Motion Segmentation by Velocity Clustering with Estimation of Subspace Dimension

Liangjing Ding*, Adrian Barbu**, Anke Meyer-Baese*

*Department of Scientific Computing, Florida State University
**Department of Statistics, Florida State University

Abstract. The performance of clustering based motion segmentation methods depends on the dimension of the subspace where the point trajectories are projected. This paper presents a strategy for estimating the best subspace dimension using a novel clustering error measure. For each obtained segmentation, the proposed measure estimates the average least square error between the point trajectories and synthetic trajectories generated based on the motion models from the segmentation. The second contribution of this paper is the use of the velocity vector instead of the traditional trajectory vector for segmentation. The evaluation on the Hopkins 155 video benchmark database shows that the proposed method is competitive with current state-of-the-art methods both in terms of overall performance and computational speed.

1 Introduction

The task of motion segmentation is to label a set of tracked feature points from several moving objects into different groups based on their motions. This is an important step in many computer vision problems, such as robotics, inspection, video surveillance, etc. Motion segmentation has been studied mostly in the case of the affine camera model, under which the vectors of feature points from each rigid motion lie in a subspace of dimension four or less [1], thus the motion segmentation problem can be posed as a subspace separation problem. The main difficulty in subspace separation is that it is usually hard to determine the number of subspaces and their dimension. For example, tracked feature points from a static background might lie on a 2-dimensional subspace, while points from other motions might lie on subspaces of dimension 3 or 4. Moreover, practical motion scenes usually exhibit partially dependent motions, such as when two objects have the same rotational but different translational motion relative to the camera [2], or for articulated motions [3].

Many methods [4], [5], [6], [7], [8] project the feature trajectories onto a smaller dimensional space and perform clustering on the projected points. This approach not only provides computational advantages, but also imposes some sort of a spatial prior on the point trajectories.

Unlike earlier attempts to find a best projection dimension for subspace separation, this paper proposes to perform subspace separation for all possible dimensions. Based on this idea, this paper proposes a motion segmentation approach which performs spectral clustering in many dimensions, and then carefully selects the result with the best separability using a novel clustering error measure.

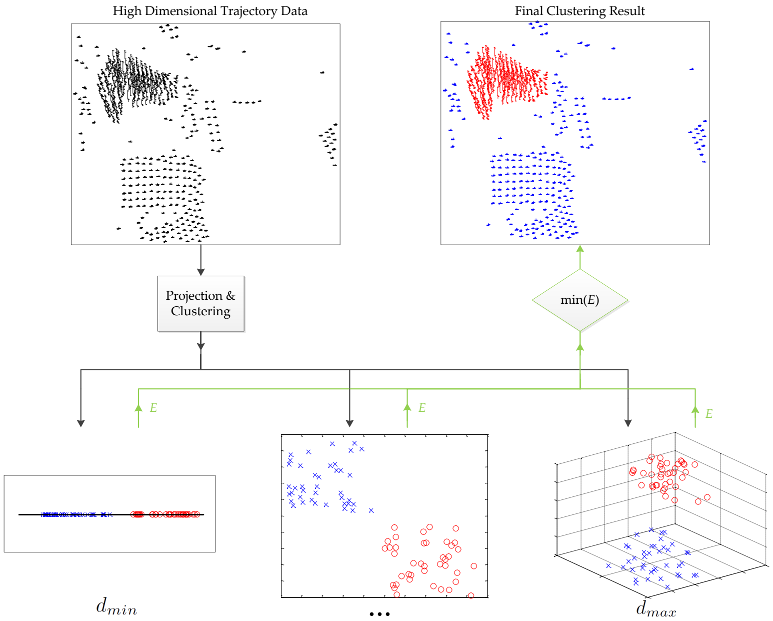


Fig. 1: Illustration of the process of selection of the best result after performing spectral clustering in spaces of dimensions in the range $[d_{min}, d_{max}]$. The selection is based on a clustering error measure described in Section 3.4.

