

# IACV Project : Robust Motion Segmentation

Pınar Erbil  
10953514

Prachi Kedar  
10948167

## Abstract

*This paper challenges the rigid choice between fundamental matrix and homography models for motion segmentation in complex real-world scenes. It argues that forcing categorization into "general" or "degenerate" motion is impractical and leads to limitations. Even in "general" cases, the fundamental matrix approach exhibits shortcomings. Therefore, the proposed approach utilizes multi-view motion segmentation for improved flexibility. Instead of K-means clustering, unsuitable for high-dimensional data, the study explores alternative techniques like DBSCAN, Hierarchical clustering, and Fuzzy C Mean clustering. Additionally, the impact of considering non-consecutive frame pairs is evaluated. Finally, it compares the performance of RANSAC, MSAC, and LMedS for fitting homography models. This research aims to move beyond the limitations of traditional approaches and present a more robust and versatile solution for motion segmentation in challenging scenarios.*

## 1. Introduction

Motion segmentation is an ongoing research topic in the literature with its applications in surveillance systems, scene reconstruction algorithms, video compression algorithms, and many medical imaging approaches. Motion segmentation aims to identify different moving objects in a given setting.

The geometric approaches [8, 3] proposed as the solution to the motion segmentation problem suffer from real-world data. They can handle motions under specific scenarios, and under complex motions, they fail. Later, Spectral clustering gained popularity by handling complex motions better. However, the dataset these algorithms test their performance on are not good enough in real-world scenarios. They do not have multiple motions and fail to represent the vanishing camera translation effect where the camera position is important in reality.

In [19], a new dataset is proposed to solve this problem and several spectral clustering algorithms to solve the motion segmentation, but this approach is not robust to outliers.

In this paper, we present several ways to improve the algorithm by changing the clustering algorithm, consecutive pairs and fitting the homography function.

## 2. Related work

One of the most important challenges in computer vision is motion segmentation, which is the art of dividing a scene into separate moving objects. The use of geometric models and multi-view cameras to handle this complexity has advanced significantly in the last few years. This review of the literature explores advances in robust motion segmentation, with an emphasis on methods that use geometric models from different points of view.

Early proposed methods for the motion segmentation problem were factorization methods [14, 2, 5]. These methods can only work with fully independent motions, and when we have a moving camera, the motions become dependent, causing factorization methods to not fit.

Later, [16, 17] presented the Generalized Principal Component Analysis (GPCA) method. It is an algebraic method to segment the subspaces of different dimensions. However, this method does not work well with increased complexity and larger data when the ambient space's dimension and number of motions increase.

Lastly, we can talk about spectral clustering algorithms [11, 7, 6, 19, 20, 1] for the motion segmentation problem. The algorithm uses the  $N \times N$  eigenvectors of the affinity matrix to calculate the embeddings.

## 3. Proposed approach

### 3.1. Base Method

We take the method proposed in [19] as our base model. The algorithm consists of 3 main parts.

1. **Geometric Model Hypothesis:** Randomly sample  $\mathbf{p}$  points visible in both frames and use these  $\mathbf{p}$  points to fit the model. We have affine ( $\mathbf{p}=3$ ), homography ( $\mathbf{p}=4$ ) and fundamental matrix ( $\mathbf{p}=8$ ).
2. **Computing Ordered Residual Kernek (ORK):** Calculates the residuals between each data point automatically without a need for a threshold.

3. **Spectral Clustering:** There are 3 single-view and 3 multi-view spectral clustering approaches presented.

**Single-view Spectral Clustering:** In single-view spectral clustering given the affinity matrix  $K$  and degree matrix  $D$ , they compute

$$\min_U \text{tr}(U^T L U)$$

such that  $U$  is a spectral embedding with  $U U^T = I$  and  $L$  is the normalized Laplacian  $L = I - D^{-0.5} K D^{-0.5}$

**Kernel Addition Multi-view Spectral Clustering:** Simple sum of all affinity matrices  $K_v$ 's:

$$\min_{U_v} \sum_v \text{tr}(U_v^T L_v U_v)$$

**Co-Regularization Multi-view Spectral Clustering:** Similar to Kernel addition, but now has an added component that encourages pair-wise consensus between two spectral embeddings. We have  $\lambda$  as the penalty coefficient, where the larger the lambda, the more similar it is to kernel addition.

