Rigid Motion Segmentation using Randomized Voting

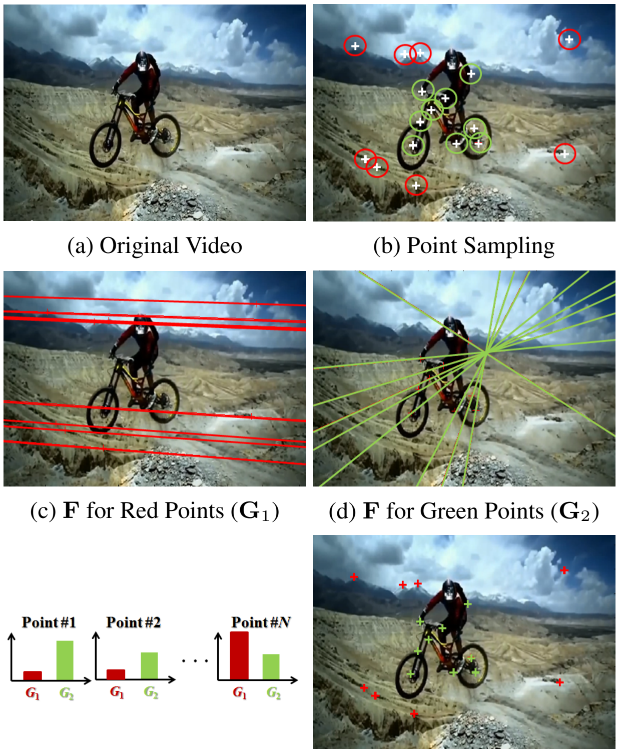
Heechul Jung[†] Jeongwoo Ju[†] Junmo Kim[‡]
Statistical Inference and Information Theory Laboratory
Korea Advanced Institute of Science and Technology
{heechul, veryju}@kaist.ac.kr[†], junmo@ee.kaist.ac.kr[‡]

Abstract

In this paper, we propose a novel rigid motion segmentation algorithm called randomized voting (RV). This algorithm is based on epipolar geometry, and computes a score using the distance between the feature point and the corresponding epipolar line. This score is accumulated and utilized for final grouping. Our algorithm basically deals with two frames, so it is also applicable to the two-view motion segmentation problem. For evaluation of our algorithm, Hopkins 155 dataset, which is a representative test set for rigid motion segmentation, is adopted; it consists of two and three rigid motions. Our algorithm has provided the most accurate motion segmentation results among all of the state-of-the-art algorithms. The average error rate is 0.77%. In addition, when there is measurement noise, our algorithm is comparable with other state-of-the-art algorithms.

1. Introduction

In the video processing, recognizing different motions among rigid objects is an essential process such as video compression, moving object segmentation and motion analysis. Many researchers have tried to resolve this problem. The motion segmentation algorithms can be classified into two categories, according to the number of views. Two-view-based motion segmentation is a traditional computer vision problem, which has been typically approached from a geometrical perspective [16, 20, 10, 7, 13]. Since the Hopkins 155 motion segmentation dataset [17] was released, various multiview-based approaches have been widely developed and applied to the dataset. The principal angles configuration (PAC) [21] and sparse subspace clustering (SSC) [4] methods have shown particularly successful results. These algorithms are robust to noises, but they are extremely slow, so they are difficult to apply to real applications. To compensate for this weakness, incrementally-complex local models-(ICLM) based motion segmentation algorithm was proposed [5]. This algorithm is fast as well as robust to data noise, comparable to state-of-the-art algorithms. However, it has lower accuracy than other state-of-the-art algorithms for noise-free data.



(a) Original Video (b) Point Sampling

(c) $\mathbf{F}$ for Red Points ($\mathbf{G}_1$) (d) $\mathbf{F}$ for Green Points ($\mathbf{G}_2$)

(e) Voted Score (f) Segmentation Result

Figure 1. **Concept of the proposed method.** (a) The video is obtained from [15]. (b) The red circle indicates the supporting points for extracting fundamental matrix for $\mathbf{G}_1$ and the green circle points estimate the fundamental matrix for $\mathbf{G}_2$. (c),(d) Epipolar lines corresponding to the fundamental matrices and circled points. (e) Voting score for each point using the distance between the epipolar line and the point. (f) Final segmentation result.

In this paper, we focus on three issues, as follows:

- We develop a motion segmentation algorithm that works within a **reasonable time** and has **high accuracy with noise-free data**.
- **Robustness** to noise is also satisfied.

- Even though there are only **a few frames** (only two frames in worst cases), the performance should not be degraded.

We propose a randomized voting(RV) method as shown in Figure 1, which votes high scores for correctly estimated motion, and low scores for invalid motion. This scheme helps in that correctly estimated motion from arbitrary points survive, and wrong one disappear. This phenomenon is our algorithm's fundamental principle, and it is useful for solving motion segmentation problems. Compared to the traditional two-view-based approach, the proposed randomized voting scheme is also based on the epipolar geometry but the key difference is that randomized voting does not throw away the information provided by epipolar geometry even when the points selected to estimate the fundamental matrix do not belong to the same motion.