Related Work. Early works of multiframe 3-D motion segmentation based on matrix factorization [9], [10] find the segmentation by thresholding the entries of a similarity matrix built from the factorization of the matrix of data points. However, the thresholding process is very sensitive to noise and such methods are only provably correct when the subspaces are independent. The Generalized Principal Component Analysis (GPCA) [7] is an algebraic method for subspace separation which could deal with dependent motions, but it is not robust to data contaminated by outliers and noise. Some statistical methods, such as Agglomerative Lossy Compression (ALC) [6], RANSAC [11], Multi-Stage Learning (MSL) [2], etc, can handle noise in the data, but their assumptions about the distribution of the noise are not optimal. In recent years, spectral clustering has become a widely used method in motion segmentation. Based on the fact that a point and its k-nearest neighbors (k-NNs) often belong to the same subspace, Local Subspace Affinity (LSA) [8], Spectral Local Best-fit Flats (SLBF) [12], Locally Linear Manifold Clustering (LLMC) [13] use the angle or distance between a point and the subspace fitted through the point and its k-NNs to construct the affinity measure for spectral clustering. However, the neighbors of a point could belong to different spaces, especially when close to the intersection of two subspaces. Also, the selected neighbors may not span the underlying subspace. The spectral clustering (SC) method [5], which uses the angular information between trajectories as affinity, is simple and efficient, but its criterion to select the best subspace dimension is noise-sensitive. More recently, some approaches such as Spectral Curvature Clustering (SCC) [14], Sparse Subspace Clustering (SSC) [4], and Low-Rank Representation (LRR) [15], use the so-called *sparsity* information as the affinity measure. Optimization is always involved in these methods, which makes them computationally expensive.

Our Contributions. In this work, we provide two main contributions. First, we use the velocity vector as a preprocessing step to reduce the influence of the errors accumulated during feature point tracking. This step proves to be very important for improving performance. Second, we present a method for estimating the optimal projection dimension for spectral clustering. We use the angular information between the points proposed in SC [5] to build the affinity matrix. Compared to the SC algorithm, the proposed method presents a different strategy for selecting the best subspace dimension. The SC finds the best dimension before performing the spectral clustering, and the dimension is determined by the so called *relative gap* which is related to the eigenvalues of a Lapacian matrix L . However, when the noise level is large, the relative gap is not very effective. Instead, our method performs spectral clustering after projecting to each of the possible dimensions in a range $[d_{min}, d_{max}]$, and then selects the best result based on a novel clustering error measure. The advantage of the proposed strategy is that the performance is much more robust to data corrupted by noise. Moreover, the complexity of the resulting algorithm remains low as long as the number of motions is small. When applied to the motion segmentation data from the Hopkins155 database [16], the proposed method is competitive with the current state-of-the-art methods both in terms of segmentation accuracy and computational speed.

2 Mathematical Background

Recent works on motion segmentation [5], [14], [4], [15] usually considered the affine camera model. The affine camera model assumes an affine projection model, which generalizes orthographic, weak-perspective and paraperspective projection. Under the affine camera model, a point on the image plane (x, y) is related to the real world point (X, Y, Z) by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \underbrace{K \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \end{bmatrix} \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}}_{A \in \mathbb{R}^{2\times 4}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{1}$$

where A is the affine motion matrix, which is determined by the camera calibration matrix $K \in \mathbb{R}^{2\times 3}$ and the relative orientation of the image plane with respect to the world coordinates $(R, t) \in SE(3)$.

Let $t = (x^1, y^1, x^2, y^2, \ldots, x^F, y^F)^T$ be a trajectory of a tracked feature point in F frames. Given P trajectories undergoing the same rigid motion, the *measurement matrix* $W = [t_1, t_2, \ldots, t_P]$ is constructed. From equation (1), W can be decomposed into a *motion matrix* $M \in \mathbb{R}^{2F\times 4}$ and a *structure matrix* $S \in \mathbb{R}^{4\times P}$ as

$$W = MS$$
$$\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_P^1 \\ y_1^1 & y_2^1 & \cdots & y_P^1 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^F & x_2^F & \cdots & x_P^F \\ y_1^F & y_2^F & \cdots & y_P^F \end{bmatrix} = \begin{bmatrix} A^1 \\ \vdots \\ A^F \end{bmatrix} \begin{bmatrix} X_1 & \cdots & X_P \\ Y_1 & \cdots & Y_P \\ Z_1 & \cdots & Z_P \\ 1 & \cdots & 1 \end{bmatrix}.$$

where A^f is the affine motion matrix at frame f. It implies that $\text{rank}(W) \leq 4$. In other words, under the affine camera model, the 2-D trajectories of a set of 3-D points from a rigidly moving object reside in a subspace of dimension at most 4. Also, it is worth noting that the rows of each A^f involve linear combinations of the first two rows of the rotation matrix R^f, hence $\text{rank}(W) \geq \text{rank}(A^f) = 2$.

