Sako and Sugaya *IPSJ Transactions on Computer Vision and Applications* (2016) 8:1
DOI 10.1186/s41074-016-0002-3

IPSJ Transactions on Computer Vision and Applications

EXPRESS PAPER Open Access

# Multibody motion segmentation for an arbitrary number of independent motions

Yutaro Sako and Yasuyuki Sugaya*

**Abstract**

We propose  a new method for segmenting feature point trajectories tracked through a video sequence without assuming a number of independent motions. Our method realizes motion segmentation of feature point trajectories by hierarchically separating the trajectories into two affine spaces in a situation that we do not know the number of independently moving objects. We judge that input trajectories should be separated by comparing the likelihoods computed from those trajectories before/after separation. We also consider integration of the resulting separated trajectories for avoiding too much segmentations. By using real video images, we confirmed the efficiency of our proposed method.

## 1 Introduction

Separating independently moving objects in a video sequence is one of the important tasks in computer vision applications. Costeira and Kanade [1] proposed a segmentation algorithm based on the shape interaction matrix. Sugaya and Kanatani [3] proposed a multi-stage learning strategy using multiple models. Yan and Pollefeys [8] proposed a new local subspace fitting scheme. Vidal et al. [7] proposed a segmentation algorithm based on generalized principal component analysis (GPCA) [6]. By introducing GPCA for computing an initial segmentation, Sugaya and Kanatani [4] improved the multi-stage learning.

However, all these methods assume the number of moving objects. Kanatani and Matsunaga [2] proposed a method for estimating the number of independently moving objects based on the rank estimation of the affine space using the geometric minimum description length (MDL). However, estimating the number of independently moving objects based on the rank of the affine space is very difficult for real image sequences. For example, if an object motion is planar, the dimension of an affine space which includes its trajectories degenerates from 3-D to 2-D. Moreover, if two objects merely translate without rotation, the two 2-D affine spaces are parallel to each other. This means that a 3-D affine space which includes those 2-D affine spaces exists.

For this problem, we propose a new method for segmenting feature point trajectories without assuming the number of objects. Based on the fact that trajectories of a rigidly moving object is constrained to a 2-D or 3-D affine space, we hierarchically separate input trajectories into two affine spaces until all the trajectories are divided into 2-D or 3-D affine spaces. In order to judge whether input trajectories should be divided or not, we compare the likelihoods before/after separation. After the separation process, we also check whether the separated trajectories should be integrated by comparing the likelihoods to avoid that the trajectories which belong to the same object are separated into different groups.

## 2 Proposed method

From the fact that trajectories of a rigidly moving object is constrained to a 2-D or 3-D affine space, we can separate independently moving objects by hierarchically separating input trajectories into two affine spaces until all the trajectories are divided into 2-D or 3-D affine spaces. In order to realize the above separation, we need to overcome two problems. One problem is to properly estimate the dimension of the affine space which includes input trajectories. The other problem is that we need to judge whether input trajectories should be divided to stop hierarchical separation.

*Correspondence: sugaya@iim.cs.tut.ac.jp
Toyohashi University of Technology, 1-1-1, Tempakucho, Hibarigaoka, Toyohashi, Aichi 441-8580, Japan

For the first problem, we can regard the rank of the moment matrix of the input trajectory vectors. The rank of the moment matrix can be obtained as the number of positive eigenvalues of the matrix. However, in the presence of noise, all eigenvalues are non-zero in general. Hence, we need to truncate small eigenvalues, but it is difficult to determine a proper threshold. We compute the rank of the moment matrix by using the geometric MDL [2].

For the second problem, we compare the average likelihoods of the trajectories for the affine spaces which are fitted to all the trajectories and the divided ones. We compute the average likelihoods before/after division and divide those trajectories if the likelihood after division is larger than that before division.