As a result, our algorithm is the most accurate among state-of-the-art algorithms, performing in a reasonable time. The algorithm is also robust to the noise comparable with other state-of-the-art algorithms. In particular, our algorithm is applicable to both two-view and multiview motion segmentation problem, in contrast to other state-of-the-art algorithms.

1.1. Related Work

Various approaches for solving the motion segmentation have been proposed such as multibody factorization [2], subspace clustering [4, 8], subspace fitting [19], epipolar geometry method [16, 20, 10, 7, 13], agglomerative lossy compression [14] and other approaches [23]. Those algorithms can be classified into two groups; two and multi view-based approaches.

Two-view-based approaches are usually based on epipolar geometry. One of recently proposed approaches tried to deal with outliers by using RANSAC and enhanced Dirichlet process mixture models [7]. Another approach [13] defined global dimension minimization in order to reveal the clusters corresponding to the underlying subspace. These two approaches use only two frames for motion segmentation. In other words, even though multiple frames are available, these approaches do not have an ability to utilize motion information in multiple frames.

Multiview-based approaches utilize the trajectory of feature points unlike two-view-based approaches. PAC [21] and SSC [18] methods have quite accurate results in both two and three motion cases. Furthermore, those algorithms are also robust to noise. However, those algorithms are extremely slow as mentioned in Section 1. Latent low-rank representation-based method (LatLRR) [11] is faster and more accurate, but this method is degraded in extremely noisy environments (Figure 6). ICLM-based approach [5], which has been recently proposed, is very fast, but has lower accuracy than other state-of-the-art approaches.

Such multiview-based approaches are more accurate than two-view-based approaches, but they do not have a good performance when there are only a few frames (Figure 7).

1.2. Background

Epipolar geometry is the intrinsic projective geometry between two views, so it depends on the camera internal parameters and relative pose. The fundamental matrix represents this epipolar geometry, which satisfies the following constraint:

$$x_2^T \mathbf{F} x_1 = 0, \tag{1}$$

where the fundamental matrix $\mathbf{F}$ is a 3×3 matrix of rank 2, x_1 and x_2 are homogeneous coordinates; $x_1 = [x \;\; y \;\; 1]^T$; and $x_2 = [x' \;\; y' \;\; 1]^T$.

In equation 1, $\mathbf{F}x_1$ represents an epipolar line in the other image and x_2 lies on the line, so the equation should be zero. However, in real environments, there are certain errors, such as measurement, quantization, and round-off errors; due to these errors, the equation is not exactly zero.

Consequently, for the matrix $\mathbf{F}$ estimation, the least square method can be utilized. Equation 1 can be rearranged as follows:

$$a^T f \approx 0, \tag{2}$$

where $a = [xx' \;\; xy' \;\; x \;\; yx' \;\; yy' \;\; y \;\; x' \;\; y' \;\; 1]^T$ and vector $f \in \mathbb{R}^9$ is obtained by scanning each row of $\mathbf{F}$. Consequently, a collection of equations 2 for m-points can be rearranged as follows:

$$\mathbf{A} f \approx 0, \tag{3}$$

where $\mathbf{A}$ is an $m \times 9$ matrix, which is a collection of vector a^T. Also, the m is the number of corresponding points.

The degree of freedom of the matrix $\mathbf{F}$ is eight since the ninth element is fixed as one; therefore, at least eight corresponding points are required for the fundamental matrix estimation. However, the matrix is of rank 2, so seven is the minimal number of points required.

$$\hat{f} = \arg\min_f ||\mathbf{A}f||, \tag{4}$$

The $\hat{f}$ is easily estimated by using a singular value decomposition method [6]. This method is known as direct linear transformation (DLT).

1.3. Notation

Before explaining our proposed algorithm, the notation is defined in this section. We usually use terms like "group" and "cluster" interchangeably; the notation is defined as follows:

$$\mathbf{X} = \mathbf{G}_1 \cup \mathbf{G}_2 \cup \ldots \cup \mathbf{G}_c, \tag{5}$$

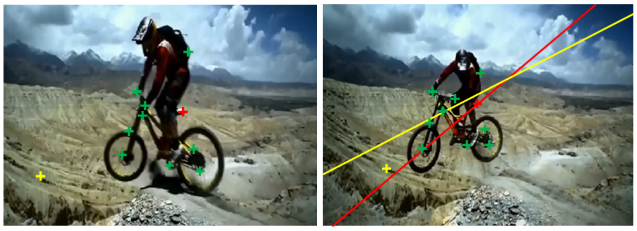

(a) Frame k (b) Frame l

Figure 2. **Motion segmentation with epipolar geometry.** (a) Green points are used for the estimation of the fundamental matrix. The red cross is the point existing on the bicycle (same motion with the green cross), and the yellow cross represents the point having different motion. (b) The red and yellow line are extracted by the red and yellow points with $\mathbf{F}$ estimated from the green points in frame k. The corresponding red point in frame l lies on the red line, but the corresponding yellow point does NOT lie on the yellow line.

