Motion Segmentation by SCC on the Hopkins 155 Database [*]

Guangliang Chen, Gilad Lerman
School of Mathematics, University of Minnesota
127 Vincent Hall, 206 Church Street SE, Minneapolis, MN 55455
glchen@math.umn.edu, lerman@umn.edu

Abstract

We apply the Spectral Curvature Clustering (SCC) algorithm to a benchmark database of 155 motion sequences, and show that it outperforms all other state-of-the-art methods. The average misclassification rate by SCC is 1.41% for sequences having two motions and 4.85% for three motions.

Supp. webpage: http://www.math.umn.edu/∼lerman/scc/

1. Introduction

Multiframe motion segmentation is a very important yet challenging problem in computer vision. Given multiple image frames of a dynamic scene taken by a (possibly moving) camera, the task is to segment the point correspondences in those views into different motions undertaken by the moving objects. A more formal definition of the problem appears below.

Problem 1. *Consider a dynamic scene consisting of K rigid-body motions undertaken by K objects relative to a moving camera. Suppose that F frames of images have been taken by the camera, and that N feature points $\mathbf{y}_1, \ldots, \mathbf{y}_N \in \mathbb{R}^3$ are detected on the objects. Let $\mathbf{z}_{ij} \in \mathbb{R}^2$ be the coordinates of the feature point $\mathbf{y}_j$ in the i^{th} image frame for every $1 \leq i \leq F$ and $1 \leq j \leq N$, and form N trajectory vectors: $\mathbf{z}_j = [\mathbf{z}'_{1j}\ \mathbf{z}'_{2j}\ \ldots\ \mathbf{z}'_{Fj}]' \in \mathbb{R}^{2F}$. The task is to separate these trajectories $\mathbf{z}_1, \ldots, \mathbf{z}_N$ into independent motions undertaken by those objects.*

There has been significant research on this subject over the past few years (see [19, 16] for a comprehensive literature review). According to the assumption on the camera model, those algorithms can be divided into the following two categories:

1. *Affine methods* [5, 15, 14, 17, 20, 9, 19, 3] assume an affine projection model, so that the trajectories associated with each motion live in an affine subspace of dimension at most three (or a linear subspace of dimension at most four containing the affine subspace). Thus, the motion segmentation problem is equivalent to a subspace clustering problem. State-of-the-art affine algorithms that have been applied to this problem include Random Sample Consensus (RANSAC) [5, 15], Multi-Stage Learning (MSL) [14], Generalized Principal Component Analysis (GPCA) [17, 9, 19], Local Subspace Affinity (LSA) [20], and Agglomerative Lossy Compression (ALC) [8, 11].

2. *Perspective methods* [7, 13, 18, 6, 12, 1] assume a perspective projection model under which point trajectories associated with each moving object lie on a multilinear variety. However, clustering multilinear varieties is a challenging task and very limited research has been done in this direction.

An extensive benchmark for comparing the performance of these algorithms is the Hopkins 155 Database [16]. It contains 155 video sequences along with features extracted and tracked in all frames for each sequence, 120 of which have two motions and the rest (35 sequences) consist of three motions.

In this paper we examine the performance of a recent affine method, Spectral Curvature Clustering (SCC) [3, 2], on the Hopkins 155 database and compare it with other affine algorithms that are mentioned above (their results have been reported in [16, 11] and also partly online at http://www.vision.jhu.edu/data/hopkins155/).

Our experiments show that SCC outperforms all the above-mentioned affine algorithms on this benchmark dataset with an average classification error of 1.41% for two motions and 4.85% for three motions. In contrast, the smallest average misclassification rate among all other affine methods is 2.40% for sequences containing two motions and 6.26% for sequences with three motions, both achieved by ALC [11].

The rest of the paper is organized as follows. We first

[*] This work was supported by NSF grants #0612608 and #0915064.

briefly review the SCC algorithm in Section 2, and then test in Section 3 the SCC algorithm against other common affine methods on the Hopkins 155 database. Finally, Section 4 concludes with a brief discussion.

2. Review of the SCC algorithm

The SCC algorithm [3, Algorithm 2] takes as input a data set $\mathrm{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, which is sampled from a mixture of affine subspaces in the Euclidean space $\mathbb{R}^D$ and possibly corrupted with noise and outliers. The number of the subspaces K and the maximum[1] of their dimensions d should also be provided by the user. The output of the algorithm is a partition of the data into K (disjoint) clusters, $\mathrm{X} = \bigcup_{1 \leq k \leq K} \mathrm{C}_k$, representing the affine subspaces.