We summarize the algorithm of our proposed method as follows:

1. Fit an affine space to the input trajectories, and compute its dimension $d$ by using the geometric MDL.
2. If $d \leq 2$, then we stop the division process for the target trajectories.
3. Divide the trajectories into two affine spaces.
   (a) Convert the trajectory vectors into 3-D vectors.
   Please refer to [4] for the detail computation.
   (b) Fit two planes to those 3-D vectors by the Taubin method.
   (c) Convert the trajectory vectors into $d$-D vectors.
   (d) Separate the $d$-D vectors into two affine spaces by the EM algorithm of Sugaya and Kanatani [3, 4].
4. Compute the average likelihoods $P$ and $P'$ of the trajectories before/after separation and accept the separation, and go to step 1 if the following inequality is satisfied.

$$\lfloor \log_{10} P' \rfloor - \lfloor \log_{10} P \rfloor > 0, \tag{1}$$

   where $\lfloor \cdot \rfloor$ is the floor function. In our experience, if we compared the average likelihoods directly, the separation judgement was not stable. Thus, we compare the exponent part of the average likelihood.
5. Else, reject the separation.

We hierarchically iterate the above procedures until all the input trajectories are not separated.

## 3 Rank estimation of the affine space

We compute the eigenvalues of the moment matrix of the $2M$-D trajectory vectors $\boldsymbol{p}_\alpha$ [4], $\alpha = 1, \ldots, N$ and estimate its rank by using the geometric MDL. $M$ is the number of the image frame.

(1) Define the $2M \times 2M$ moment matrix $\boldsymbol{M}$ by

$$\boldsymbol{M} = \sum_{\alpha=1}^{N} (\boldsymbol{p}_\alpha - \boldsymbol{p}_C)(\boldsymbol{p}_\alpha - \boldsymbol{p}_C)^\top, \quad \boldsymbol{p}_C = \frac{1}{N}\sum_{\alpha=1}^{N} \boldsymbol{p}_\alpha. \tag{2}$$

(2) Compute the eigenvalues of $\boldsymbol{M}$, and let $\lambda_1 \geq \cdots \geq \lambda_{2M}$ be the sorted eigenvalues.

(3) Compute the residuals $J_r$, $r = 2, ..., 2M$, for the fitted $r$-D affine space by

$$J_r = \sum_{\beta=r+1}^{2M} \lambda_\beta. \tag{3}$$

(4) Compute the geometric MDLs [2] for each rank by

$$\text{G-MDL}(r) = J_r - \Big(rN + (r+1)(2M-r)\Big)\epsilon^2 \log\Big(\frac{\epsilon}{L}\Big)^2, \tag{4}$$

where $\epsilon$ is the standard deviation of feature point tracking accuracy, which we call this *noise level*, and $L$ is a reference length, for which we can use an arbitrary value whose order is approximately the same as the data, say the image size.

(5) Estimate the rank $\hat{r}$ of the affine space which includes the input trajectories as

$$\hat{r} = \arg\min_r \text{G-MDL}(r) \tag{5}$$

## 4 Separation of the trajectories

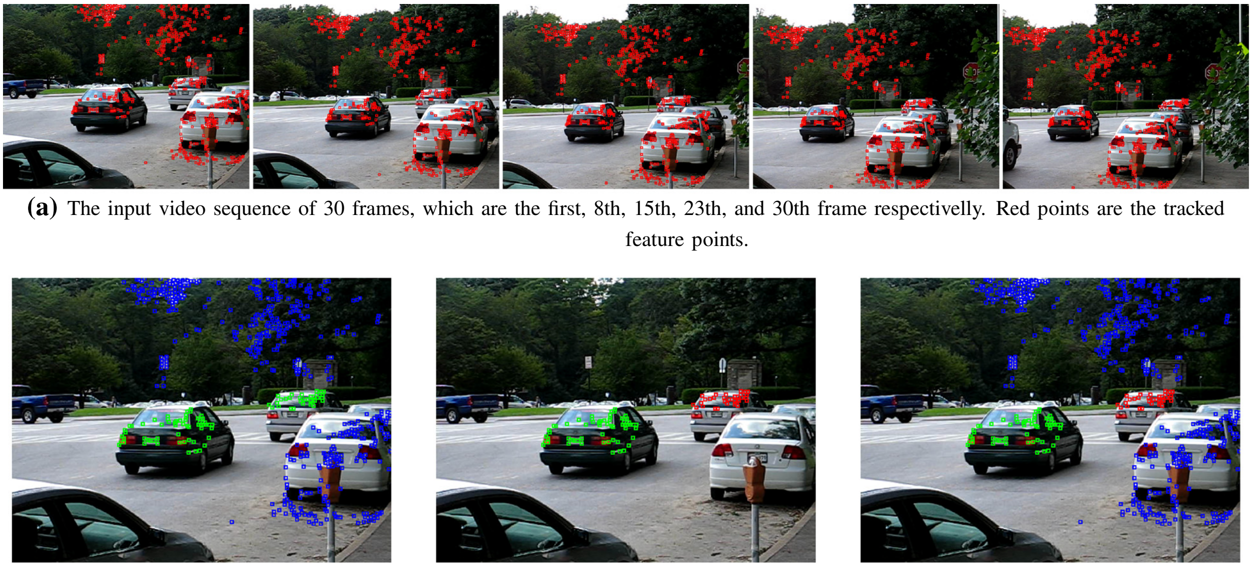
We separate the input trajectories by using the EM algorithm of Sugaya and Kanatani [3, 4] and can compute the likelihood $P(\alpha|k)$ of the $\alpha$-th point for the fitted affine space $k$ in the separation process. The likelihood $P(\alpha|k)$ can be computed in the form