$$\min_{U_v} \sum_v \text{tr}(U_v^T L_v U_v) - \lambda \sum_v \sum_{w, v \neq w} \text{tr}(U_v U_v^T U_w U_w^T)$$

**Subset Constrained Multi-view Spectral Clustering:** Introduces a subset constrained for the views. The inliers for a homography matrix are also inliers for a specific fundamental matrix, whereas the inliers for an affinity matrix are inliers for a certain homography one. This creates a hierarchical constraint  $K_A \leq K_H \leq K_F$ . This method uses this as a constraint to transform the problem:

$$\min_{U_v} \sum_v \text{tr}(U_v^T L_v U_v) - \gamma \text{tr}(U_v^T Q_v U_v)$$

$$Q_v = \begin{cases} \mathbb{1}(\hat{K}_{v+1} < 0) \circ \hat{K}_{v+1} & v = 1 \\ \mathbb{1}(\hat{K}_{v+1} > 0) \circ \hat{K}_{v-1} + \mathbb{1}(\hat{K}_{v+1} < 0) \circ \hat{K}_{v+1} & v = 2 \\ \mathbb{1}(\hat{K}_{v+1} > 0) \circ \hat{K}_{v-1} & v = 3 \end{cases}$$

where  $v = 1$  is affinity,  $v = 2$  is homography, and  $v = 3$  is fundamental

In the last step of spectral clustering, K-Means is applied on  $U$  and groups the points into  $M$  number of motions.

Our proposed approach consists of three parts. The first part consists of altering the clustering algorithm used, in the second part, we focus on changing the pairs used for the hypotheses. Currently, the algorithm uses consecutive pairs, we analyze the results for non-consecutive pairs. Lastly, we implemented MSAC and LMedS algorithms to fit the homography model to reduce the outliers.

## 3.2. Changing the Clustering Algorithm

There are several drawback of the clustering algorithm used by the base model. The method implemented in the previous approach was based on K-Means clustering algorithm which has some limitations such as:

- **Fixed Number of Clusters:** K-Means inputs the number of clusters and this can be challenging in real-life scenarios where one may not know the number of clusters beforehand.
- **Sensitivity to Noise and Outliers:** K-Means is sensitive to outliers and noise, which is the problem in our situation.
- **Assuming Spherical Clusters:** K-Means assumes clusters are spherical, this can especially in real-world data can create problems because we mostly have non-spherical objects.

### 3.2.1 Hierarchical

Hierarchical clustering [10] creates a hierarchy of clusters, allows to explore different granularities and potentially identifies hidden substructures within the motion patterns. This is helpful in understanding the complexity of the scene and uncovering finer details. A cluster tree or dendrogram is created using hierarchical clustering to group data across several scales. The tree is a hierarchical hierarchy, with clusters at one level joining to form clusters at the next level. This allows us to choose the level or scale of clustering that is most suited to the application.

### 3.2.2 DBSCAN

DBSCAN [13] automatically identifies clusters based on density, eliminating the need to predefine the number of clusters. This flexibility might be useful if the clusters have varying densities or irregular shapes. This makes it a good candidate for motion segmentation tasks, where the goal is to segment moving objects from the background based on their motion patterns.

### 3.2.3 Fuzzy C-Means

Fuzzy C-Means clustering (FCM) [18] is a type of clustering algorithm that allows data points to belong to multiple clusters with different degrees of membership. This is in contrast to traditional clustering algorithms, such as k-means clustering, which only allow data points to belong to a single cluster.

FCM is based on the idea of fuzzy sets, which are sets that can have a degree of membership between 0 and 1. In FCM, each data point is assigned a membership value for each cluster. The membership value for a data point in a

cluster indicates how strongly the data point belongs to that cluster.

The FCM algorithm works by iteratively updating the cluster centroids and the membership values of the data points.

### 3.3. Non-consecutive Pairs

The authors in [19] used consecutive pairs for their analysis. As another approach we experimented with pairs having 2 or 3 frame gaps, we will refer to them as 2-framegap and 3-framegap pairs throughout the paper.