Additionally, the entries of the last row of the structure matrix S are identically 1. It is easy to derive the orthographic camera model [1]. Define the registered trajectories as

$$\tilde{t}_i = t_i - \frac{\sum_{i=1}^{P} t_i}{P},$$

then the registered measurement matrix

$$\tilde{W} = [\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_P] \tag{2}$$

is at most rank 3. This means that the trajectories are in a 3-D affine subspace within the 4-D space.

3 Motion Segmentation by Spectral Clustering

This paper only focuses on the problem of segmentation of tracked feature point trajectories. The goal is to find labels for all trajectories, to group them according to their corresponding motions. Also, we assume that the number of different motions is already known.

3.1 Noise Reduction using Velocity Vectors

Methods for reducing the noise level in the trajectory data is an area that did not receive enough attention in previous work. Noise is an inevitable by-product of feature tracking. Tracking errors are introduced with each new frame, due to factors such as aliasing, non-constant brightness, lack of texture, occlusion, and so on. These errors tend to accumulate and the total tracking error tends to grow as the number of frames increases.

In order to reduce the effect of the accumulated error in the motion segmentation, we use the velocity vector to characterize the trajectories, which is defined by

$$[x^1 - x^2, y^1 - y^2, \dots x^{F-1} - x^F, y^{F-1} - y^F, x^i, y^i]^T, i \in [1, \cdots, F] \tag{3}$$

With the exception of the last two rows, the entries of the other rows are replaced with the corresponding velocities. In the last two rows, the feature locations of the i-th frame are kept. The selection of i is not crucial. In this paper, we use $i = F$ but we could as well use $i = 1$ for example. The advantage is that the velocities in each frame contain only the tracking error from the previous frame to the current frame, and not the error accumulated from the starting frame. A similar velocity has been used to measure the distance between trajectories for motion segmentation in [17].

It is easy to see that when the measurement matrix W' is built from the velocity vectors, no information is lost since the original measurement matrix W can be recovered from W' by simple row operations. Because of this, the ranks of W and W' are the same. In other words, the subspace clustering problem has not been changed. However, even though the velocity matrix differs from the original measurement matrix only by row operations, the subspace projections are different because these row operations cannot be represented by a rotation matrix.

Table 1: The SSE and variance of the distances from the projected points to the fitted subspaces in 3D for a synthetic experiment. The projected points were generated from trajectories with different signal-to-noise ratio (SNR).

	SSE	Variance
Distance Vector (No noise added)	0	0
Velocity Vector (No noise added)	0	0
Distance Vector (SNR = 10)	0.256e-5	0.0011e-5
Velocity Vector (SNR = 10)	0.106e-5	0.0004e-5
Distance Vector (SNR = 5)	1.058e-5	0.005e-5
Velocity Vector (SNR = 5)	0.208e-5	0.001e-5

The noise reduction effect of using the velocity vector can be well observed in a synthetic experiment. For this purpose, 242 synthetic trajectories of length 20 were generated for two different motions, perfectly following the affine camera model. The starting feature points were randomly chosen in the first frame, and different levels of Gaussian tracking errors were introduced based on the displacement of feature points. If denote the tracker as f, and noise as n, to a point p_i in frame i, the tracked point in the next frame would be $p_{i+1} = f(p_i) + n$.

The trajectories were projected to a 3D subspace by truncated SVD. A plane was fitted in a least squares sense to the projected points of each motion. The sum of squared error (SSE) and variance of the distances from projected points to the fitted planes are shown in table 1. One could see that by using velocity vectors the noise is reduced, and the reduction is greater when the tracking errors are larger. Since the projected points obtained by velocity clustering are closer to

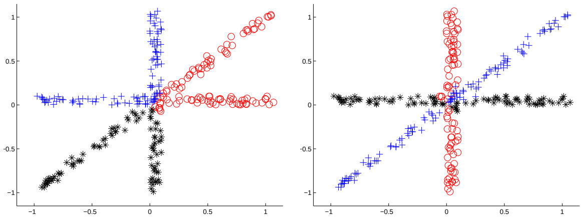


Fig. 2: Spectral clustering of lines with a distance-based affinity mixes points from different subspaces (left), while the angle-based affinity (4) separates them very well (right).

satisfying the planarity assumption, it should be expected that the segmentation results would also be better.

3.2 Spectral Clustering of Subspaces

Spectral clustering [18], [19] is a popular technique for solving motion segmentation problems [20], [21], [8], [12], [13], [4], [15], [14]. One challenge in applying spectral clustering is the construction of a good affinity matrix. Two points that lie in two different subspaces and are near the intersection of the subspaces may be close to each other. Conversely, a pair of points in the same subspace could be far from each other. As a consequence, one cannot use the typical distance-based affinity.

