \begin{bmatrix}\mathbf{V}_{1,t}^T & \mathbf{V}_{2,t}^T & \cdots & \mathbf{V}_{m,t}^T\end{bmatrix}^T$
    $\mathbf{W}_{\text{McRSIM}} \leftarrow \mathbf{V}_t\mathbf{V}_t^T$
    $\mathbf{W}_{\text{McRSIM}}(p,q) \leftarrow (\mathbf{W}_{\text{McRSIM}}(p,q))^{\gamma}, \forall p, q$
    (Compute MDD affinity matrix $\mathbf{W}_{\text{MDD}}$)
    $\mathbf{W}_{\text{MDD}} \leftarrow$ Compute $\mathbf{W}_{\text{MDD}}$ as in Algorithm 1
    (Compute combined affinity matrix $\mathbf{W}$)
    $\mathbf{W}_s \leftarrow \mathbf{W}_{\text{McRSIM}} \circ \mathbf{W}_{\text{MDD}}$
    Labels $\mathbf{z}_s \leftarrow$ spectral clustering on $\mathbf{W}_s$
    $c(s) \leftarrow \frac{\text{minCut}\{\mathbf{X}_1,\mathbf{X}_2,\cdots,\mathbf{X}_k\}}{\lambda_k - \lambda_{k+1}}$,  $s \leftarrow s+1$
end for
$\hat{s} \leftarrow \text{argmin}_s c(s)$
Output: $\mathbf{z}_{\hat{s}}$
```

Algorithm 2 is robust to gross contamination due to the following reasons: Firstly, RSIM has built-in robustness [7]. Secondly, in the presence of gross contamination, the null space of the corrupted Gram matrix will be generally closer to the true one than to those corresponding to other motions. Hence the JBLD will yield a smaller distance to the true motion. In addition, it can be modified to handle missing data, as follows: (i) use RSIM-M [7] in lieu of RSIM and (ii) if the data in $[t_1, t_2]$ is missing, ignore it and form the Hankel matrices $\mathbf{H}_1, \mathbf{H}_2$ from trajectories in $[0, t_1-1]$ and $[t_2+1, F]$, respectively. The matrix $\mathbf{G}_m \doteq \mathbf{H}_1^T\mathbf{H}_1+\mathbf{H}_2^T\mathbf{H}_2$ has the same null space of the "complete" matrix $\mathbf{G}$ and thus can be used in its place in the algorithm.

For benchmarking purposes, we introduce Algorithm 3, where we first perform motion segmentation in each camera, followed by computing the Stein mean [2] of the regularized Gram matrices of all trajectories in a cluster. Cross-camera matching is posed as a linear assignment problem between these means and solved using the code from [8].

Algorithm 3 Two cameras motion segmentation with RSIM-MDD-LA

```
Input: Data matrices $\mathbf{X}_i \in \mathbb{R}^{3f\times n_i}, i = 1,2$ where $n_i$ is
    the number of trajectories in the $i$th camera; $k$ (number
    of motions); $r_{\text{max}}$ and $r_{\text{min}}$ (upper and lower bound of
    $\frac{\text{rank}}{\text{number of motions}}$); $\gamma$ (the power parameter).
for $i := 1$ to $2$ do
    $\mathbf{z}_i \leftarrow$ Perform RSIM-MDD algorithm on $\mathbf{X}_i$
    (Put together data in the same cluster)
    for $j := 1$ to $k$ do
        initialize a set $\mathbb{S}_j \leftarrow \emptyset$
    end for
    for $l := 1$ to $n_i$ do
        Compute $\hat{\mathbf{G}}_{l,\sigma}$
        $\mathbb{S}_{z_{il}} = \mathbb{S}_{z_{il}} \cup \{\hat{\mathbf{G}}_{l,\sigma}\}$
    end for
    (Get Stein mean from each cluster)
    for $j := 1$ to $k$ do
        $\mathbf{M}_{ij} \leftarrow$ get Stein mean of all elements in $\mathbb{S}_j$
    end for
end for
(Match cluster labels between two cameras)
$\mathbf{D}_{pq} = \delta_{\text{ld}}(\mathbf{M}_{1p}, \mathbf{M}_{2q}),\ \forall p, q = 1, 2, \cdots, k$
Matching index $\mathbf{y} \leftarrow$ linear assignment (LA) on $\mathbf{D}$
$\mathbf{z}_{2l} \leftarrow \mathbf{y}_{\mathbf{z}_{2l}},\ \forall l = 1, 2, \cdots, n_2$
Output: $\mathbf{z}_1, \mathbf{z}_2$
```

5. Experimental results

In this section, we illustrate the effectiveness of the proposed method, both in single and multi-camera scenarios. In all cases we used $\gamma = 3.5$ for all RSIM related methods.