The initial step of the SCC algorithm is to randomly select from the data c subsets of (distinct) points with a fixed size $d+1$. Based on these c $(d+1)$-tuples, an affinity matrix $\mathbf{A}_c \in \mathbb{R}^{N \times c}$ is formed in the following way. Let $\mathrm{J}_1, \ldots, \mathrm{J}_c$ be the index sets of the c subsets. Then for each $1 \leq r \leq c$ and $1 \leq i \leq N$, if $i \in \mathrm{J}_r$, we set $\mathbf{A}_c(i, r) = 0$ by default; otherwise, we form the corresponding union $\mathrm{I} := [i\ \mathrm{J}_r]$ and define

$$\mathbf{A}_c(i, r) := e^{-c_{\mathrm{p}}^2(\mathrm{I})/(2\sigma^2)}, \tag{1}$$

in which $\sigma > 0$ is a fixed constant whose automatic choice is explained later, and $c_{\mathrm{p}}^2(\mathrm{I})$ is the (squared) polar curvature [3] of the corresponding $d+2$ points, $\mathbf{x}_{\mathrm{I}} := [\cdots \mathbf{x}_i \cdots]_{i \in \mathrm{I}}$. That is,

$$\begin{aligned} c_{\mathrm{p}}^2(\mathrm{I}) := & \max_{j,k \in \mathrm{I}} \|\mathbf{x}_j - \mathbf{x}_k\|_2^2 \\ & \cdot \frac{1}{d+2} \sum_{j \in \mathrm{I}} \frac{\det(\mathbf{x}_{\mathrm{I}}' \cdot \mathbf{x}_{\mathrm{I}} + 1)}{\prod_{k \in \mathrm{I}, k \neq j} \|\mathbf{x}_j - \mathbf{x}_k\|_2^2}. \end{aligned} \tag{2}$$

Note that the numerator $\det(\mathbf{x}_{\mathrm{I}}' \cdot \mathbf{x}_{\mathrm{I}} + 1)$ is, up to a factor, the (squared) volume of the $(d+1)$-simplex formed by the $d+2$ points $\mathbf{x}_{\mathrm{I}}$. Therefore, the polar curvature can be thought of as being the volume of the simplex, normalized at each vertex, averaged over the vertices, and then scaled by the diameter of the simplex. When $d+2$ points are sampled from the same subspace, we expect the polar curvature to be close to zero and consequently the affinity close to one. On the other hand, when they are sampled from mixed subspaces, the polar curvature is expected to be large and the affinity close to zero.

The SCC algorithm next forms pairwise weights $\mathbf{W}$ from the above multi-way affinities:

$$\mathbf{W} = \mathbf{A}_c \cdot \mathbf{A}_c', \tag{3}$$

and applies spectral clustering [10] to find K clusters $\mathrm{C}_1, \ldots, \mathrm{C}_K$.

In order to refine the clusters, SCC then re-samples c/K $(d+1)$-tuples from each of the clusters $\mathrm{C}_k, 1 \leq k \leq K$, and re-applies the rest of the steps. This procedure is repeated until convergence for a best segmentation, and is referred to as *iterative sampling* (see [3, Sect. 3.1.1]). Its convergence is measured by the total *orthogonal least squares (OLS)* error of d-dimensional affine subspace approximations $F_1, \ldots, F_K$ to the clusters $\mathrm{C}_1, \ldots, \mathrm{C}_K$:

$$e_{\mathrm{OLS}}^2 = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathrm{C}_k} \mathrm{dist}^2(\mathbf{x}, F_k). \tag{4}$$

In situations where the ground truth labels of the data points are known, we also compute the misclassification rate:

$$e_{\%} = \frac{\text{\# of misclassified points}}{N} \cdot 100\%. \tag{5}$$

The parameter σ of Eq. (1) is automatically selected by SCC at each iteration in the following way. Let $\mathbf{c}$ denote the vector of all the $(N-d-1) \cdot c$ squared polar curvatures computed in an arbitrarily fixed iteration. The algorithm applies the following set of candidate values which represent several scales of the curvatures:

$$\{\mathbf{c}\left((N-d-1) \cdot c/K^q\right) \mid q = 1, \ldots, d+1\}, \tag{6}$$

and chooses the one for which the error of Eq. (4) is minimized. A quantitative derivation of the above selection criterion for σ appears in [3, Section 3.1.2]. It is also demonstrated in [3] that SCC will often fail with arbitrary choices of σ.