$\mathbf{X}$ is the set of whole points and $\mathbf{G}_i$ the disjoint set of points belonging to the ith group where $i = 1, \ldots, c$. Also, c represents the total number of clusters.

To express points, we use x_i where $i = 1, \ldots, n$. This is a homogeneous coordinate of the point, and n is the total number of points. Hence, the point set X is defined as follows:

$$\mathbf{X} = \{x_1, x_2, \ldots, x_n\}, \tag{6}$$

In the video, there are many frames, so we express the frame index by a superscript, such as $x_i^{(j)}$ where $i = 1, \ldots, n$, $j = 1, \ldots, f$, where f is the total frame number of the video. Therefore, $x_i^{(j)}$ represents the homogeneous coordinate of the i-th point in frame j. A superscript without parenthesis is defined as an iteration number. In other words, $\mathbf{G}_i^t$ means the i-th group at iteration t. Also, we define matrix $\mathbf{F}^{(k,l)}$ as a fundamental matrix estimated by the k and l-th frames. The histogram storing voting score is denoted by $h[\cdot]$, and $h_i[\cdot]$ means the histogram for point $i = 1, \ldots, n$ with c bins.

2. Methodology

2.1. Motion Segmentation with Epipolar Geometry

The proposed motion segmentation algorithm is based on the epipolar geometry, which is represented by matrix $\mathbf{F}$. This matrix $\mathbf{F}$, obtained from the moving object, indicates motion because it encapsulates the intrinsic geometry. We utilize two properties of the fundamental matrix for motion segmentation as follows:

Property 1. *Points on the same moving rigid object have the same matrix* $\mathbf{F}$.

Property 2. *If a set of points lies on the same moving object in an image, the corresponding points in the other frame lie on the corresponding epipolar lines. Even if the points are not used for estimating the matrix* $\mathbf{F}$, *the same moving object's corresponding points lie on the lines.*

For instance, if there are ten points on the same moving object, nine particular points can be used for estimating the matrix $\mathbf{F}$. Then, the distance between the remaining one point and the corresponding epipolar line, computed from the matrix $\mathbf{F}$, is zero, from equation 1. On the other hand, if a point does not lie on the corresponding epipolar line, two cases can be considered; the point does not lie on the same moving object, or the estimated matrix $\mathbf{F}$ is invalid. Figure 2 demonstrates these properties. We use the properties to solve the rigid motion segmentation problem. If the distance between the epipolar line and the point is small, the point is likely to belong to the same group and vice-versa.

Various methods, such as algebraic distance, and geometric distance and its approximation, are used to measure the distance between the point and the epipolar line [6]. We utilize the Sampson distance, which is the first order approximation of a reprojection error, as follows:

$$\mathbf{SD}(x, y, \mathbf{F}) = \frac{y^T \mathbf{F} x}{(\mathbf{F}x)_1^2 + (\mathbf{F}x)_2^2 + (\mathbf{F}y)_1^2 + (\mathbf{F}y)_2^2}, \tag{7}$$

where, for instance, $(\mathbf{F}x)_j^2$ represents the square of the j-th entry of the vector $\mathbf{F}x$. If we know the matrix $\mathbf{F}$ for each object, the sum of the distances over the whole frames is approximately zero when there are noises, as follows:

$$E(\mathbf{X}_k, \mathbf{F}_g) = \sum_{i=1}^{f} \sum_{j=1}^{f} \mathbf{SD}(x_k^{(i)}, x_k^{(j)}, \mathbf{F}_g^{(i,j)}) \approx 0, \tag{8}$$

where $\mathbf{F}_g$ is a set of fundamental matrices estimated from object g and $\mathbf{X}_k$ is a trajectory of the k-th point. Consequently, for motion segmentation, we can find the label that minimizes the function.

$$\hat{g} = \arg\min_g E(\mathbf{X}_k, \mathbf{F}_g), \tag{9}$$

where $\hat{g}$ is an estimated group of the k-th point with $\mathbf{F}_g$. Thus, if the fundamental matrices for all the motions are known, whole points can be easily clustered into the corresponding objects and vice-versa. However, in motion segmentation, both labels and fundamental matrices are unknown. This is a chicken and egg problem.

2.2. Randomized Voting Algorithm

As mentioned in the previous section, the proposed epipolar geometry-based motion segmentation approach is difficult due to the chicken and egg problem. To resolve the problem, we approach it using an iterative method. First, the