5.1. Hopkins 155 data set

The results of this experiment (Table 1), illustrate the fact that combining dynamic and geometric information leads to improved results, even in the single camera case.

Table 1: % Clustering error on Hopkins 155 data set. (R)SIM means (Robust) Shape Interaction Matrix ([7])[3]; SSC means Sparse Subspace Clustering; [4]; LRR means Low Rank Representation[12]; LRR-H and LRR-H2 mean LRR with the heuristic [4] and [11]; EDSC means Efficient Dense Subspace Clustering [6]; EDSC-H means EDSC with Heuristic.

Methods	SIM	SSC	LRR	LRR-H	LRR-H2	EDSC	EDSC-H	RSIM	**RSIM-MDD**
2 motions									
Mean	6.50	1.53	4.10	2.13	1.33	2.67	0.86	0.78	**0.52**
Median	1.14	0.00	0.22	0.00	0.00	0.00	0.00	0.00	0.00
3 motions									
Mean	12.26	4.40	9.89	4.03	2.51	8.06	2.49	1.77	**1.55**
Median	6.12	6.22	0.56	1.43	0.00	2.53	0.21	0.28	0.25
Overall									
Mean	7.80	2.18	5.41	2.56	1.60	4.04	1.23	1.01	**0.75**
Median	1.53	0.00	0.53	0.00	0.00	0.30	0.00	0.00	0.00

5.2. Modified Hopkins 155 data set with rotation and translation

To test the proposed multi-camera segmentation algorithm in a scenario with a known ground truth, we performed an experiment where the Hopkins 155 data set was used to generate simulated multi-camera data. To this effect, we randomly selected, in each sequence, half of the trajectories. These trajectories were rotated 45° and moved to the right by 300 pixels and downward by 200 pixels to simulate data obtained from a second camera, rotated and translated with respect to the first. Note that in this scenario there are no common trajectories across cameras. The results of these experiments are shown in Table 2. In all cases {method}-MDD denotes using the affinity matrix obtained from the Hadamard product of the affinity matrix from {method} and MDD. The suffix LA means performing {method} on each view, computing the Stein mean of each cluster and using a Linear Assignment (LA) [8] to match cluster labels in different cameras. Note that here, the error rate of RSIM is much larger than in the single camera case, illustrating its difficulty in handling rotations and translations. On the other hand, these transformations are readily handled by the McRSIM-MDD method.

5.3. Modified Hopkins 155 data set with time delay

The goal here is to test the effect of a time delay between the two cameras. To simulate this situation, we randomly divided the trajectories of each sequence in the Hopkins 155 data set into two sets. Suppose the total number of frames is F and the time delay is τ frames. We picked Frames 1 to $F-\tau$ in the first set, and for Set 2, we picked Frames $\tau+1$

Table 2: Clustering error (in %) on Hopkins 155 data set with half its trajectories rotated 45° and translated 300 pixels in x axis and 200 pixels in y axis. Parameters are $r_{\min} = 1$, $r_{\max} = 4$ for all RSIM and McRSIM related methods.

Methods	SSC	SSC-MDD	SSC-MDD-LA	LRR-H2	LRR-H2-MDD	LRR-H2-MDD-LA	RSIM	RSIM-MDD	RSIM-MDD-LA	McRSIM	McRSIM-MDD
2 motions clustering error (in %)											
Mean	33.28	32.04	3.28	17.97	16.89	3.23	25.73	25.94	2.10	0.98	**0.83**
Median	43.46	42.71	0.00	15.54	12.31	0.00	30.82	31.82	0.00	0.00	0.00
3 motions clustering error (in %)											
Mean	44.29	43.89	8.20	26.33	25.76	6.33	30.18	27.03	2.34	2.40	**2.09**
Median	46.84	46.79	5.06	26.81	26.56	1.80	33.33	31.40	0.28	0.67	0.58
Overall clustering error (in %)											
Mean	35.77	34.72	4.39	19.86	18.90	3.93	26.74	26.18	2.15	1.30	**1.11**
Median	44.35	44.34	0.00	19.23	18.10	0.00	31.82	31.82	0.00	0.00	0.00
Running time (in seconds)											
Total	88.46	186.38	118.84	138.61	242.16	241.02	28.19	131.24	94.58	28.01	129.72
Avg	0.57	1.20	0.77	0.89	1.56	1.55	0.18	0.85	0.61	0.18	0.84

Table 3: Clustering error (in %) on Hopkins 155 data set with half its trajectories delayed 4 frames. Parameters are $r_{\min} = 1$, $r_{\max} = 4$ for all RSIM and McRSIM related methods.