The increased gap between the frames can better show the method’s performance when we have a more dynamic scene. Moreover, it helps the algorithm to distinguish the motion better and can increase the robustness to noise.

In addition to changing the pairs, we also changed the number of hypotheses generated per frame, and  $\lambda$  and  $\gamma$  values used in co-regularization and subset constrained multi-view spectral clustering methods.

### 3.4. Fitting Homography

Base model uses RANSAC to fit an homography and we implemented two other methods, MSAC and LMeds.

#### 3.4.1 MSAC

MSAC [9] is a robust estimation algorithm commonly used in tasks like motion segmentation, especially when dealing with outliers in data. It iteratively fits a model to a small set of randomly selected data points (“samples”) and then evaluates how well the model fits the remaining points. The cost function makes MSAC more robust to partially corrupted data. Moreover, it is more efficient for large datasets because it converges faster compared to RANSAC.

#### 3.4.2 LMeds

LMeds [12] is a robust estimation method that finds model parameters that minimize the median of the squared errors. It follows a median-based approach which makes it robust to outliers. Also, compared to RANSAC it is less sensitive to parameter choices like the inlier threshold.

## 4. Experiments

In this section, we will present our experiment results.

**Dataset:** We used the KITTI 3D Motion Segmentation Benchmark (KT3DMoSeg) [19] based on the KITTI dataset [4]. KT3DMoSeg investigates motion segmentation in self-driving settings and solves the limitations of other popular datasets, such as the Hopkins155 dataset [15], which fails to capture real-world circumstances and complex environments.

KT3DMoSeg has 22 video clips from KITTI, each of which is manually picked and annotated with a foreground object’s attributes for building motion hypotheses. Overall, the KT3DMoSeg dataset has 22 brief clips, with 10-20 frames.

**Base Model setup:** For the base model, we have

- Frame Gap = 1
- Number of hypotheses = 500
- $\lambda$  value for CoReg =  $10^{-2}$
- $\gamma$  value for Subset =  $10^{-2}$

### 4.1. Changing the Clustering Algorithm

From Figure 1 we can see that there is a small difference between mean error and median error as the value of alpha progresses for the affine model; whereas, in Figure 2 for the fundamental model, there is a significant difference between mean and median since it is not able to fit the model for a particular frame sequence. A similar pattern of fundamental model’s result is in Figure 3 for the homography model.

From Figures 4, 5, 6 we can see that DBSCAN works poorly for Fundamental because there is huge difference between mean and median error. This suggest more outliers are present in the data. On the other hand, the results for affine and homography shows that the DBSCAN is working good.

Lastly, we implemented Fuzzy C-Mean clustering algorithm. The results are in Figure 7, 8 and 9. We can see it is able to handle the outliers quite well because the difference between mean and median errors are small.

The overall results are reported in the Table 1, we get the values for alpha = 15. We can see that DBSCAN works the best for Affine and Homography, AHF\_Subset and for AHF\_Reg; whereas, Fuzzy C-Means works the best with the Fundamental and AHF\_KerAdd. Unfortunately, none of the models outperformed K-means which forces us to try again with different paramaters for clustering algorithms.

### 4.2. Non-consecutive Pairs

We experimented the algorithm with 2-framegap and 3-framegap pairs. Table 2 and Table 3 summarizes the average and median errors reported by the methods. We also change the hypotheses numbers and report the mean values for alpha = 9,10, and 11.

Most of the time, the error decreases with non-consecutive pairs. The results for 2-framegap pairs are more promising compared to 3-framegap ones that may be because it handles both short-term motion dynamics and occlusions better. We have a roughly average of 2% decrease in the average error rates over all methods.