Algorithm 1 Randomized Voting

```
Input: $\{\mathbf{X}_i, \mathbf{G}_g, h_i[\cdot]\}; i = 1, \dots, n; g = 1, \dots, c$
Output: $\{\mathbf{G}_g, h_i[\cdot]\}; i = 1, \dots, n; g = 1, \dots, c$

$k, l \leftarrow$ random frame number
for each group $\mathbf{G}_g \subset \mathbf{X}$ do
    $p \sim U(0, 1)$ // uniform random variable
    if $p < T_p$ then
        Select $m$-points in $\mathbf{G}_g$ with the highest votes
    else
        Randomly select $m$-points from $\mathbf{G}_g$
    end if
    Estimate $\mathbf{F}_g$ using selected $m$-points
    for each point $x_i^{(k)} \in \mathbf{X}$ do
        $d_i = \mathbf{SD}(x_i^{(k)}, x_i^{(l)}, \mathbf{F}_i^{(k,l)})$
        $h_i[g] = h_i[g] + e^{-\lambda d_i}$
        $h_i[\tilde{g}] = h_i[\tilde{g}] - e^{-\lambda d_i} + 1$ for all $\tilde{g} \neq g$
    end for
end for

Update each $\mathbf{G}_g, g = 1, \dots, c$
$h_i[\cdot] \leftarrow \alpha h_i[\cdot]$ for $i = 1, \dots n$
```

labels are initialized randomly, and the fundamental matrices are estimated using the initial labels. Next, we update the labels using the fundamental matrices, and the process is repeated. In this process, we adopt a voting scheme to utilize the information from a previously estimated fundamental matrices. We call this algorithm "randomized voting". This method is well illustrated in Figure 3.

Our algorithm randomly selects m-points and estimates the fundamental matrix $\mathbf{F}$ using the selected points. Then, the Sampson distance is calculated by using the matrix $\mathbf{F}$ for all the points. We vote scores for the whole points based on the distance. Finally, the label is assigned according to the value of the score. The randomized voting algorithm is explained in Algorithm 1.

2.2.1 Random Frame Selection

For the estimation of a matrix $\mathbf{F}$, two-view images are necessary, basically. Neighboring frames can definitely be used as two-view images, but the objects movements may not show up. Furthermore, this can lead to inaccurate estimation of $\mathbf{F}$, so a different motion may not be well separated.

To avoid this problem, we utilize all combinations of views, and this method generates many more cases; if there are f frames, we have $\binom{f}{2}$ combinations of views. However, to deal with all cases can be a hard work, if f is increasing, so we utilize a subset of the combinations of views by selecting two frames randomly.

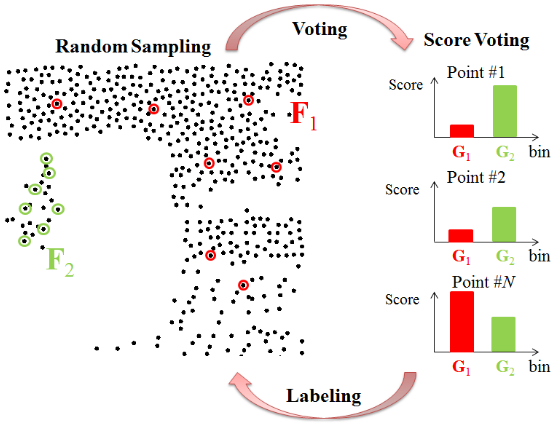


Figure 3. **Randomized voting.** To resolve the chicken and egg problem, our algorithm initializes the points as random label. Based on the label, each matrix $\mathbf{F}$ is estimated and we vote scores for each point. Then, each point is relabeled using the score. This process is iteratively performed to find the solution.

2.2.2 m-Points Random Sampling

When m-points are selected for the estimation of $\mathbf{F}$, most of points should lie on the same moving object. In order to satisfy this condition, neighboring feature points may be useful; because they are located close to each other, so they are more likely to belong to the same moving object. However, the movement of feature points in the local area can be different from the object's motion. Therefore, we present another method to select m-points using the accumulated histogram.

In our method, with probability $(1 - T_p)$, m-points are uniformly and globally selected from the same cluster, but with probability T_p they are selected from among best-voted m-points. The best-voted points are occasionally used to increase the probability of the points belonging to same object. If this method is solely used, the matrix $\mathbf{F}$ may be inaccurate due to fitting to the local motion. Finally, this causes wrong segmentation results.

For instance, suppose there is a long rectangular object that is rotating and the instantaneous center is the center of the object. When the points are selected far from the instantaneous center, the matrix $\mathbf{F}$ cannot represent the entire motion of the object well, due to the local motion ambiguity with the translation motion. To prevent this phenomenon, we also select m-points randomly from all the points belonging to the corresponding group.

2.2.3 Fundamental Matrix Estimation

We utilize the fundamental matrix $\mathbf{F}$ to represent the motion of an object. The matrix $\mathbf{F}$ can be estimated using several methods such as normalized-DLT and gold-standard algorithm. Most representative method for estimating the matrix $\mathbf{F}$ is a normalized-DLT method [6], which minimizes algebraic error. The solution is simply computed by using a singular value decomposition, so it is fast and efficient. However, because it minimizes algebraic error, this method can be less accurate compared to other methods like a gold-standard algorithm [6].

Our algorithm iteratively finds a solution, therefore the estimation should be performed several times. Any algorithm can be used for the estimation of the matrix $\mathbf{F}$, but we adopt normalized-DLT method to make faster algorithm. In the proposed algorithm, the estimation error does not matter, because the error effect is decreased during voting score.

