# Motion Segmentation from Feature Trajectories with Missing Data

Carme Julià, Angel Sappa, Felipe Lumbreras, Joan Serrat, and Antonio López

Computer Vision Center and Computer Science Department,
Universitat Autònoma de Barcelona,
08193 Bellaterra, Spain
{cjulia,asappa,felipe,joans,antonio}@cvc.uab.es

**Abstract.** This paper presents a novel approach for motion segmentation from feature trajectories with missing data. It consists of two stages. In the first stage, missing data are filled in by applying a factorization technique to the matrix of trajectories. Since the number of objects in the scene is not given and the rank of this matrix can not be directly computed, a simple technique for matrix rank estimation, based on a frequency spectra representation, is proposed. In the second stage, motion segmentation is obtained by using a clustering approach based on the normalized cuts criterion. Finally, the shape $S$ and motion $M$ of each of the obtained clusters (i.e., single objects) are recovered by applying classical SFM techniques. Experiments with synthetic and real data are provided in order to demonstrate the viability of the proposed approach.

## 1 Introduction

Several techniques have been proposed for the motion segmentation problem by feature trajectory grouping. Some of these approaches are formulated under the framework of factorization methods (e.g., [1,2,3], to mention a few). Features are tracked over time and their coordinates are stacked into a matrix $W_{2f\times p}$, where $f$ and $p$ are the numbers of frames and feature points respectively—referred to as matrix of trajectories hereinafter $W$. The key point is that under affine camera model, feature trajectories corresponding to the same object lie in the same linear subspace. Therefore the aim of the different proposed approaches is to find each of these linear subspaces in order to reduce $W$ to a form that allows an easy identification of them.

The aforementioned problem becomes more difficult when the matrix of trajectories contains missing data; that is, when not all the feature point trajectories are visible during the whole sequence (e.g., due to object occlusions, features missed by the tracker or new detected features) and no information of the objects in the scene nor the rank of $W$ are given. In this case two different problems should be faced up. Firstly, the unknown entries in the matrix of trajectories must be filled in. Secondly, once $W$ has been filled in, feature trajectories corresponding to the same object should be clustered together without previous

J. Martí et al. (Eds.): IbPRIA 2007, Part I, LNCS 4477, pp. 483–490, 2007.

**Table 1.** Summary of relevant features in previous techniques

| **method** | **data** | **rank value of** $W$ |
|---|---|---|
| Boult, Brown [1] | full | estimated(singular values) |
| Costeira, Kanade [2] | full | estimated (interaction matrix) |
| Han, Kanade [3] | full | estimated (maximum 6) |
| Kanatani [4] | full | estimated (model selection) |
| Zelnik-Manor, Irani [5] | full | estimated (singular values) |
| Yan, Pollefeys [6] | full | estimated (model selection[4]) |
| Vidal, Hartley [7] | missing | 5 |

knowledge of the number of objects. Although some approaches have been proposed for this second problem (e.g. [1,2,3,4,5,6]), as far as we know, the missing data case is only tackled in [7] by imposing a rank five for $W$. Table 1 summarizes the most relevant features of previous works, related to our current proposal.

The current work is focused on motion segmentation from feature trajectories that contain missing data. A robust approach to deal with the two problems mentioned above is presented. On a first stage, a strategy to fill in the matrix of trajectories is introduced. It uses a factorization technique by firstly estimating *the best* rank of $W$, when no prior information about the scene is given. Rank estimation is based on a novel *goodness* measurement, which considers not only the initial entries of $W$ but also the recovered missing ones. The hypothesis of the proposed goodness measurement is that the *frequency spectra* of the input matrix $W$ should be similar after recovering missing entries. On a second stage, an approach similar to the one presented in [6] is used to obtain the feature trajectory clusters. Once the segmentation is obtained, the shape $S$ and motion $M$ of each of the clusters can be recovered by using any SFM technique (e.g., [8]).

The paper is organized as follows. Section 2 presents the proposed approach for estimating the rank of the matrix $W$ and filling in its missing entries. Section 3 summarizes the procedure used for motion segmentation once the given matrix $W$ has been filled. Experimental results with synthetic and real sequences, testing different percentages of missing data and without a prior knowledge of the scene, are presented in section 4. Conclusions and future work are given in section 5.