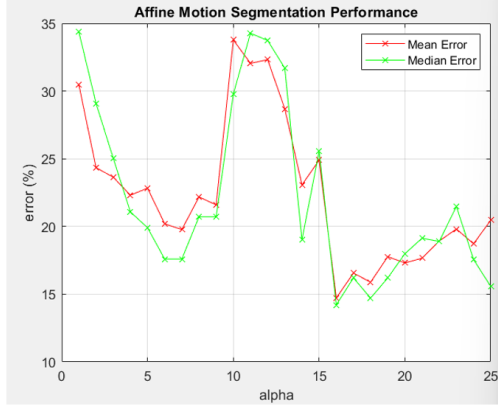


Figure 1. Hierarchical Clustering for Affine

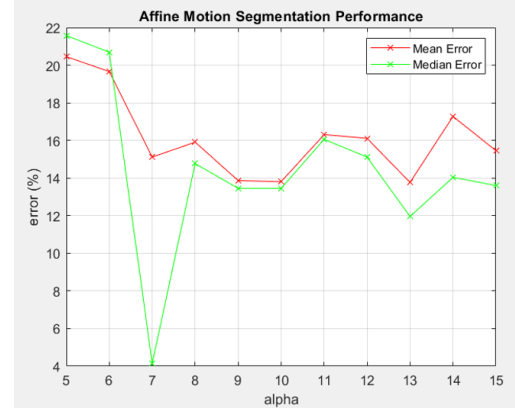


Figure 4. DBSCAN for Affine

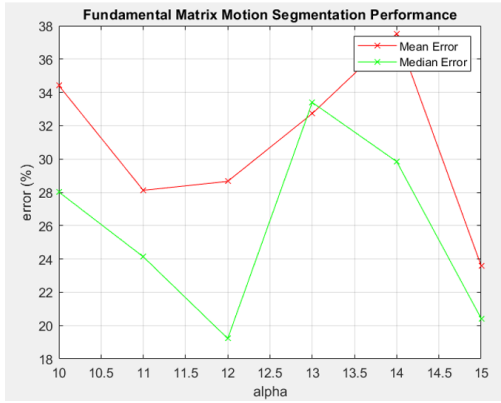


Figure 2. Hierarchical Clustering for Fundamental

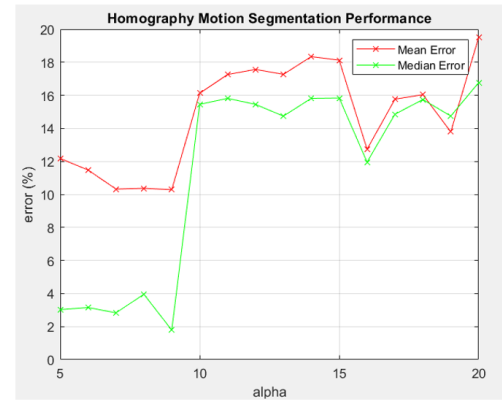


Figure 5. DBSCAN for Homography

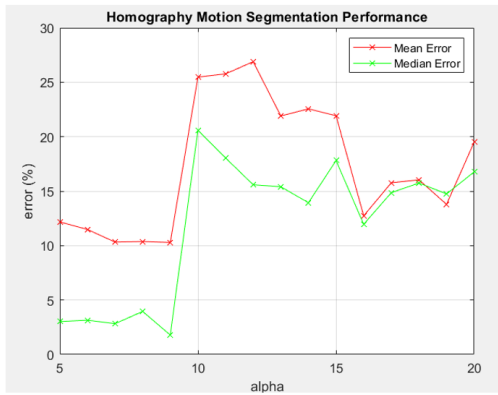


Figure 3. Hierarchical Clustering for Homography

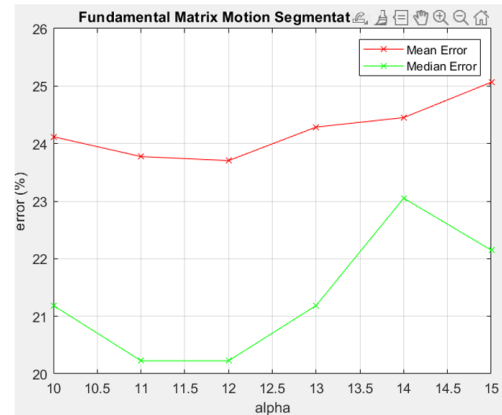


Figure 6. DBSCAN for Fundamental

In Figure 10 and Figure 12 we report the average errors for different  $\lambda$  values. We can say that as the  $\lambda$  increases, the average error decreases. This is mainly because as we increase the  $\lambda$ , we increase the penalty coefficient forcing the embeddings to approach each other.