We present (a simplified version of) the SCC algorithm below (in Algorithm 1). We note that the storage requirement of the algorithm is $O(N \cdot (D+c))$, and the total running time is $O(n_{\mathrm{s}} \cdot (d+1)^2 \cdot D \cdot N \cdot c)$, where n_{s} is the number of sampling iterations performed (till convergence, typically $O(d)$).

3. Results

We compare the SCC algorithm with other state-of-the-art affine methods, such as ALC [8, 11], GPCA [17, 9, 19], LSA [20], MSL [14], and RANSAC [5, 15], using the Hopkins 155 benchmark [16]. We also compare the performance of affine methods with an oracle, the Reference algorithm (REF) [16], which fits subspaces using the ground truth clusters and re-assigns points to its nearest subspace. Though it cannot be used in practice, REF verifies the validity of affine camera model and provides a basis for comparison among practical algorithms. The results of the latter six methods (including REF) are already published in [19, 11], so we simply copy them from there.

[1] By using only the maximal dimension we treat all the subspaces to be d-dimensional. This strategy works quite well in many cases, as demonstrated in [3].

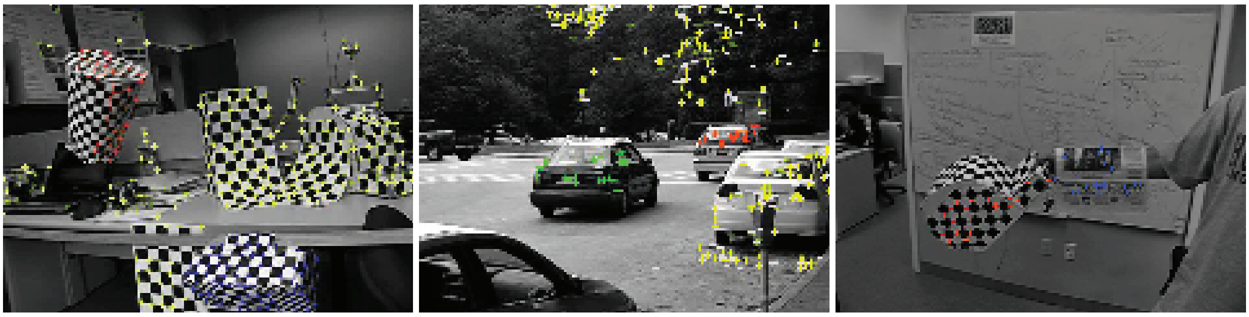


Figure 1. A sample image from each of the three categories in the Hopkins155 database.

Algorithm 1 Spectral Curvature Clustering (SCC)

```
Input: Data set X, maximal intrinsic dimension $d$, and
    number of subspaces $K$ (required); number of sampled
    subsets $c$ (default = $100 \cdot K$)
Output: $K$ disjoint clusters $\mathrm{C}_1, \dots, \mathrm{C}_K$.
Steps:
1: Sample randomly $c$ subsets of X (with indices
   $\mathrm{J}_1, \dots, \mathrm{J}_c$), each containing $d+1$ distinct points.
2: For each sampled subset $\mathrm{J}_r$, compute the squared polar
   curvature of it and each of the remaining $N-d-1$
   points in X by Eq. (2). Sort increasingly these $(N-d-1) \cdot c$
   squared curvatures into a vector $\mathbf{c}$.
3: for $q = 1$ to $d+1$ do
       • Form the matrix $\mathbf{A}_c \in \mathbb{R}^{N \times c}$ by setting $\sigma^2 =
         \mathbf{c}((N-d-1) \cdot c/K^q)$ in Eq. (1), and estimate
         the weights $\mathbf{W}$ via Eq. (3)
       • Apply spectral clustering [10] to these weights and
         find a partition of the data X into $K$ clusters
   end for
   Record the partition $\mathrm{C}_1, \dots, \mathrm{C}_K$ that has the smallest
   total OLS error, i.e., $e^2_{\mathrm{OLS}}$ of Eq. (4), for the corresponding
   $K$ $d$-dimensional affine subspaces.
4: Sample $c/K$ subsets of points (of size $d+1$) from each
   $\mathrm{C}_k$ found above and repeat Steps 2 and 3 to find $K$
   newer clusters. Iterate until convergence to obtain a
   best segmentation.
```

The Hopkins 155 database contains sequences with two and three motions, and consists of three categories of motions (see Figure 1 for a sample image in each category and Table 1 for some summary information of each category, e.g., number of sequences, average number of tracked features, and average number of frames):

- **Checkerboard**: this category consists of 104 sequences of indoor scenes taken with a handheld camera under controlled conditions.