SC [5] proposes an affinity measure based on the angle between two vectors, defined by

$$A_{ij} = (\frac{x_i^T x_j}{\|x_i\|_2 \|x_j\|_2})^{2\alpha}, i \neq j, \alpha \in \mathbb{N} \qquad (4)$$

where x_i, x_j are two vectors. The parameter $\alpha > 1$ is used to increase the separation and should be tuned according to the noise level. It has been proved [5] that the proposed affinity measure (4) guarantees that each point x_i has a higher connection with its own group than the others. Figure 2 shows the power of angle-based affinity over the distance-based affinity in clustering 1D subspaces in 2D. This paper also uses the angular information to build the affinity matrix. While SC [5] suggests to set $\alpha = 4$ for motion segmentation, in the experiments of section 5, we find that $\alpha = 2$ could produce better results for our algorithm.

3.3 Best Subspace Dimension

Most motion segmentation methods usually require the projection to a low dimensional space where the clustering is performed. The dimension of this projection space has a large impact on the speed and accuracy of the final result. GPCA [7] suggests to project trajectories onto a 5-dimensional space. However, five dimensions are not sufficient to complex scenes, such as scenes with articulated or nonrigid motions. Motivated by compressive sensing [22], ALC [6] chooses to use the sparsity-preserving dimension

$$d_{sp} = \min_{d \geq 2D \log(2F/d)} d$$

for D-dimensional subspaces (with $D = 4$ for motion segmentation). SC [5] wants the intersection of different subspaces to have minimal dimension and proposes to set dimension $d = kD+1$, where k is the number of motions and $1 \leq D \leq 4$ for motion segmentation; the d used for clustering is searched in range $[k+1, 4k+1]$ by some relative gap.

The main difficulty for selecting the best subspace dimension is that the dimension of one affine subspace is not fixed. If one tries to find the correct dimension by setting a threshold of noise, this scheme will not work well because different scenes may have different thresholds.

The search strategy in SC [5] is innovative, but the range of possible dimensions that are searched is a parameter that needs to be tuned. Moreover, the criterion to select the best dimension in SC [5] is related to the noise level, and is not optimal in some scenarios, as we will see in experiments.

In this paper, we don't look for the best subspace dimension directly. Instead, we employ an exhaustive strategy. Since the best dimension is unknown and hard to determine, our method performs clustering after projecting to spaces of all possible dimensions, then the best result is chosen by a clustering measure. Based on this idea, finding the best subspace dimension is not necessary in this paper. What we need to do is to find a bound on the possible dimensions.

The dimension of one affine subspace S is not fixed but is bounded by

$$2 \leq \dim(S) \leq 4.$$

If there are k linear affine subspaces in general position embedded in space S_k, we would expect

$$2k \leq \dim(S_k) \leq 4k.$$

This is the range of space dimensions that will be used in our method. The best dimension will be determined using the clustering error measure defined in the next section.

3.4 Motion Error Measure

When the spectral clustering is performed in the selected spaces, a number of results will be obtained. A question is raised naturally: how to select the best one? In this paper we investigate two types of estimators of the segmentation error, both based on a RMSE error measure for each trajectory.

The Tomasi-Kanade factorization [1] allows us to write the registered matrix $\tilde{W}$ in equation (2) as

$$\tilde{W} = \tilde{M}\tilde{S}$$

where $\tilde{M}$ is a $2F \times 3$ matrix and $\tilde{S}$ is a $3 \times P$ matrix. There is an inherent ambiguity in $\tilde{M}$ and $\tilde{S}$ but we will show that it is irrelevant for the error measure.

Any registered trajectory $\tilde{t}$ in $\tilde{W}$ will have a corresponding point $\tilde{P} \in \mathbb{R}^3$ obtained by least squares:

$$\tilde{P} = \operatorname*{argmin}_{\tilde{P}} \|\tilde{t} - \tilde{M}\tilde{P}\|^2.$$

We define the RMSE error of $\tilde{t}$ as

$$\mathrm{RMSE}_{\tilde{W}}(\tilde{t}) = \sqrt{\frac{\min_{\tilde{P}} \|\tilde{t} - \tilde{M}\tilde{P}\|^2}{F}} \tag{5}$$

The RMSE error is measured in pixels and can be viewed as the tracking error for one trajectory.