## 2 Fill in Process

The main objective at this stage is to fill in the matrix $W$ in order to proceed with its corresponding segmentation. Factorization techniques have been widely used to tackle this problem in the single object case. The central idea is to express a matrix $W$ as the product of two unknown matrices: $W = AB$. Hence, our motivation is to extend to the multiple object case this factorization-based strategy for filling in the matrix $W$—concretely, we will use as factorization the *Alternation* technique [9], which deals with missing data in $W$. In the case of a single rigid moving object, the rank of $W$ is at most four. Therefore, in general $W$ is filled, assuming that its rank is $r = 4$, by minimizing $\|W_{2f\times p} - A_{2f\times r}B_{r\times p}\|_F^2$,

where $\|.\|_F$ is the Frobenius norm [10]. Unfortunately, with multiple objects the rank is neither bounded nor easy to estimate, since information about the number of objects or about their motions is not given. Our strategy consists in applying the *Alternation* by assuming different rank values $r_0^k$ for $W$, obtaining, thus, a *filled* matrix $W_{fill}^k$ for each case. Then, the goodness of these filled matrices is studied and *the best* one is taken for the next stage.

Although different goodness measurements could be defined, it could be noticed that both known and missing entries of $W$ should be equally considered in order to obtain a fair value. For instance, selecting the rank that corresponds to the filled matrix with the minimum $rms$[1] could be wrong, since as it is pointed out in [11], no goodness measurement of recovered data is used. In this context we propose a novel goodness measurement detailed below.

The philosophy of the proposed approach consists in studying the *frequency spectra* of the input matrix $W$. The hypothesis of the goodness measurement is that, since feature point trajectories belong to surfaces of rigid objects, the behaviour of the missing data should be similar to the visible one. This similar behaviour is identified with the fact that the computed matrices $W_{fill}^k$ and the input one $W$ have a similar *frequency* content. In order to do that, the *Fast Fourier Transform* (FFT) is applied to each of the columns of the matrices $W_{fill}^k$ and also to the columns of $W$ (adding zeros to its missing entries) for comparing their modulus. Since the idea is to group features according to their motion, the columns of the matrices are taken instead of considering the rows or the two dimensions at the same time. In summary, the strategy is the following:

1. Take different rank values for $W$: $r_0^k$, where $k = [5, ..., 15]$ in our experiments.
2. For each $r_0^k$, apply the *Alternation* technique to fill in the matrix of trajectories, obtaining a $W_{fill}^k$ for each one.
3. Apply the FFT to $W$ and to each $W_{fill}^k$ and compute their modulus: $F_0 = |FFT(W)|$, $F_k = |FFT(W_{fill}^k)|$
4. Choose the $W_{fill}^k$ (referred to as $W_{fill}$ hereinafter) for which the following expression is minimum: $\|F_0 - F_k\|_F = \sqrt{\sum_{i,j} ((F_0)_{ij} - (F_k)_{ij})^2}$

## 3 Motion Segmentation

In this second stage, a similar approach to the one proposed in [6] is used to segment the trajectories. It consists in estimating a local subspace for each feature trajectory, and then compute an affinity matrix based on principal angles between each pair of these estimated subspaces. Finally, the segmentation of the feature trajectories is obtained by applying spectral clustering [12] to this affinity matrix, using the *normalized cut criterion* [13]. The steps of the algorithm are briefly described below.

**Rank detection**. In the first step of the algorithm, the rank of the filled matrix $W_{fill}$ is computed. In general, in presence of noise all singular values are nonzero.

[1] $rms^k = \|W - W_{fill}^k\|_F / \sqrt{\frac{q}{2}}$, where $q$ is the number of known entries in $W$.

Therefore, the smallest ones must be truncated in order to estimate the rank. However, it is difficult to set an appropriate threshold. In [14], authors propose the *model selection* for rank detection. Based on that, the following expression is used to estimate the rank in presence of noise:

$$r_m = argmin_r \frac{\lambda_{r+1}^2}{\sum_{j=1}^{r} \lambda_j^2} + \mu r, \tag{1}$$

where $\lambda_i$ corresponds to the $i$-th singular value of the matrix, and $\mu$ is a parameter that depends on the amount of noise. The higher the noise level is, the larger $\mu$ should be (in our experiments, $\mu = 10^{-7}$). Therefore, the $r$ that minimizes this expression is considered as the rank of $W_{fill}$. Notice that it does not have to coincide with the rank value used in the previous stage to fill in the matrix of trajectories, $r_0^k$. In most of the cases, error is added to the entries of $W$ in the previous stage, hence its rank could vary.

**Data transformation**. If $W_{fill}$ is a $2f \times p$ matrix, the idea is to consider each of its $p$ columns as a vector in $\mathbf{R}^{2f}$ and to project them onto the unit sphere in $\mathbf{R}^r$, being $r$ the estimated rank value in the previous step. The SVD decomposes the matrix of trajectories as $W_{fill} = U_{2f\times 2f} S_{2f\times p} V_{p\times p}^t$. In order to project the trajectories onto $\mathbf{R}^r$, only the first $r$ rows of $V^t$ are considered: $V_{r\times p}^t$. Finally, the $p$ columns of this matrix are normalized to project them onto the unit sphere.