In Figure 11 and Figure 13 we report the average errors

for different  $\gamma$  values for the subset method. The  $\gamma$  value has the opposite effect compared to the  $\lambda$ . We can see that as we decrease the  $\gamma$ , we decrease the average error as well. It is because we are reducing the effect of the constraint



Figure 7. Fuzzy C Mean Clustering for Affine

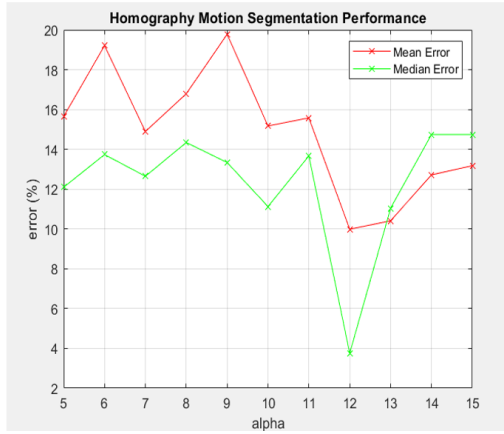


Figure 8. Fuzzy C Mean Clustering for Homography

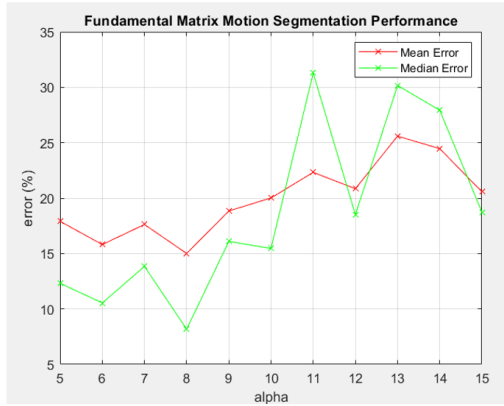


Figure 9. Fuzzy C Mean Clustering for Fundamental

and putting more emphasis on the first term, making the algorithm closer to a kernel addition.

Overall, we can say that for multi-view spectral clustering, kernel addition works the best for this dataset; whereas,

Models	K-means	Hierarchical	DBSCAN	Fuzzy C Mean
Affine	<b>15.45</b>	24.90	21.58	22.56
Fundamental	<b>20.19</b>	23.58	25.07	20.56
Homography	<b>14.26</b>	21.93	18.13	24.53
AHF.Subset	<b>9.45</b>	27.11	20.39	27.96
AHF.KerAdd	<b>8.78</b>	30.98	19.65	17.53
AHF.CoReg	<b>7.81</b>	28.60	20.17	24.60

Table 1. Overall Results of Different Classification Methods, best values are in bold. For Affine, Fundamental and Homography models, Average Mean Error (%) is reported. For AHF.Subset, AHF.KerAdd and AHF.CoReg, Overall Miss Classification Error (%) is reported.

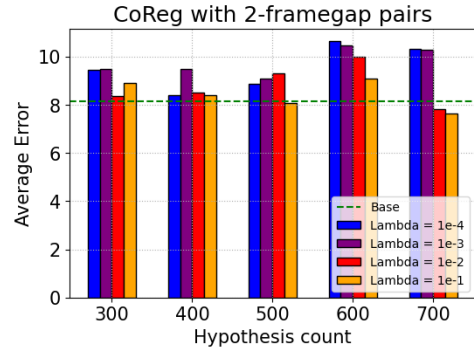


Figure 10. Effect of lambda for the co-regularization algorithm with 2-framegap pairs

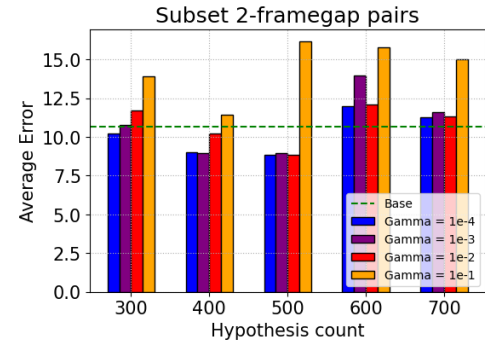


Figure 11. Effect of gamma for the subset algorithm with 2-framegap pairs

homography performs the best for the single-view spectral clustering.