Methods	SSC	SSC-MDD	SSC-MDD-LA	LRR-H2	LRR-H2-MDD	LRR-H2-MDD-LA	RSIM	RSIM-MDD	RSIM-MDD-LA	McRSIM	McRSIM-MDD
2 motions clustering error (in %)											
Mean	23.23	22.82	3.59	18.64	19.22	5.53	9.21	9.67	2.93	1.57	**0.94**
Median	26.11	25.86	0.00	7.74	12.22	0.00	0.00	0.00	0.00	0.00	0.00
3 motions clustering error (in %)											
Mean	35.34	34.84	8.96	31.66	31.98	11.12	28.53	27.82	4.43	2.22	**2.14**
Median	45.92	43.78	5.39	35.53	36.96	3.80	33.33	33.33	0.48	0.67	0.64
Overall clustering error (in %)											
Mean	25.96	25.54	4.80	21.58	22.10	6.79	13.57	13.78	3.27	1.71	**1.21**
Median	33.33	31.71	0.00	25.68	26.16	0.00	0.27	0.26	0.00	0.00	0.00
Running time (in seconds)											
Total	86.77	180.95	118.69	128.86	226.92	225.83	26.35	125.42	91.04	27.86	128.26
Avg	0.56	1.17	0.77	0.83	1.46	1.46	0.17	0.81	0.59	0.18	0.83

to F. Thus trajectories from Camera 2 have a delay of τ frames relative to those of Camera 1. As shown in Table 3, McRSIM combined with Gram JBLD has much lower error rate than other methods, illustrating its robustness to delays.

5.4. Multi-view motion segmentation data set

To test the proposed algorithms with real-world data, we created a new data set, RSL 12, consisting of 12 pairs of two camera sequences, captured using common phone cameras, and thus, unsynchronized. We used KLT trackers to extract trajectories from each video sequences and manually labeled each motion cluster. The clustering results, shown in Table 4, illustrate that, in multi-camera scenarios, the proposed McRSIM-MDD based method outperforms state-of-the-art clustering methods such as RSIM and SSC. It is worth nothing that in all cases, the use of single camera methods in each view, followed by a Linear Assignment using the MDD feature (in the spirit of Algorithm 3) yielded near optimal performance, illustrating the advantages of using dynamic information.

5.5. Hopkins 155 data set with gross contamination

To show that the proposed method is robust to gross contamination, we used a modified version of Hopkins 155 data set. From all tracked feature points in each video, we replaced 5% entries with random values from uniform distribution between -1 and 1 (Note that since we used normalized data, all entries are between -1 and 1). We ran each video 10 times with different random seed. Tables 5 and 6 show that the proposed method performs better than RSIM both in the single and multi-camera cases.

5.6. Hopkins 12 Real Sequences with Missing data

To show that the proposed method is robust to missing entries, we experimented on Hopkins 12 Real Sequences With Missing Data. As too-short trajectories do not contain enough dynamic information, we removed the trajectories whose number of visible entries is less than 5. Tables 7 and 8 indicate that the adapted method McRSIM-M-MDD performs best, both in single and multi-camera scenarios.

6. Conclusions

In this paper, we address the problem of multi-camera motion segmentation on possibly asynchronous videos. The proposed method exploits both shape and motion cues and does not require that the same features appear in both views.

Table 4: Clustering error (in %) on Multiview Motion RSL 12 data set. Parameters are $r_{\min} = 1$, $r_{\max} = 2$ for all RSIM and McRSIM related methods.

Methods	SSC	SSC-MDD	SSC-MDD-LA	LRR-H2	LRR-H2-MDD	LRR-H2-MDD-LA	RSIM	RSIM-MDD	RSIM-MDD-LA	McRSIM	McRSIM-MDD
2 motions clustering error (in %)											
Mean	2.78	2.78	**0.00**	9.54	9.40	**0.00**	8.72	8.68	**0.00**	0.03	**0.00**
Median	0.00	0.00	0.00	2.91	2.51	0.00	0.20	0.20	0.00	0.00	0.00
3 motions clustering error (in %)											
Mean	8.36	7.22	0.29	29.55	29.45	0.32	24.08	23.98	**0.00**	**0.00**	**0.00**
Median	2.70	2.70	0.00	33.29	33.29	0.00	32.41	32.41	0.00	0.00	0.00
Overall clustering error (in %)											
Mean	5.57	5.00	0.15	19.54	19.43	0.16	16.40	16.33	**0.00**	0.02	**0.00**
Median	0.95	0.95	0.00	25.70	25.70	0.00	7.72	7.41	0.00	0.00	0.00
Running time (in seconds)											
Total	24.46	53.58	29.55	19.22	51.80	42.42	3.25	35.18	20.50	3.03	34.56
Avg	2.04	4.46	2.46	1.60	4.32	3.53	0.27	2.93	1.71	0.25	2.88

Table 5: Clustering error (in %) on Hopkins 155 sequence with grossly contaminated entries (%5 corrupted).