2.2.4 Score Voting

For computing the score of a point, we use the Sampson distance, which is a suitable metric to use as an evaluation criteria. Once the fundamental matrix $\mathbf{F}$ is estimated, the distance $d_i = \mathbf{SD}(x_i^{(k)}, x_i^{(l)}, \mathbf{F}_g^{(k,l)})$ between the epipolar line from the matrix $\mathbf{F}_g^{(k,l)}$, and each point $x_i^{(k)}$ is calculated, where $i = 1, \ldots, n$, $g = 1, \ldots, c$. Then each point $x_i \in \mathbf{X}$ is voted by $e^{-\lambda d_i}$ where the parameter λ controls the voting strength. If the value of λ is small, then it gives a large voting value to the point and vice-versa. This means that the lower value, more robust, but discriminative power also decreases.

$$h_i[g] \leftarrow h_i[g] + e^{-\lambda d_i}, \tag{10}$$

where $h_i[g]$ is a histogram with bin $g = 1, \ldots, c$ and $i = 1, \ldots, n$. In other words, if the distance value is large, a histogram of the point is accumulated by small values where the accumulated histogram has c bins. In this step, our algorithm also votes scores to other groups with the same $d_i = \mathbf{SD}(x_i^{(k)}, x_i^{(l)}, \mathbf{F}_g)$, as follows:

$$h_i[\tilde{g}] \leftarrow h_i[\tilde{g}] - e^{-\lambda d_i} + 1 \quad \text{for all} \quad \tilde{g} \neq g. \tag{11}$$

This method helps to escape easily from incorrect label.

If the algorithm is repeated infinitely, the accumulated value of the bin of the histogram $h_i[g]$ represents the likelihood of point x_i belonging to the corresponding group $\mathbf{G}_g$; and for $i = 1, \ldots, n$ we assign point i to group $\mathbf{G}_{\hat{g}}$, where

$$\hat{g} = \arg\max_g h_i[g] \quad \text{for} \quad i = 1, \ldots, n. \tag{12}$$

It is natural that each point belongs to the highest voted cluster. Finally, the histogram decays by multiplying a forgetting factor α, because incorrect information can exist in the accumulated score. If the α is small, incorrect information can be reduced, but the algorithm is difficult to converge. On the contrary, if its value is large, the algorithm is easily converged. However, it does not have good performance.

Algorithm 2 Motion Segmentation Algorithm

```
Input: $\mathbf{X_i}, i = 1, \ldots, n$ // points trajectories.
Output: $\mathbf{G}_g, g = 1, \ldots, c$

t ← 1 // initialize the trial number
repeat
    $t \leftarrow t + 1$
    Randomly initialize $\mathbf{G}_g^t$ for $g = 1, \ldots, c$
    $h_i^t[g] \leftarrow 0$ for $i = 1, \ldots, n$; $g = 1, \ldots, c$
    $k \leftarrow 0$ // initialize the voting number
    for $k < T_i$ do
        $\{\mathbf{G}_g^t\}_{g=1}^c, \{h_i^t[\cdot]\}_{i=1}^n$
        $\leftarrow$ RV( $\{\mathbf{X}_i\}_{i=1}^n, \{\mathbf{G}_g^t\}_{g=1}^c, \{h_i^t[\cdot]\}_{i=1}^n$)

        if $k = T_r$ then // reinitialization at $T_r$
            $h_i^t[g] \leftarrow 0$ for $i = 1, \ldots, n$; $g = 1, \ldots, c$
        end if
        $k \leftarrow k + 1$
    end for
until $t = T$ or $\{\mathbf{G}_g^t\}_{g=1}^c$ is not changed over $T_c$

if $\{\mathbf{G}_g^t\}_{g=1}^c$ is not changed over $T_c$ then
    $\mathbf{A} \leftarrow$ AffinityMatrix($\{h_i^t[\cdot]\}_{i=1}^n, \{\mathbf{G}_g^t\}_{g=1}^c$)
    $\mathbf{G}_g \leftarrow$ SpectralClustering($\mathbf{A}$)
else
    $\mathbf{G}_g \leftarrow \mathbf{G}_g^t$ for $g = 1, \ldots, c$ // when converged
end if
```

2.3. Motion Segmentation

In this section, we present a motion segmentation algorithm using the randomized voting algorithm results. This algorithm is explained in Algorithm 2. First, it starts as a randomly initialized group, and performs randomized voting until T_i iteratively. Then, the histogram is accumulated by randomized voting process, but it is possible to be incorrect due to random initialization, so our algorithm reinitializes each histogram at T_r while preserving $\mathbf{G}_g$, where $g = 1, \ldots, c$.

To define the termination condition, we set up two criteria: iteration number and convergence. To test for convergence, our algorithm checks to insure that the randomized votings result is not changed during T_c. If T_c is large, our algorithm can be more stable, because it mostly depends on the spectral clustering. If the results are not converged, our algorithm is terminated by force at $t = T$.