### 4.3. Fitting Homography

The results for LMeds and MSAC is in Table 4.3. The average mean error for LMeds comes out is 15.39% which is slightly higher than the RANSAC implementation which was 14.26%. The average mean error for MSAC comes out is 10.12% which is better than the RANSAC implementation which was 14.26%. Hence MSAC can be good option

Model	Base		Number of Hypothesis Generated									
			300		400		500		600		700	
	Average	Median	Average	Median	Average	Median	Average	Median	Average	Median	Average	Median
Affine	18.85	17.3	17.7	16.7	<b>14.13</b>	14.34	<b>14.15</b>	<b>9.3</b>	15.7	15.3	15.3	13.7
Homography	12.9	10.2	<b>8.3</b>	4	8.7	<b>2.3</b>	10	4.3	14	12.7	10.7	6.7
Fundamental	15.2	10.8	12.7	6.3	<b>10</b>	<b>4.3</b>	12.3	6	11	5	11	4.7
KerAdd	8.5	1.9	<b>7.7</b>	<b>1</b>	9	2	8	<b>1</b>	9.3	2.7	9.3	2
CoReg	8.7	2.4	8.7	<b>1</b>	<b>8.3</b>	<b>1</b>	8.7	<b>1</b>	10	2	<b>8.3</b>	<b>1</b>
Subset	11	2.1	12.3	1.3	10.7	<b>1</b>	10	<b>1</b>	12.7	<b>1</b>	<b>9.7</b>	<b>1</b>

Table 2. Average and Median Errors (%) for 2-framegap Pairs. The best values are in bold, for base we are reporting the results for consecutive-pairs with hypothesis number = 500

Model	Base		Number of Hypothesis Generated									
			300		400		500		600		700	
	Average	Median	Average	Median	Average	Median	Average	Median	Average	Median	Average	Median
Affine	18.9	17.3	19	<b>16</b>	18	16.3	19	17.3	18.7	17.7	<b>17.7</b>	18.3
Homography	12.9	10.2	12.3	11	13.3	7.3	<b>10.3</b>	6.3	<b>10.3</b>	7.3	12.3	<b>6</b>
Fundamental	15.2	10.8	<b>12.3</b>	9.7	13.7	11.3	14.3	8.3	15.3	9	12.7	<b>7</b>
KerAdd	<b>8.5</b>	1.9	9.3	<b>1</b>	9.3	2.3	10	3.7	11	3.7	11.3	3.7
CoReg	<b>8.7</b>	2.4	9.3	3.3	9.3	2	<b>8.7</b>	<b>1.3</b>	<b>8.7</b>	2.3	10	2
Subset	<b>11</b>	2.1	12.3	5	12	4.7	13.3	<b>2</b>	13	2.3	13	<b>2</b>

Table 3. Average and Median Errors (%) for 3-framegap Pairs. The best values are in bold, for base we are reporting the results for consecutive-pairs with hypothesis number = 500

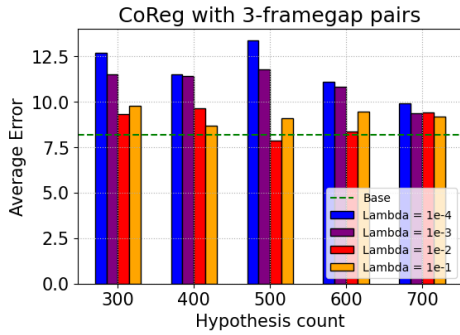


Figure 12. Effect of lambda for the co-regularization algorithm with 3-framegap pairs

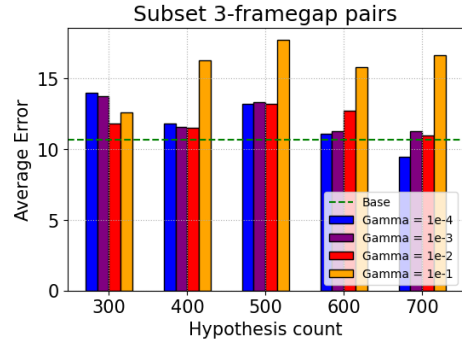


Figure 13. Effect of lambda for the subset algorithm with 3-framegap pairs

Models	RANSAC	MSAC	LMeds
Mean Error (%)	14.26	<b>10.12</b>	15.39

Table 4. Results for Homography Fitting

for the homography estimation.