%	RSIM	RSIM-MDD
2 motions		
Mean	17.22	**16.97**
Median	12.32	12.15
3 motions		
Mean	28.90	**28.78**
Median	25.63	25.53
Overall		
Mean	19.86	**19.64**
Median	17.99	17.71

Table 6: Clustering error (in %) on Hopkins 155 sequence with grossly contaminated entries (%5 corrupted). Half the trajectories are rotated 45° and serve as Camera 2.

Methods	RSIM	RSIM-MDD	McRSIM	McRSIM-MDD
2 motions				
Mean	31.76	31.39	21.79	**21.64**
Median	34.66	34.05	22.52	22.19
3 motions				
Mean	45.05	44.77	36.13	**36.02**
Median	47.42	47.32	35.66	35.41
Overall				
Mean	34.76	34.41	25.03	**24.89**
Median	38.67	38.19	25.95	25.97

Table 7: Clustering error (in %) on Hopkins 12 Real Motion Sequences With Incomplete Data.

%	RSIM-M	RSIM-M-MDD
Mean	0.69	**0.61**
Median	0.70	0.64
Max	1.74	1.64
Std	0.59	0.51

Table 8: Clustering error (in %) on Hopkins 12 Real Motion Sequences With Incomplete Data. Half of the trajectories are rotated 45° and serve as Camera 2.

Methods	RSIM-M	RSIM-M-MDD	McRSIM-M	McRSIM-M-MDD
Mean	35.78	35.58	7.45	**7.15**
Median	38.06	37.96	1.89	1.69
Max	47.40	47.63	31.83	31.83
Std	11.63	11.45	11.45	11.58

We introduce a new feature, MDD, as a dynamics based comparison of point trajectories which is invariant to affine transformations and time delays and is complementary to geometric based features used in most of the common motion segmentation methods. Additionally, we propose a multi-camera generalization of RSIM, McRSIM, which is substantially more robust to affine transformations and time delays and can be combined with the MDD feature, leading to an affinity matrix that incorporates multi-camera dynamics and geometric information. Finally, we propose a multi-camera motion segmentation algorithm based on spectral clustering on this combined affinity matrix. With both synthetic and real-world experiments on a new data set, specifically created to benchmark this scenario, we show that it achieves better performance than the state of the art. Further, since the computational complexity of the algorithm is dominated by that of the singular value decomposition step, it can comfortably handle large data sets.

References

[1] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295. Springer, 2010.

[2] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos. Jensen-bregman logdet divergence with application to

efficient similarity search for covariance matrices. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9):2161–2174, 2013.

[3] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 1071–1076. IEEE, 1995.

[4] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[5] T. Gaspar, P. Oliveira, and P. Favaro. Synchronization of two independently moving cameras without feature correspondences. In *European Conference on Computer Vision*, pages 189–204. Springer, 2014.

[6] P. Ji, M. Salzmann, and H. Li. Efficient dense subspace clustering. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 461–468. IEEE, 2014.

[7] P. Ji, M. Salzmann, and H. Li. Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4687–4695, 2015.

[8] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[9] T. Lai, H. Wang, Y. Yan, T.-J. Chin, and W.-L. Zhao. Motion segmentation via a sparsity constraint. *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[10] F. Lauer and C. Schnörr. Spectral clustering of linear subspaces for motion segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 678–685. IEEE, 2009.

[11] G. Liu. Lrr software. `https://sites.google.com/site/guangcanliu/lrr%28motion_face%29.rar?attredirects=0&d=1`, 2015. [Online; accessed 19-July-2015].

[12] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[13] R. Lublinerman, M. Sznaier, and O. Camps. Dynamics based robust motion segmentation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1176–1184. IEEE, 2006.

[14] P. Ochs and T. Brox. Higher order motion models and spectral clustering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 614–621. IEEE, 2012.

[15] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2010.

[16] R. S. Sanchez-Pena and M. Sznaier. *Robust systems theory and applications*. John Wiley & Sons, Inc., 1998.

[17] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[18] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using powerfactorization and gpca. *International Journal of Computer Vision*, 79(1):85–105, 2008.

[19] L. Zelnik-Manor and M. Irani. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–287. IEEE, 2003.

[20] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4498–4507, 2016.