Remark 1 *The $RMSE_{\tilde{W}}(\tilde{t})$ is invariant to the choice of $\tilde{M}$ and $\tilde{S}$ in the decomposition $\tilde{W} = \tilde{M}\tilde{S}$.*

Proof. To any 3×3 invertible matrix A, $\tilde{W} = \tilde{M}AA^{-1}\tilde{S}$. It can be easily verified that $\mathrm{RMSE}_{\tilde{W}}(\tilde{t})$ in equation (5) does not change when $\tilde{M}$ is multiplied by an invertible matrix A. Moreover, any decomposition $\tilde{W} = \tilde{M}'\tilde{S}'$ has $\tilde{M}' = \tilde{M}A$ for some invertible matrix A. ■

Given a labeling L of the trajectories, obtain for each label l the registered measurement matrix $\tilde{W}^l$ containing all trajectories with label l. Based on $\tilde{W}^l$ we define two types of estimators of the segmentation error.

The first type is just the sum of the RMSE errors of all registered trajectories based on their corresponding motion matrices

$$E(L) = \sum_{l=1}^{k} \sum_{i, L(i)=l} \mathrm{RMSE}_{\tilde{W}^l}(\tilde{t}_i). \tag{6}$$

The second type makes the contribution of each registered trajectory comparing to a threshold τ

$$E_\tau(L) = \sum_{l=1}^{k} \sum_{i, L(i)=l} I(\mathrm{RMSE}_{\tilde{W}^l}(\tilde{t}_i) \geq \tau). \tag{7}$$

where $I(\cdot)$ is the indicator function taking on value 1 if its argument is true or 0 otherwise.

In a perfect segmentation result, each trajectory would have a small RMSE error because of the affine camera model, resulting in a small clustering error $E(L)$ and $E_\tau(L)$.

A number of segmentation results can be obtained by projecting the original trajectories to spaces of different dimensions and performing clustering in those spaces. The problem is how to select from the obtained segmentations the one with the smallest error. For that we can use an estimator that correlates well with the segmentation error.

We propose to use the measures $E(L)$ and $E_\tau(L)$ to rank the obtained segmentations. We evaluated the capability of these error measures to find the better

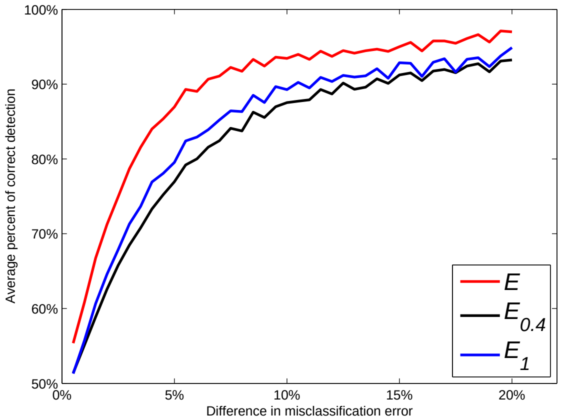


Fig. 3: The average percentage of times the proposed error estimators find the better segmentation out of two segmentations vs. their difference in misclassification errors for sequences with 3 motions in the Hopkins 155 dataset. If one segmentation is much better than the other, it will be found most of the time.

one out of two segmentations on the sequences with 3 motions in the Hopkins 155 dataset (See section 5). It is expected that when one segmentation is much better than the other (i.e. the error difference is large), the better segmentation should be found more often. Different segmentations were obtained in this way: for one sequence, a random set of $p\%$ of trajectories ($p \leq 50$ is a random number) were assigned random labels, while the labels of the remaining trajectories were untouched. 500 sets of segmentation were generated for each sequence (25000 segmentations in total for all sequences). At last, we calculated the difference in misclassification error and the average correct detection rate shown in Figure 3. One can see that if one segmentation is much better than the other, it will be found most of the time. Also, $E(L)$ always outperforms the other two estimators. Thus in this paper, $E(L)$ is adopted.

4 Complete Procedure

Dimension Reduction. Dimension reduction seems to be a standard procedure for motion segmentation by spectral clustering in [5], [6], [7]. It can improve the computational tractability without adversely affecting the quality of the segmentation, since in general the projection onto an arbitrary d-dimensional space preserves the multi-subspace structure of data lying on subspaces with dimensionality strictly less than d. There are two different strategies in dimension reduction: the random sampling [14], [4] and the truncated SVD [9], [5], [2]. This paper uses the latter method for dimension reduction from $W' \in \mathbb{R}^{2F\times P}$ to $X = [x_1, ..., x_P]^T \in \mathbb{R}^{D\times P}$ in our framework, where D is the dimension of the subspace. The truncated SVD is related to the factorization-based methods [23], [24], which use the SVD, $W = U\Sigma V^T$, to obtain a shape interaction matrix $Q = VV^T$. In order to deal with the noise and dependencies, we use the truncated SVD of the velocity measurement matrix, $W' \approx U_D \Sigma_D V_D^T$.