$$P(\alpha|k) = \frac{e^{-(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)}, \boldsymbol{V}^{(k)-1}(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)}))/2}}{\sqrt{\det \boldsymbol{V}^{(k)}}}, \tag{6}$$

where $\boldsymbol{p}_C^{(k)}$ and $\boldsymbol{V}^{(k)}$ are the centroid and the covariance matrix of class $k$, respectively.

We compute the likelihoods $P(\alpha)$ and $P'(\alpha)$ of $\boldsymbol{p}_\alpha$ for the affine spaces before/after separation and then compute the average likelihoods $P$ and $P'$ by

$$P = \frac{1}{N}\sum_{\alpha=1}^{N} P(\alpha), \quad P' = \frac{1}{N}\sum_{\alpha=1}^{N} P'(\alpha). \tag{7}$$



**(a)** The input video sequence of 30 frames, which are the first, 8th, 15th, 23th, and 30th frame respectivelly. Red points are the tracked feature points.

**(b)** Separation result of the first stage.

**(c)** Separation result of the second stage.

**(d)** Final result.

**Fig. 1** Separation process of our method

## 5 Integration of the separated trajectories

After separation, we check whether the separated trajectories should be integrated by comparing the average likelihoods to avoid that the trajectories which belong to the same object are separated into different groups.

For all the separated trajectory groups, we integrate the two groups of the separated trajectories if the average likelihood $Q$ before integration and $Q'$ after integration satisfy the following inequality:

$$\lfloor \log_{10} Q' \rfloor - \lfloor \log_{10} Q \rfloor > 0, \quad (8)$$

where $Q$ and $Q'$ are computed from Eqs. (6) and (7) for each affine space.

## 6 Real image experiments

By using real video images, we tested our method and computed the separation accuracy with RANSAC and LSA of Yan and Pollefeys [8] and GPCA of Vidal et al. [7].

### 6.1 Separation process

Figure 1 shows a separation process of our method. Figure 1a shows the five decimated images of the input video sequence. The red points are the tracked feature points by the KLT tracker. We explain this separation process by showing tree expression of our separation in Fig. 2.

First, we estimated the dimension of the affine space which includes all the input trajectories and obtained that its dimension was 4. Since the resulting dimension was larger than 2, then we separated those input trajectories. We show the separation result in Fig. 1b. For this result, we computed the average likelihoods for the affine spaces before/after separation and obtained the results $5.63 \times 10^{-7}$ and $8.11 \times 10^{-5}$, respectively. Since these values satisfied the inequality in Eq. (1), we accepted the separation.

In the second stage, we estimated the dimensions of the affine spaces for the separated trajectories shown in Fig. 1b of green and blue points. The resulting dimension of the blue points was 2 and stopped separation. Since the estimated dimension of the green points was 3, we separated those trajectories. The result is shown in Fig. 1c. For this separation, the computed average likelihoods satisfied Eq. (1), and then, we accepted the separation. In the third stage, we estimated the dimensions of the affine spaces for the separated trajectories shown in Fig. 1c of red and green points. The resulting dimension of the red points