Table 1. Summary information of the Hopkins 155 database: number of sequences (# Seq.), average number of feature points (N), and average number of frames (F) in each category for two motions and three motions separately.

	2 motions			3 motions		
	# Seq.	N	F	# Seq.	N	F
Checker.	78	291	28	26	437	28
Traffic	31	241	30	7	332	31
Other	11	155	40	2	122	31
All	120	266	30	35	398	29

- **Traffic**: this category consists of 38 sequences of outdoor traffic scenes taken by a moving handheld camera.

- **Other (Articulated/Non-rigid)**: this category contains 13 sequences displaying motions constrained by joints, head and face motions, people walking, etc.

It is proved (e.g., in [9]) that the trajectory vectors associated with each motion live in a distinct affine subspace of dimension $d \leq 3$ (or a linear subspace of dimension $d \leq 4$ containing the affine subspace). Also, it is possible to cluster the trajectories either in the full space $\mathbb{R}^{2F}$ (F is the number of frames) or in some projected space (after dimensionality reduction by PCA), e.g., $\mathbb{R}^{4K}$ (K is the number of motions) or $\mathbb{R}^{d+1}$. Thus, we will apply the SCC algorithm (Algorithm 1) to each of the 155 motion sequences to segment d-dimensional subspaces in $\mathbb{R}^D$ in six ways: $(d, D) = (3,4), (3,4K), (3,2F), (4,5), (4,4K), (4,2F)$. Each case is correspondingly represented by the shorthand *SCC* (d, D).

We use the default value $c = 100 \cdot K$ for all SCC (d, D) when applied to the 155 sequences. Also, in order to mitigate the randomness effect due to initial sampling, we repeat the experiment 100 times and record only the average misclassification rate. For each SCC (d, D), we report in Table 2 the mean and median of the averaged errors for sequences with two motions, and in Table 3 results on three motions. Figure 2 shows histograms of the misclassifica-

Table 2. Misclassification rates for sequences with *two* motions. ALC 5 and ALC sp respectively represent ALC with projection dimensions 5 and a sparsity-preserving dimension, LSA n means applying LSA in the projected space $\mathbb{R}^n$ (after dimensionality reduction), and REF refers to the reference algorithm.

	Checkerboard		Traffic		Other		All	
	mean	median	mean	median	mean	median	mean	median
ALC 5	2.66%	0.00%	2.58%	0.25%	6.90%	0.88%	3.03%	0.00%
ALC sp	1.55%	0.29%	1.59%	1.17%	10.70%	0.95%	2.40%	0.43%
GPCA	6.09%	1.03%	1.41%	0.00%	2.88%	0.00%	4.59%	0.38%
LSA 5	8.84%	3.43%	2.15%	1.00%	4.66%	1.28%	6.73%	1.99%
LSA $4K$	2.57%	0.27%	5.43%	1.48%	4.10%	1.22%	3.45%	0.59%
MSL	4.46%	0.00%	2.23%	0.00%	7.23%	0.00%	4.14%	0.00%
RANSAC	6.52%	1.75%	2.55%	0.21%	7.25%	2.64%	5.56%	1.18%
REF	2.76%	0.49%	0.30%	0.00%	1.71%	0.00%	2.03%	0.00%
SCC $(3,4)$	2.99%	0.39%	1.20%	0.32%	7.71%	3.67%	2.96%	0.42%
SCC $(3,4K)$	1.76%	0.01%	0.46%	0.16%	4.06%	1.69%	1.63%	0.06%
SCC $(3,2F)$	1.77%	0.00%	0.63%	0.14%	4.02%	2.13%	1.68%	0.07%
SCC $(4,5)$	2.31%	0.25%	0.71%	0.26%	5.05%	1.08%	2.15%	0.27%
SCC $(4,4K)$	1.30%	0.04%	1.07%	0.44%	3.68%	0.67%	1.46%	0.16%
SCC $(4,2F)$	1.31%	0.06%	1.02%	0.26%	3.21%	0.76%	1.41%	0.10%