**Subspace estimation.** For each point $\alpha$ in the transformed space, its local subspace is computed, formed by itself and its $n$ closest neighbours: $[\alpha, \alpha_1, \cdots, \alpha_n]$, being $n + 1 = d$; where $d$ is the highest dimension of the linear subspaces generated by each cluster (e.g., 4 for the rigid object case). The closest neighbours are selected using the Euclidean distance between the transformed points.

**Affinity matrix**. Instead of computing a distance between points, the distance between the local subspaces estimated in the previous step is used, which is measured by principal angles [10]. The affinity $A$ of two points $\alpha$ and $\beta$ is defined as the distance between their estimated local subspaces $S(\alpha)$ and $S(\beta)$:

$$A(\alpha, \beta) = e^{-\sum_{i=1}^{M} sin(\theta_i)^2}, \tag{2}$$

where $\theta_i$ is the $i$-th principal angle between the subspaces $S(\alpha)$ and $S(\beta)$ and $M$ the minimum of their dimensions.

**Spectral clustering**. Finally, the motion segmentation is obtained by applying spectral clustering [12] to the affinity matrix computed in the previous step. Concretely, the *normalized cut criterion*, presented in [13], is used to segment the data. This criterion measures both the total dissimilarity between the different clusters as well as the total similarity within the clusters and it can be optimized by using a technique based on a generalized eigenvalue problem.

## 4 Evaluation Study

In this section, the performance of the proposed approach is studied by using synthetic and real data. Actually, only the 2–objects case is studied in this paper.

Considering different percentages of missing data, 25 attempts are repeated and the percentage of bad-clustered features over the total of features in $W$ is computed. This gives a measure of error in the clustering. Finally, the mean and median of errors in all the attempts are presented.

Given a full matrix, the missing data are generated by automatically removing parts of random columns in order to simulate the behaviour of tracked features. Non-filled columns correspond to features missed by the tracker or to new features detected after the first frame.

### 4.1 Synthetic Data

Two different objects are used. The first one is generated by randomly distributing 3D feature points over the surface of a cylinder, see Fig 1 (left). The second object is generated from a set of 3D points, which correspond to a Beethoven sculptured surface represented by a triangular mesh, see Fig 1 (middle).

Taking these two 3D objects, different sequences are obtained by performing rotations and translations over both of them. At the same time, the camera also rotates and translates. Although self-occlusions are produced, all the points are stacked into the matrix of trajectories, since this is a synthetic experiment. The full obtained trajectories of a sequence with a cylinder and a Beethoven are shown in Fig 1 (right). This sequence is defined by 50 frames containing 451 features (185 from the cylinder and 266 from the Beethoven sculpture).

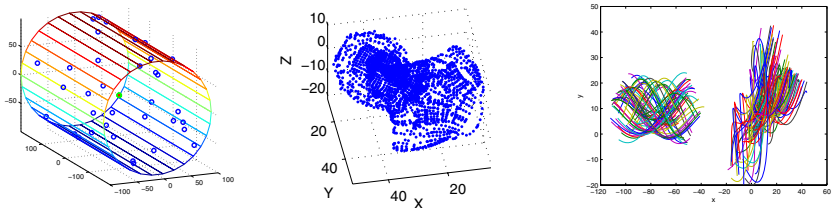


**Fig. 1.** Synthetic objects: (left) Cylinder. (middle) Beethoven. (right) Full feature trajectories of the second sequence in table 2, plotted in the image plane.

Table 2 presents the mean and median of the error obtained in the 25 attempts for each sequence and each percentage of missing data. In the first two sequences the objects move independently, while in the last one the rotation of both objects is identical and consequently the motion is dependent. Independently of the object motion's dependency, good results are obtained as long as the percentage of missing data is below 50%.

Although out of the scope of this work, since our main target is motion segmentation, Fig 2 shows an illustration of the shape and motion of each object recovered by applying the *Alternation* to the results of the proposed technique. These results correspond to the second sequence in Table 2, 20% of missing data.

**Table 2.** Synthetic experiments

| **sequence** | **2 cyl.**, $W_{180\times145}$ | | | | **cyl., Beet.**, $W_{100\times451}$ | | | | **2 cyl.**, $W_{180\times160}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **missing data** | 20% | 30% | 40% | 50% | 20% | 30% | 40% | 50% | 20% | 30% | 40% | 50% |
| **mean error** | 0.02 | 0 | 0.57 | 15.14 | 0.07 | 0.41 | 0.66 | 4.98 | 0.25 | 0.32 | 0.20 | 15.90 |
| **median error** | 0 | 0 | 0 | 2.75 | 0 | 0 | 0 | 0.66 | 0 | 0 | 0 | 10.62 |