The result of the randomized voting is reliable when the result is converged. However, if the iteration is terminated without convergence until the maximum number of trials, the results are less reliable. In other words, the results can vary, so each point cannot belong to a particular group. In this case, our algorithm aggregates all the his-

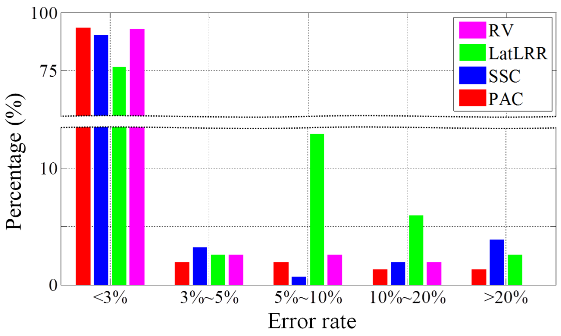


Figure 5. **Distributions of 155 error rates.** Our algorithm is compared to state-of-the-art algorithms.

tograms and grouping results over the T trials to decide the final grouping. To this end, we utilize a spectral clustering method [12].

First, we define matrix $\mathbf{H}^t$ as follows:

$$\mathbf{H}^t = \begin{bmatrix} h_1^t[1] & h_1^t[2] & \cdots & h_1^t[c] \\ h_2^t[1] & h_2^t[2] & \cdots & h_2^t[c] \\ \vdots & \vdots & \ddots & \vdots \\ h_n^t[1] & h_n^t[2] & \cdots & h_n^t[c] \end{bmatrix}, \qquad (13)$$

where $\mathbf{H}^t$ is an $n \times c$ matrix. We build the matrix $\mathbf{V}$ using $\mathbf{H}^t$ and $\mathbf{G}^t$.

$$\mathbf{V} = \left[\mathbf{H}^1 \quad \cdots \quad \mathbf{H}^T \quad \mathrm{l}(\mathbf{G}^1) \quad \cdots \quad \mathrm{l}(\mathbf{G}^T)\right], \qquad (14)$$

where $\mathbf{V}$ is an $n \times (c+1)T$ matrix. Also, $\mathrm{l}(\cdot)$ is a column vector of the label index.

Finally, affinity matrix $\mathbf{A}$ is built using an inner product between the row of $\mathbf{V}$ and column of $\mathbf{V}$ as follows:

$$\mathbf{A} = \mathbf{V}\mathbf{V}^T, \qquad (15)$$

where matrix $\mathbf{A}$ is an $n \times n$ affinity matrix. This $\mathbf{A}$ matrix is used as an input for the spectral clustering.

Figure 4 (a), (b) shows the final motion segmentation results and the (c) is final voting score for each group. While iterating the randomized voting algorithm, the score increases as shown in Figure 4 (d), (e), (f). Especially, when the k reaches to the T_r, the score is initialized to the zero.

3. Experimental Evaluation

We adopted the Hopkins155 dataset [17] to evaluate our algorithm. This dataset has been commonly used for testing motion segmentation problems. It consists of 155 sequences: 120 and 35, for two and three rigid motions, respectively. We compared the error rate of the proposed algorithm with those of state-of-the-art methods.

RANSAC is basic subspace fitting algorithm, and a generalized principal component analysis (GPCA) is proposed in [19]. SCC means spectral curvature clustering which is presented in [1]. The spectral clustering (SC) method and ranking of locally sampled subspaces (RLS) are presented in [9] and [3], respectively. Moreover, LCV [23] is linear combination of views based algorithm. SSC [4], PAC [21], LatLRR [11], and ICLM [5] are mentioned in Section 1.1. Also, LBF is local best-fit flats [22].

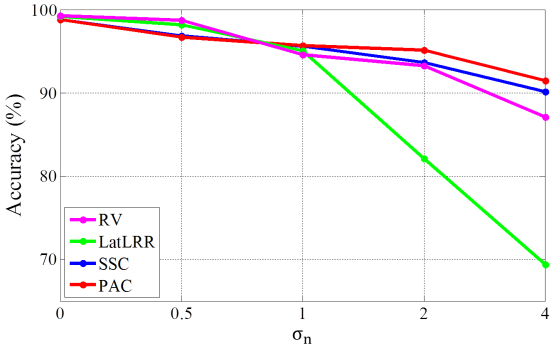


Figure 6. **Accuracy with multiple levels of Gaussian noise.** We performed just one test for SSC and PAC due to high computation time requirements. Our algorithm RV, and LatLRR were performed 10 times and 20 times, respectively.