Details of Spectral Clustering. After the projection for dimension reduction, the spectral clustering method is applied to obtain the clustering result.

The affinity matrix is constructed using the angular affinity metric in equation (4). In fact, the affinity matrix can be easily calculated as $Q = (\tilde{V}_D \tilde{V}_D^T)^{2\alpha}$, where $\tilde{V}_D$ is the V_D with normalized rows. This normalization ensures that only the angular information is taken into account.

From the affinity matrix, the corresponding Laplacian matrix L is obtained. Then the k largest eigenvectors of L are found, where k is the number of clusters. A matrix A is formed by stacking the k eigenvectors in columns. Finally, the segmentation of the trajectories follows by applying K-means clustering to the rows of $\tilde{A}$, which is obtained by normalizing the rows of A .

Selection of the best result. According to section 3.3, to ensure that the best result is not missed, an exhaustive search strategy is employed. Let $d_{min} = 2k$ and $d_{max} = 4k$ be the minimal and maximal subspace dimensions, motion segmentation is performed in spaces with all dimensions D in the range $D \in [d_{min}, d_{max}]$. Then the best result is selected among all results based on the smallest clustering error (6) or (7). The whole procedure is illustrated in Figure 1 and described in Algorithm 1.

Algorithm 1 Velocity Clustering with Estimation of Subspace Dimension

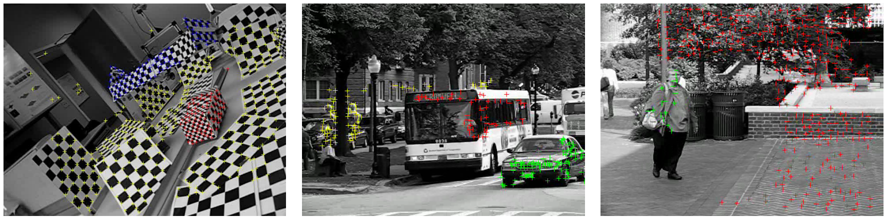
```
Input: The measurement matrix $W = [t_1, t_2, \ldots, t_P] \in \mathbb{R}^{2F \times P}$ whose columns are
point trajectories, and the number of clusters $k$.
Preprocessing: Build the velocity measurement matrix $W'$ by row transformations
of $W$ given by eq. (3).
for $D = d_{min}$ to $d_{max}$ do
  1. Perform SVD: $W' = U \Sigma V^T$
  2. Build the $N$-by-$D$ data matrix

                         $X_D = [v_1, ..., v_D]$

  where $v_i$ is the $i$-th column of $V$.
  3. Apply spectral clustering to the $N$ points in $X_D$ using the affinity measure (4).
  4. Compute the clustering error $E_D$ of the segmentation result using eq. (6).
end for
Output: The segmentation result with the smallest error $E_D$.
```

5 Experiments

The Hopkins 155 Dataset [16] has been created with the goal of providing an extensive benchmark for testing feature based motion segmentation algorithms.



(a) Checkerboard (b) Traffic (c) Articulated

Fig. 4: Sample images from some sequences of three categories in the Hopkins 155 database with ground truth superimposed.

It contains video sequences along with some feature points extracted and tracked in all the frames. The ground-truth segmentation is also provided for evaluation purposes. Based on the content of the video and the type of motion, the 155 sequences can be categorized into three main groups: *checkerboard*, *traffic* and *articulated*. Figure 4 shows sample frames from three videos of the Hopkins 155 database with the feature points superimposed. The sequences contain degenerate and non-degenerate motions, independent and partially dependent motions, nonrigid motions, etc. Since the trajectories were obtained by an automatic tracker, they could be considered as slightly corrupted by noise.

We have tested our algorithm on the image sequences from the Hopkins 155 database, as well as several other state-of-the-art algorithms: ALC [6], SC [5] and SSC [4]. For each algorithm on each sequence, we recorded the misclassification rate defined as

$$\text{Misclassification Rate} = \frac{\#\text{ of misclassified points}}{\text{total }\#\text{ of points}} \tag{8}$$

The parameter setting in our method are $\alpha = 2$, $d_{min} = 2k$, $d_{max} = 4k$, and the locations of the last frame are kept to build the velocity matrix. The results on sequences with 2, 3 motions and the whole dataset are presented in Table 2 and compared with the three state-of-the-art and baseline methods. We also show in the table the results of the algorithm with fixed subspace dimension $D = 4k$ as well as results without using the velocity preprocessing step.