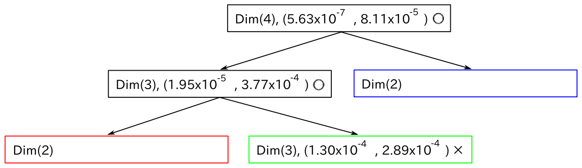

Dim(4), ($5.63\text{x}10^{-7}$ , $8.11\text{x}10^{-5}$ ) ○
Dim(3), ($1.95\text{x}10^{-5}$ , $3.77\text{x}10^{-4}$ ) ○
Dim(2)
Dim(2)
Dim(3), ($1.30\text{x}10^{-4}$ , $2.89\text{x}10^{-4}$ ) ×

**Fig. 2** Tree expression of the separation for Fig. 1. Dim (·) indicates the estimated dimension of the affine space which includes input trajectories. (·, ·) is paired likelihoods for the affine spaces before/after separation. The symbols *circle* and *multiplication* mean that the input trajectories should be separated or not. *Blue*, *red*, and *green rectangles* correspond to the *boxes* in Fig. 1d

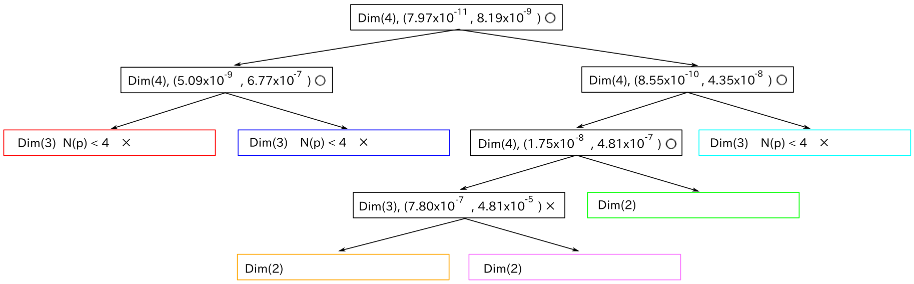

Dim(4), ($7.97 \times 10^{-11}$, $8.19 \times 10^{-9}$) ○
Dim(4), ($5.09 \times 10^{-9}$, $6.77 \times 10^{-7}$) ○
Dim(4), ($8.55 \times 10^{-10}$, $4.35 \times 10^{-8}$) ○
Dim(3) N(p) < 4 ×
Dim(3) N(p) < 4 ×
Dim(4), ($1.75 \times 10^{-8}$, $4.81 \times 10^{-7}$) ○
Dim(3) N(p) < 4 ×
Dim(3), ($7.80 \times 10^{-7}$, $4.81 \times 10^{-5}$) ×
Dim(2)
Dim(2)
Dim(2)

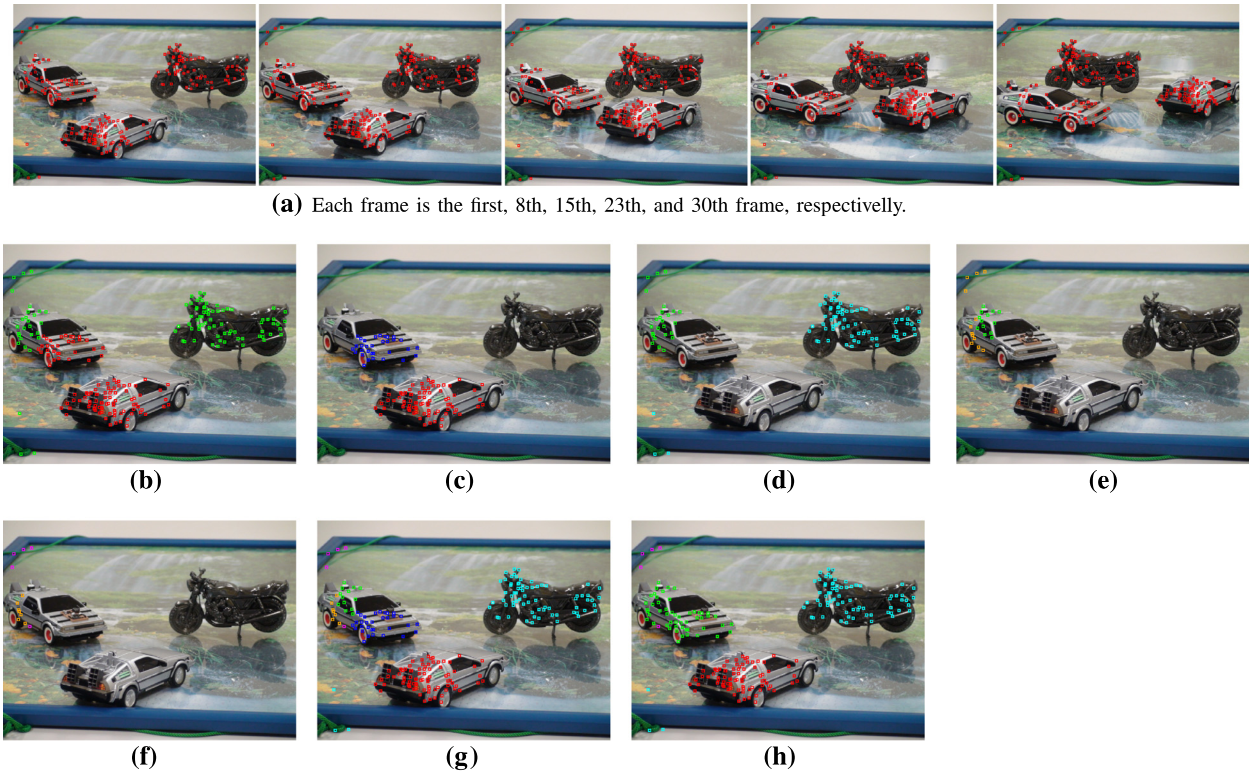
**Fig. 3** Tree expression of the separation for Fig. 4. The tree is created in the same manner of Fig. 2. $N(p)$ indicates the number of points in the separated affine space