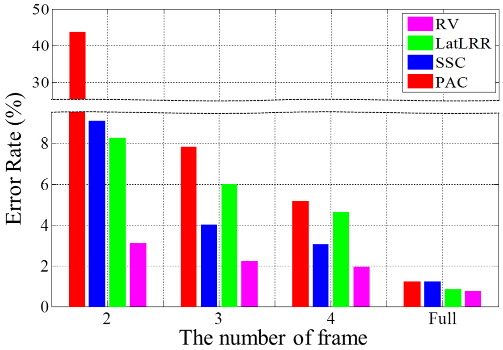
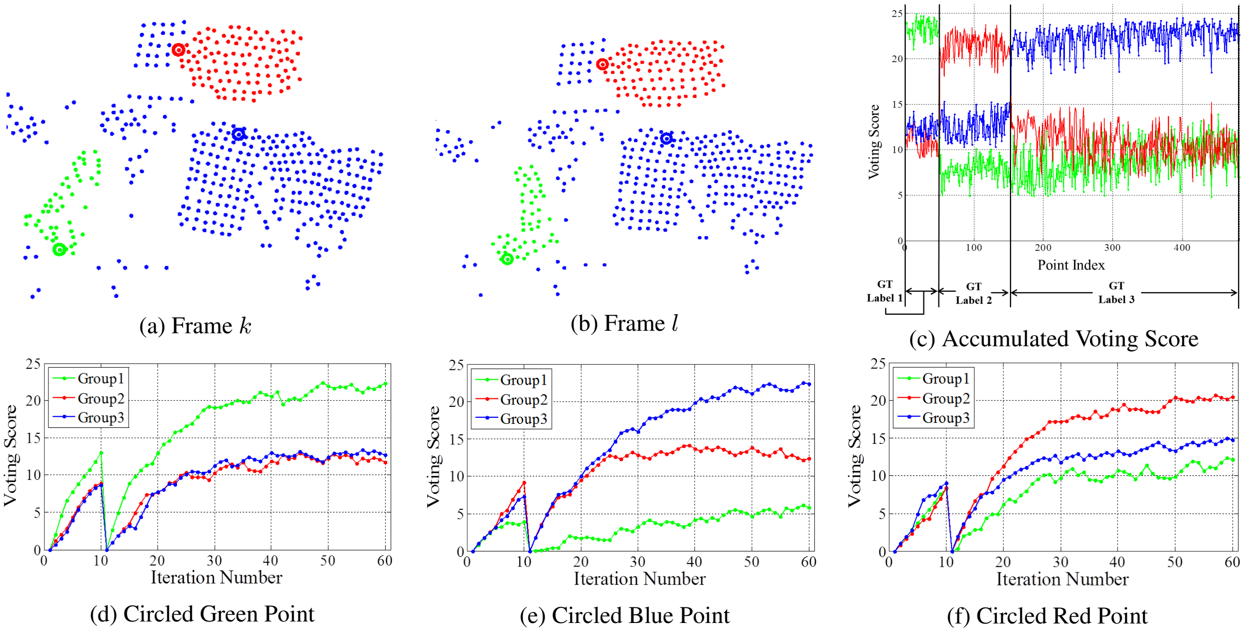


Figure 7. **The average error rates according to the number of input frames.** Our algorithm showed the lowest average error rates even if there are few frames.

Accuracy with noise-free data. Table 1 lists average error rates according to two-motions, three-motions, and both. Our algorithm is based on random initialization and selection, so there were variations. Consequently, we repeated our algorithm 150 times, and the average was computed. Our algorithm achieved a 0.77% average error rate, and its standard deviation was 0.27%. Similarly, we achieved 0.44% and 1.88% average error rates for two- and three-

	SSC	PAC	LatLRR	ICLM	**RV**
sec	11026	123960	557	217	1017

Table 2. **Total processing times.** These are measured in the same computer system, except for ICLM.



(a) Frame k

(b) Frame l

(c) Accumulated Voting Score

(d) Circled Green Point

(e) Circled Blue Point

(f) Circled Red Point

Figure 4. **Demonstration of voting score.** (a),(b) Segmentation results with two-view. Green points belong to Group 1, and red and blue points are assigned to Groups 2 and 3, respectively. (c) Final voting scores for points. The point index is rearranged according to the ground truth (GT) label. (b),(e),(f) Voting scores for points circled by red, blue, and green, respectively, in (a).

	RANSAC	GPCA	SCC	LCV	SSC	SC	LBF	PAC	LatLRR	RLS	ICLM	**RV**
Total	9.48	10.02	2.70	1.86	1.24	1.2	1.35	1.24	0.85	1.03	3.81	**0.77**
Two	5.56	4.59	1.77	1.25	0.82	0.94	0.98	0.96	n/a	0.65	n/a	**0.44**
Three	22.94	28.66	5.89	3.97	2.45	2.11	2.64	2.22	n/a	2.11	n/a	**1.88**

Table 1. **Total error rates.** The first row represents average error rates for all (155 sequences). The second and third rows are two (120 sequences) and three (35 sequences) motions, respectively. N/a means the value is not presented in the corresponding paper.

motions cases.

Figure 5 shows the distribution of 155 error rates. Our algorithm is comparable to the best performance (PAC) when the error rate is below 3%. Using our algorithm, there were no cases with error rates greater than 20%.

Robustness test. For another experimental setup, under noisy environmental conditions, we directly added artificial noise to the raw feature points, in contrast to method [5]. We used four Gaussian noises with a zero mean and different diagonal covariance matrix $\mathbf{\Sigma}_n = \sigma_n^2\mathbf{I}$, $\sigma_n \in \{0.5, 1, 2, 4\}$, where $\mathbf{I}$ is a 2×2 identity matrix. The accuracies with the noise are shown in Figure 6. Our algorithm showed comparable performance to the others.