## 5. Conclusion

In conclusion, in this paper, we propose improvements for the motion segmentation algorithm [19] to make it robust to outliers. We implemented and compared different clustering algorithms but the algorithms perform worse compared to K-means. In the non-consecutive pair exper-

iments, we did not see drastic improvements; however, 2-framegap pairs outperform the base model in every setting. Lastly, previously RANSAC was implemented to fit the homography model with a mean error 14.26%, we reduced this error by 4% with a mean error of 10.12% by using MSAC.

For future work, it is possible to study the outcomes of the combinations of these methods to see any improvement and to test on other possible clustering algorithms.

## References

- [1] F. Arrigoni and T. Pajdla. Robust motion segmentation from pairwise matches. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 671–681, 2019. [1](#)
- [2] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29:159–179, 1998. [1](#)
- [3] R. Dragon, B. Rosenhahn, and J. Ostermann. Multi-scale clustering of frame-to-frame correspondences for motion segmentation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part II 12*, pages 445–458. Springer, 2012. [1](#)
- [4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. [3](#)
- [5] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004. [1](#)
- [6] T. Lange and J. Buhmann. Fusion of similarity data in clustering. *Advances in neural information processing systems*, 18, 2005. [1](#)
- [7] F. Lauer and C. Schnörr. Spectral clustering of linear subspaces for motion segmentation. In *2009 IEEE 12th International Conference on Computer Vision*, pages 678–685. IEEE, 2009. [1](#)
- [8] Z. Li, J. Guo, L.-F. Cheong, and S. Z. Zhou. Perspective motion segmentation via collaborative clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1376, 2013. [1](#)
- [9] L. Magri, F. Andrea, et al. Robust multiple model fitting with preference analysis and low-rank approximation. In *Proceedings of the British Machine Vision Conference 2015*, pages 20–1, 2015. [3](#)
- [10] T. A. Mirko Wachter. Hierarchical segmentation of manipulation actions based on object relations and motion characteristics. *2015 International Conference on Advanced Robotics (ICAR)*, 6:1–8, 2015. [2](#)
- [11] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001. [1](#)
- [12] L. Peng, Y. Zhang, H. Zhou, and T. Lu. A robust method for estimating image geometry with local structure constraint. *IEEE Access*, 6:20734–20747, 2018. [3](#)
- [13] D. Szolgay, J. Benois-Pineau, R. M  gret, Y. Ga  stel, and J.-F. Dartigues. Detection of moving foreground objects in videos with strong camera motion. *Pattern Analysis and Applications*, 14:311–328, 2011. [2](#)
- [14] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9:137–154, 1992. [1](#)
- [15] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007. [3](#)
- [16] R. Vidal and R. Hartley. Motion segmentation with missing data using powerfactorization and gpca. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II. IEEE, 2004. [1](#)
- [17] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *IEEE transactions on pattern analysis and machine intelligence*, 27(12):1945–1959, 2005. [1](#)
- [18] U. B. Weijie Chen, Maryellen L. Giger. A fuzzy c-means (fcm)-based approach for computerized segmentation of breast lesions in dynamic contrast-enhanced mr images. *Academia Radiology*, 13:63–72, 2006. [2](#)
- [19] X. Xu, L. F. Cheong, and Z. Li. Motion segmentation by exploiting complementary geometric models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2859–2867, 2018. [1](#), [3](#), [6](#)
- [20] J. Zhang and V. Ila. Multi-frame motion segmentation for dynamic scene modelling. In *The 20th Australasian Conference on Robotics and Automation (ACRA)*. Australian Robotics & Automation Association, 2018. [1](#)