One could see that by using the velocity for clustering the misclassification error decreases by about 0.8% while by using the clustering error measure to decide the best segmentation the error decreases from 4.91% to 0.99%. Thus the clustering error measure has a large impact in the spectral clustering performance while the velocity clustering has a smaller but also important impact.

Compared to other motion segmentation algorithms, our approach outperforms for the 3 motion sequences and for all the sequences combined and is outperformed on the two motion sequences by SC [5] and SSC [4]. We achieve an overall misclassification error of 1.10% for 3 motions, around half of the best reported result (SC [5]); an overall error of 0.96% for 2 motions, coming close to the best performing SSC [4]; and an overall error of 0.99% for the whole database, which is better than the other methods. Our method always obtains good results for checkerboard sequences which have the most complicated scenes (including both translation and rotation motions) in the dataset.

Table 2: Misclassification rate (in percent) for sequences of full trajectories in the Hopkins 155 dataset (Subscript $4k$ means using fixed dimension $4k$ instead of dimension search, and superscript $*$ means not using velocity for clustering).

Method	ALC	SC	SSC	Our Method$^*_{4k}$	Our Method*	Our Method$_{4k}$	Our Method
Checkerboard (2 motion)							
Average	1.55	0.85	1.12	2.07	1.38	1.38	**0.67**
Median	0.29	0.00	0.00	0.30	0.00	0.00	0.00
Traffic (2 motion)							
Average	1.59	0.90	**0.02**	6.87	1.35	8.25	0.99
Median	1.17	0.00	0.00	1.33	0.30	1.09	0.22
Articulated (2 motion)							
Average	10.70	1.71	**0.62**	6.02	2.56	2.46	2.94
Median	0.95	0.00	0.00	0.99	0.88	0.88	0.88
All (2 motion)							
Average	2.40	0.94	**0.82**	3.67	1.48	3.25	0.96
Median	0.43	0.00	0.00	0.51	0.00	0.00	0.00
Checkerboard (3 motion)							
Average	5.20	2.15	2.97	4.38	1.06	2.28	**0.74**
Median	0.67	0.47	0.27	1.37	0.58	0.51	0.21
Traffic (3 motion)							
Average	7.75	1.35	**0.58**	27.80	8.22	19.21	1.13
Median	0.49	0.19	0.00	32.27	1.42	28.28	0.21
Articulated (3 motion)							
Average	21.08	4.26	**1.42**	6.18	6.18	18.95	5.65
Median	21.08	4.26	0.00	6.18	6.18	18.95	5.65
All (3 motion)							
Average	6.69	2.11	2.45	9.17	2.78	6.62	**1.10**
Median	0.67	0.37	0.20	1.99	0.67	0.85	0.22
All sequences combined							
Average	3.37	1.20	1.24	4.91	1.78	4.01	**0.99**
Median	0.49	0.00	0.00	0.57	0.00	0.24	0.00

The performance on the articulated sequences with 3 motions is worse than the SC, possibly because these sequences don't obey the rigid motion model and thus the RMSE measure might not be accurate. On the other hand, when the motions follow the rigid model, the RMSE measure helps obtain very good results. This is clearly visible in the three motion checkerboard sequences, where our algorithm obtains errors less than half of the other algorithms.

From the cumulative distributions in Figure 5, we see that for 2 motions, our method is comparable to the best method SSC; and for 3 motions, our method outperforms all others. Moreover, the largest error of our method for 3 motions is about 10%, while that of the other methods is around 40%.

Table 3: Average computing time for sequences in the Hopkins 155 database.

	ALC	SC	SSC	Our Method
2 motions	7.85m	0.53s	2.27m	0.72s
3 motions	16.77m	1.34s	4.08m	1.81s

Table 3 shows that the average computing time per sequence (obtained on a 2.66GHz Core 2 Duo computer with Matlab on Linux) for sequences with 2 motions is less than 1 second, while that for sequences with 3 motions is less than 2 seconds. In comparison to other methods, our method is much faster than ALC and SSC, but slightly slower than SC.