The number of frame. To evaluate the performance according to the input number of frames, we manually chose the frames for three cases as follows: 1st and 5th frames (1-5) for two-view, 1-5-10 for three-view, and 1-5-10-15 for four-view. As depicted in Figure 7, the average error rates tend to decrease as the number of input frames increase. In all three cases, the performance of our algorithm showed the lowest error rate when compared to the other algorithms.

Computational time. Table 2 indicates the total processing times for the state-of-the-art algorithms with full frames. We measured the computation times in the same computer system. Our algorithm's running time was 1 or 2 orders of magnitude faster than SSC and PAC, and close to LatLRR in terms of magnitude. The computation time for ICLM was obtained from [5] due to the absence of source code.

4. Conclusion

In this work, we proposed a novel rigid motion segmentation algorithm based on the randomized voting method, which was inspired by a voting scheme. Our proposed algorithm has an ability to provide large scores for correctly estimated motion and low scores for invalid motion.

As a result, our algorithm achieved excellent performances with average error rates of 0.44% for two-motions, 1.88% for three-motions, and 0.77% for the total motion using full frame. Additionally, our algorithm is also applicable to two-view motion segmentation, because our algorithm is basically based on the two-view-based framework. Consequently, we applied our algorithm to few-frame-based motion segmentation problem, and our algorithm was not degraded like other state-of-the-art algorithms.

Remaining issues for our algorithm are as follows: We did not consider controlling outliers in this paper. However, this is also important issue in motion segmentation problem. Also, our algorithm's accuracy decreases, when the σ of noise has a high value.

In conclusion, our algorithm achieved, within a reasonable time, the highest performance of all other state-of the-art algorithms, and also achieved comparable accuracy under noisy environmental conditions.

Acknowledgements

This work was supported in part by the Technology Innovation Program, 10045252, Development of robot task intelligence technology, funded by the Ministry of Trade, industry & Energy (MOTIE, Korea). Furthermore, this research was also supported in part by Leading Industry Development for Chung-cheong Economic Region (A0022 00627) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (2010-0028680).

References

[1] G. Chen and G. Lerman. Motion segmentation by scc on the hopkins 155 database. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 759–764. IEEE, 2009. 6

[2] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998. 2

[3] N. Dimitriou and A. Delopoulos. Improved motion segmentation using locally sampled subspaces. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 309–312. IEEE, 2012. 6

[4] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009. 1, 2, 6

[5] F. Flores-Mangas and A. D. Jepson. Fast rigid motion segmentation via incrementally-complex local models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2259–2266. IEEE, 2013. 1, 2, 6, 7

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000. 2, 3, 4, 5

[7] Y.-D. Jian and C.-S. Chen. Two-view motion segmentation with model selection and outlier removal by ransac-enhanced dirichlet process mixture models. *International Journal of Computer Vision*, 88(3):489–501, 2010. 1, 2

[8] F. Lauer and C. Schnorr. Spectral clustering of linear subspaces for motion segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 678–685. IEEE, 2009. 2

[9] F. Lauer and C. Schnorr. Spectral clustering of linear subspaces for motion segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 678–685. IEEE, 2009. 6

[10] H. Li. Two-view motion segmentation from linear programming relaxation. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007. 1, 2

[11] G. Liu and S. Yan. Latent low-rank representation for subspace segmentation and feature extraction. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1615–1622. IEEE, 2011. 2, 6

[12] A. Y. Ng et al. On spectral clustering: Analysis and an algorithm. 2002. 6

[13] B. Poling and G. Lerman. A new approach to two-view motion segmentation using global dimension minimization. *International Journal of Computer Vision*, pages 1–21, 2013. 1, 2

[14] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(10):1832–1845, 2010. 2

[15] TheCoolTrick. Best of freeride downhill - mtb. Retrieved from http://www.youtube.com/watch?v=szA983CnNPg. 1

[16] P. H. Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356(1740):1321–1340, 1998. 1, 2

[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. 1, 6

[18] R. Vidal. Subspace clustering. *Signal Processing Magazine, IEEE*, 28(2):52–68, 2011. 2

[19] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1945–1959, 2005. 2, 6

[20] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. Segmentation of dynamic scenes from the multibody fundamental matrix. *Urbana*, 51:61801, 2002. 1, 2

[21] L. Zappella, E. Provenzi, X. Lladó, and J. Salvi. Adaptive motion segmentation algorithm based on the principal angles configuration. In *Computer Vision–ACCV 2010*, pages 15–26. Springer, 2011. 1, 2, 6

[22] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Randomized hybrid linear modeling by local best-fit flats. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1927–1934. IEEE, 2010. 6

[23] V. Zografos and K. Nordberg. Fast and accurate motion segmentation using linear combination of views. In *Proc. BMVC*, pages 12.1–12.11, 2011. http://dx.doi.org/10.5244/C.25.12. 2, 6