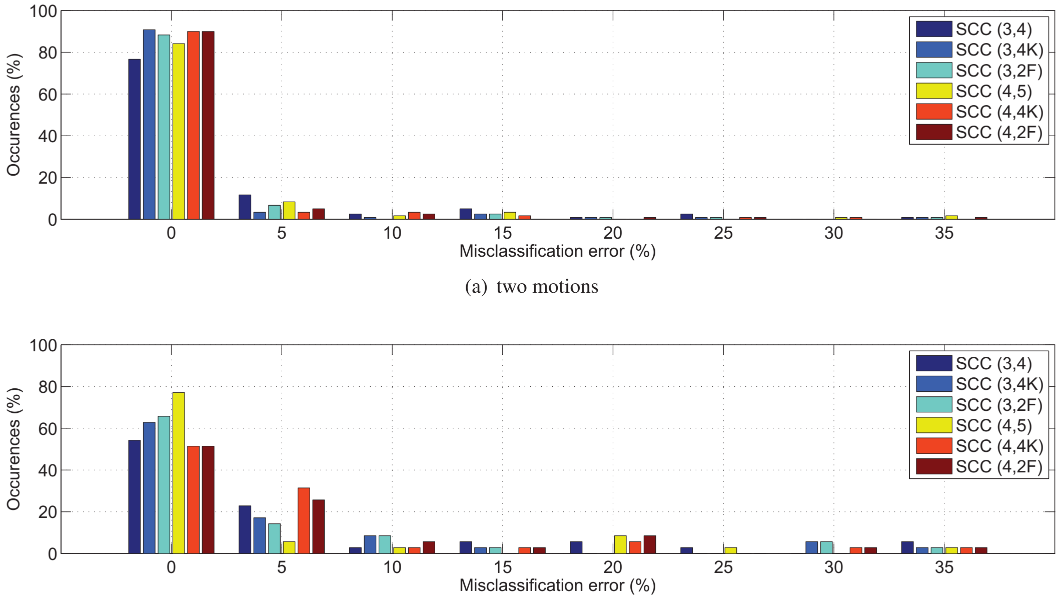
Table 3. Misclassification rates for sequences with *three* motions. ALC 5 and ALC sp respectively represent ALC with projection dimensions 5 and a sparsity-preserving dimension, LSA n means applying LSA in the projected space $\mathbb{R}^n$ (after dimensionality reduction) and REF refers to the reference algorithm.

	Checkerboard		Traffic		Other		All	
	mean	med.	mean	med.	mean	med.	mean	med.
ALC 5	7.05%	1.02%	3.52%	1.15%	7.25%	7.25%	6.26%	1.02%
ALC sp	5.20%	0.67%	7.75%	0.49%	21.08%	21.08%	6.69%	0.67%
GPCA	31.95%	32.93%	19.83%	19.55%	16.85%	16.85%	28.66%	28.26%
LSA 5	30.37%	31.98%	27.02%	34.01%	23.11%	23.11%	29.28%	31.63%
LSA $4K$	5.80%	1.77%	25.07%	23.79%	7.25%	7.25%	9.73%	2.33%
MSL	10.38%	4.61%	1.80%	0.00%	2.71%	2.71%	8.23%	1.76%
RANSAC	25.78%	26.01%	12.83%	11.45%	21.38%	21.38%	22.94%	22.03%
REF	6.28%	5.06%	1.30%	0.00%	2.66%	2.66%	5.08%	2.40%
SCC $(3,4)$	7.72%	3.21%	0.52%	0.28%	8.90%	8.90%	6.34%	2.36%
SCC $(3,4K)$	6.00%	2.22%	1.78%	0.42%	5.65%	5.65%	5.14%	1.67%
SCC $(3,2F)$	6.23%	1.70%	1.11%	1.40%	5.41%	5.41%	5.16%	1.58%
SCC $(4,5)$	5.56%	2.03%	1.01%	0.47%	8.97%	8.97%	4.85%	2.01%
SCC $(4,4K)$	5.68%	2.96%	2.35%	2.07%	10.94%	10.94%	5.31%	2.40%
SCC $(4,2F)$	6.31%	1.97%	3.31%	3.31%	9.58%	9.58%	5.90%	1.99%

tion rates with the percentage of sequences in which each algorithm achieved a certain error. The corresponding histograms for other methods are shown in [19, Figure 3].