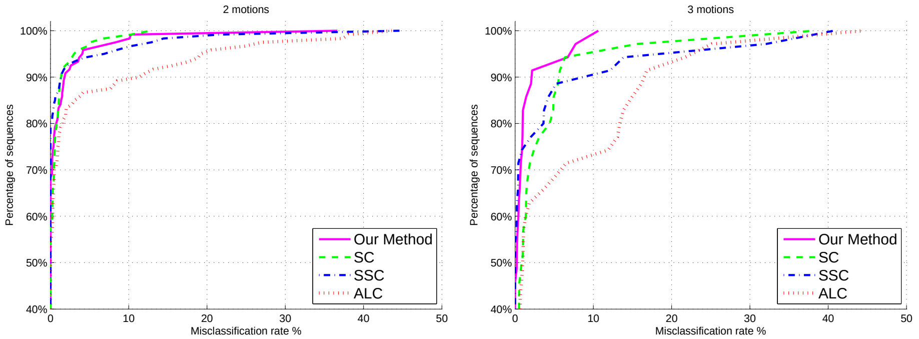


Fig. 5: The cumulative distribution of the misclassification rate for two and three motions in the Hopkins 155 database.

6 Conclusion and Future Work

In this paper, we presented a method for segmenting moving objects using spectral clustering. The method uses the velocity vectors as the input for clustering, which is more robust to accumulated errors, and then applies spectral clustering in all possible subspace dimensions. The final segmentation is selected from the obtained results using a novel clustering error measure. Our evaluation on the Hopkins 155 database shows that the method is competitive with current state-of-the-art methods, both in terms of overall performance and computational speed. The algorithm has been shown to be robust to different types of scenes and motions present in the Hopkins 155 database, while remaining very efficient in computation time.

Future work will study how to extend the method to deal with incomplete trajectories, and hopefully, treat complete and incomplete trajectories equally.

References

1. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision **9** (1992) 137–154
2. Sugaya, Y., Kanatani, K.: Geometric structure of degeneracy for multi-body motion segmentation. In: Workshop on statistical methods in video processing. (2004)
3. Yan, J., Pollefeys, M.: A factorization approach to articulated motion recovery. In: IEEE conference on computer vision and pattern recognition. Volume II. (2005) 815–821
4. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: IEEE conference on Computer Vision and Pattern Recognition. (2009)

5. Lauer, F., Schnörr, C.: Spectral clustering of linear subspaces for motion segmentation. In: IEEE International Conference on Computer Vision. (2009)
6. Rao, S., Tron, R., Vidal, R., Ma, Y.: Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 1832–1845
7. Vidal, R., Ma, Y., Sastry, S.: Generalized principal component analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 1945–1959
8. Yan, J., Pollefeys, M.: A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: European Conference on Computer Vision. (2006) 94–106
9. Costeira, J., Kanade, T.: A multibody factorization method for independently moving objects. International Journal of Computer Vision **29** (1998) 159–179
10. Gear, C.W.: Multibody grouping from motion images. International Journal of Computer Vision **29** (1998) 133–150
11. Fischler, M.A., Bolles, R.C.: RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **26** (1981) 381–395
12. Zhang, T., Szlam, A., Wang, Y., Lerman, G.: Hybrid linear modeling via local best-fit flats. In: IEEE Conference on Computer Vision and Pattern Recognition. (2010) 1927–1934
13. Goh, A., Vidal, R.: Segmenting motions of different types by unsupervised manifold clustering. In: IEEE Conference on Computer Vision and Pattern Recognition. (2007)
14. Chen, G., Lerman, G.: Spectral curvature clustering. International Journal of Computer Vision **81** (2009) 317–330
15. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: International Conference on Machine Learning. (2010)
16. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2007) 1–8
17. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. European Conference on Computer Vision (2010) 282–295
18. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000)
19. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems 14. Volume 2. (2001)
20. Wang, H., Culverhouse, P.: Robust motion segmentation by spectral clustering. In: Proc. of the British Machine Vision Conference. (2003) 639–648
21. Park, J., Zha, H., Kasturi, R.: Spectral clustering for robust motion segmentation. In: European Conference on Computer Vision. (2004) 390–401
22. Donoho, D., Tanner, J.: Counting faces of randomly-projected polytopes when the projection radically lowers dimension. American Mathematical Society **22** (2009) 1–53
23. Costeira, J., Kanade, T.: A multi-body factorization method for motion analysis. In: Proceedings of the 5th International Conference on Computer Vision. (1995) 1071–1076
24. Kanatani, K.: Motion segmentation by subspace separation and model selection. In: Proceedings of the 8th IEEE International Conference on Computer Vision. Volume 2. (2001) 586–591