was 2 and stopped separation. The estimated dimension of the green points was 3, but the average likelihoods before/after separation did not satisfy Eq. (1). Hence, we stop separation. We show the final separation result in Fig. 1d. In this sequence, the number of independently moving objects, which includes background points, is 3 and our method correctly separates the input trajectories into three groups.

### 6.2 Integrating process

We show another result in Fig. 4. Figure 4a shows the five decimated images of the input video sequence. We also show the separation process in Fig. 3 and show the results in each separation process in Fig. 4b–f. Figure 4g is the final separation result. From this result, we can see that miss-separation exists in Fig. 4e and the points which belong to the same object are separated into three groups, which are the blue, green, and orange points in Fig. 4g.

For all the pairs of the separated groups, we computed the average likelihoods before/after integration and check whether the separated groups should be integrated or not. Table 1 shows the computed average likelihoods. From this, the blue, green, and orange points are integrated. Figure 4g shows the integrated result.


**(a)** Each frame is the first, 8th, 15th, 23th, and 30th frame, respectivelly.

**(b)** **(c)** **(d)** **(e)**

**(f)** **(g)** **(h)**

**Fig. 4** Separation and integration process. **a** The input video sequence of 30 frames. *Red points* are the tracked feature points. **b**–**f** Results of the hierarchical separation process. **g** Separation result. **h** Integration result

**Table 1** Comparison of the likelihoods before/after integration of Fig. 4f. The top and middle rows in each cell show the average likelihoods before/after integration. The bottom row in each cell shows whether Eq. (8) is satisfied or not

| | Cyan | Green | Orange | Magenta |
|---|---|---|---|---|
| Red | $4.89 \times 10^{-7}$<br>$8.40 \times 10^{-10}$<br>× | $8.76 \times 10^{-7}$<br>$5.73 \times 10^{-9}$<br>× | $3.76 \times 10^{-8}$<br>$4.65 \times 10^{-9}$<br>× | $8.81 \times 10^{-7}$<br>$6.15 \times 10^{-9}$<br>× |
| Blue | $1.05 \times 10^{-7}$<br>$1.73 \times 10^{-9}$<br>× | $1.03 \times 10^{-10}$<br>$6.03 \times 10^{-9}$<br>○ | $3.12 \times 10^{-10}$<br>$5.52 \times 10^{-9}$<br>○ | $2.93 \times 10^{-7}$<br>$1.68 \times 10^{-8}$<br>× |
| Cyan | | $1.63 \times 10^{-6}$<br>$2.24 \times 10^{-9}$<br>× | $7.48 \times 10^{-7}$<br>$3.02 \times 10^{-9}$<br>× | $7.07 \times 10^{-8}$<br>$3.57 \times 10^{-9}$<br>× |
| Green | | | $1.52 \times 10^{-29}$<br>$6.58 \times 10^{-9}$<br>× | $4.26 \times 10^{-8}$<br>$9.15 \times 10^{-9}$<br>× |

### 6.3 Accuracy comparison

We compared the accuracy of our method with RANSAC and LSA of Yan and Pollefeys [8] and GPCA of Vidal et al. [7]. We used the T-Hopkins database [5] and original data and computed the separation accuracy for each method. Table 2 shows the result. The accuracy is computed by (number of correctly separated points)/(number of input points). As we can see, our method is superior to the compared existing methods.