4. Discussion

Looking at Tables 2 and 3, we conclude that the SCC algorithm (with all six pairs (d, D)) outperforms all competing methods (in terms of the mean error) and is very close to the reference algorithm (REF). In the checkerboard category, it even has a better performance than REF. In addition, SCC has the following two strengths in comparison with most other affine methods. First, as we observed in experiments, the performance of SCC is not so sensitive to its free parameter c. In contrast, the ALC algorithm is very sensitive to its distortion parameter ε and often gives incorrect number of clusters, requiring running it for many choices of ε while having no theoretical guarantee. Second, SCC can be directly applied to the original trajectory vectors (which are very high dimensional), thus preprocessing of the trajectories, i.e., dimensionality reduction, is not necessary (unlike GPCA and LSA). Finally, we remark that SCC also outperforms some perspective methods, e.g., Local Linear Manifold Clustering



(a) two motions

(b) three motions

Figure 2. Histograms of misclassification errors obtained by SCC.

(LLMC) [6] (their misclassification rates are also available at http://www.vision.jhu.edu/data/hopkins155/).

The histograms (in Figure 2) show that the SCC algorithm obtains a perfect segmentation for 80% of two-motion sequences and for over 50% of three-motion sequences. Under this criterion, SCC is at least comparable to the best algorithms (ALC, LSA $4K$, MSL) and the reference algorithm (REF); see [11, Figure 4] and [19, Figure 3]. Moveover, SCC has the shortest tails; its worst case segmentation error (about 35%) is much smaller than those of other methods some of which are as large as 50%.

Regarding running time, the SCC algorithm generally takes 1 to 2 seconds to process one sequence on a compute server with two dual core AMD Opteron 64-bit 280 processors (2.4 GHz) and 8 GB of RAM. It is much faster than the best competitors such as ALC, LSA $4K$, and MSL (see their computation time in [11, Table 6] and [16, Tables 3 & 5] while also noting that there were all performed on faster machines).

At the time of finalizing this version we have found out about the very recent affine method of Sparse Subspace Clustering (SSC) [4] which reportedly has superb results on the Hopkins 155 database and outperforms the results reported here for both SCC and REF. It will be interesting to test its sensitivity to its tuning parameter λ in future work.

Acknowledgements

We thank the anonymous reviewers for their helpful comments and for pointing out reference [4] to us. Special thanks go to Rene Vidal for encouraging comments when this manuscript was still at an early stage and for referring us to the workshop. Thanks to the Institute for Mathematics and its Applications (IMA), in particular Doug Arnold and Fadil Santosa, for an effective hot-topics workshop on multi-manifold modeling that we participated in. The research described in this paper was partially supported by NSF grants #0612608 and #0915064.

References

[1] G. Chen, S. Atev, and G. Lerman. Kernelized spectral curvature clustering (KSCC). In *ICCV Workshop on Dynamical Vision*, 2009.

[2] G. Chen and G. Lerman. Foundations of a multi-way spectral clustering framework for hybrid linear modeling. *Found. Comput. Math.*, 2009. DOI 10.1007/s10208-009-9043-7.

[3] G. Chen and G. Lerman. Spectral curvature clustering (SCC). *Int. J. Comput. Vision*, 81(3):317–330, 2009.

[4] R. V. E. Elhamifar. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[5] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, June 1981.

[6] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *IEEE*

Conference on Computer Vision and Pattern Recognition, pages 1–6, June 2007.

[7] R. Hartley and R. Vidal. The multibody trifocal tensor: motion segmentation from 3 perspective views. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 769–775, 2004.

[8] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1546–1562, September 2007.

[9] Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50(3):413–458, 2008.

[10] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856, 2001.

[11] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[12] S. Rao, A. Yang, S. Sastry, and Y. Ma. Robust algebraic segmentation of mixed rigid-body and planar motions. Submitted to International Journal of Computer Vision, 2008.

[13] K. Schindler, J. U, and H. Wang. Perspective n-view multibody structure-and-motion through model selection. In *Proc. of the 9th European conference on computer vision*, volume 1, pages 606–619, 2006.

[14] Y. Sugaya and K. Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. In *Lecture Notes in Computer Science*, volume 3247/2004, chapter Statistical Methods in Video Processing, pages 13–25. Springer Berlin / Heidelberg, 2004.

[15] P. H. S. Torr. Geometric motion segmentation and model selection. *Phil. Trans. R. Soc. Lond. A*, 356:1321–1340, 1998.

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

[17] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12), 2005.

[18] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *Int. J. Comput. Vis.*, 68(1):7–25, June 2006.

[19] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using powerfactorization and GPCA. *Int. J. Comput. Vis.*, 79:85–105, 2008.

[20] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and nondegenerate. In *ECCV*, volume 4, pages 94–106, 2006.