## 7 Conclusions

In this paper, we propose a new method for segmenting feature point trajectories without assuming the number of objects. Based on the fact that trajectories of a rigidly moving object is constrained to a 2-D or 3-D affine space, we hierarchically separate input trajectories into two affine spaces until all the trajectories are divided into 2-D or 3-D affine spaces. In order to judge whether input trajectories should be divided or not, we compare the likelihoods before/after separation. After the separation process, we also check whether the separated trajectories should be integrated by comparing the likelihoods to avoid that the trajectories which belong to the same object are separated into different groups.

By using real video sequences, we checked the separation and integration processes of our method and confirmed the accuracy of our method by comparing existing methods.

**Table 2** Accuracy comparison

| | Our method (%) | RANSAC (%) | LSA (%) | GPCA (%) |
|---|---|---|---|---|
| Data1 | 100 | 92.50 | 93.43 | 89.31 |
| Data2 | 98.16 | 57.60 | 77.42 | 62.21 |
| Data3 | 100 | 84.82 | 96.43 | 81.25 |
| Data4 | 99.52 | 80.95 | 90.48 | 80.95 |
| Data5 | 100 | 80.30 | 92.75 | 80.30 |
| Data6 | 99.06 | 68.40 | 99.53 | 68.40 |
| Data7 | 100 | 45.49 | 96.99 | 45.49 |
| Data8 | 90.82 | 78.31 | 89.54 | Not converge |
| Data9 | 100 | 72.22 | 96.43 | 90.87 |
| Data10 | 100 | 91.03 | 93.40 | 92.08 |
| Data11 | 99.61 | 92.87 | 76.11 | 85.55 |
| Data12 | 99.63 | 85.56 | 94.44 | 80.00 |
| Data13 | 100 | 90.26 | 56.06 | 77.67 |

#### Competing interests

The authors declare that they have no competing interests.

#### Authors' contributions

YSa introduced the hierarchical separation of the feature point trajectories, carried out all the experiments, and wrote the manuscript. YSu introduced the rank estimation of the affine space using the geometric MDL and separation judgement using likelihoods of the fitted affine spaces and revised the manuscript written by YSa. Both authors read and approved the final manuscript.

Received: 5 May 2016 Accepted: 17 June 2016
Published online: 02 August 2016

#### References

1. Costeira JP, Kanade T (1998) A multibody factorization method for independently moving objects. Int J Comput Vision 29(3):159–179
2. Kanatani K, Matsunaga C (2002) Estimating the number of independent motions for multibody motion segmentation. In: Proc of the 5th Asian Conference on Computer Vision (ACCV2002), Melbourne, Australia. pp 7–12
3. Sugaya Y, Kanatani K (2004) Multi-stage unsupervised learning for multi-body motion segmentation. IEICE Trans Inform Syst E87-D(7):1935–1942
4. Sugaya Y, Kanatani K (2010) Improved multistage learning for multibody motion segmentation. In: Proc. of International Conference of Computer Vision Theory and Applications (VISAPP2010), Angers, France Vol. 1. pp 199–206
5. Sugaya Y KanataniK (2013) Removing mistracking of multibody motion video database Hopkins155. In: Proc. of the 24th British Machine Vision Conference (BMVC2013), Bristol, U.K.
6. Vidal R, Ma Y, Sastry S (2005) Generalized principal component analysis (GPCA). IEEE Trans Pattern Anal Mach Intell 27(12):1945–1959
7. Vidal R, Tron R, Hartley R (2008) Multiframe motion segmentation with missing data using PowerFactorization and GPCA. Int J Comput Vis 79(1):85–105
8. Yan J, Pollefeys M (2006) A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: Proc. of the 9th European Conference on Computer Vision (ECCV2006), Graz, Austria. pp 94–106