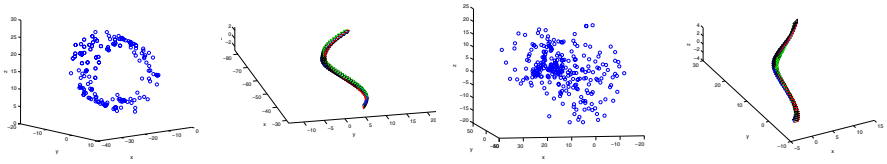

**Fig. 2.** Recovered 3D shape and motion: (left) Cylinder. (right) Beethoven.

### 4.2 Real Data

The same procedure applied to the synthetic data is now used with real data. The two objects studied for these real data experiments are shown in Fig 3 (left) and (middle), respectively. For each object, a real video sequence with a resolution of 640 × 480 pixels is used. A single rotation around a vertical axis is performed to each of the objects. Feature points are selected by means of a corner detector algorithm and only points distributed over the squared-surfaces (box and cylinders) visible in all the frames are considered. More details about corner detection and tracking algorithm can be found in [15].

The input matrices of trajectories corresponding to sequences of multiple objects are generated by merging different matrices of trajectories (corresponding or not to the same object) after having interchanged the $x$ and $y$ coordinates. Overlapping between objects is avoided for clarity by applying a translation. The first studied sequence is generated by using the first object twice, while the second sequence uses both objects. The obtained full trajectories in this second case are plotted in Fig 3 (right). This sequence is defined by 61 frames and 275 features (87 from the first object and 188 from the second one).

Table 3 summarizes the obtained results. It can be seen that the error in the clustering is higher than in the synthetic case, even working with smaller percentages of missing data. The main reason is that, working with real noisy

**Fig. 3.** Real objects: (left) First object. (middle) Second object. (right) Full feature trajectories plotted in the image plane, second sequence in table 3.

**Table 3.** Real experiments

| **sequence** | **First sequence**, $W_{202\times174}$ | | | | | **Second sequence**, $W_{122\times275}$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **missing data** | 10% | 20% | 30% | 40% | 50% | 10% | 20% | 30% | 40% | 50% |
| **mean error** | 5.19 | 8.00 | 14.52 | 15.10 | 24.06 | 11.95 | 8.21 | 16.49 | 23.52 | 43.02 |
| **median error** | 1.72 | 3.44 | 5.74 | 5.17 | 20.68 | 3.63 | 4.00 | 7.63 | 29.09 | 46.54 |

data, the *Alternation* propagates the noise to the filled matrices in the filling in process. Consequently, the error in the clustering is higher than working with synthetic free of noise data.

Finally, Fig 4 shows an example of the recovered shape and motion obtained by applying *Alternation* to the trajectories corresponding to the two objects of the second sequence in Table 3, 20% of missing data.

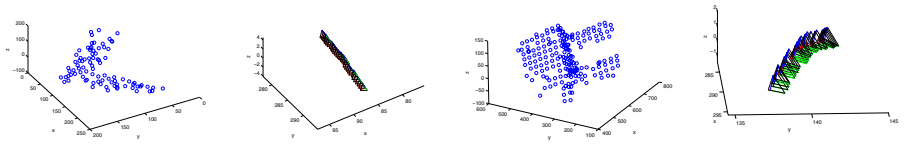

**Fig. 4.** Recovered shape and motion: (left) First object. (right) Second object.

## 5 Conclusions and Future Work

In this paper, an approach for motion segmentation from feature trajectories with missing data is presented. It consists of two stages. In the first stage, the missing data in the feature trajectories are filled in. Since working with missing data and with no prior knowledge of the number of objects in the scene, the rank of the matrix of trajectories can not be directly computed, a novel technique to estimate it is proposed. It is based on a *frequency spectra* study of $W$ and motivated by the fact that feature point trajectories belong to surfaces of rigid objects. Therefore the filled matrices should contain a *frequency spectra* similar to the one of the input matrix. In the second stage, motion segmentation is obtained by using a clustering technique based on the *normalized cut criterion*.

Although we focus our work on the study of the error in the clustering, it should be mentioned that, in the first stage, the rank of the input matrix $W$ is properly estimated in most of the cases by using the proposed goodness measurement (it can be checked, since the full initial matrices are known).

Experiments with independent and dependent motions are presented and it is shown that, although the approach performs well in both cases, better results are obtained when the motion subspaces are independent. In the experiments with real data, the error in the clustering is higher than in the synthetic ones. This is due to the added error in the feature trajectories during the first stage.

Further work will include a study of robustness of the proposed approach to noisy data.

**Acknowledgments.** This work has been supported by the Government of Spain under the MEC project TRA2004 - 06702/AUT. The second author has been supported by The Ramón y Cajal Program.

## References

1. Boult, T., Brown, L.: Factorization-based segmentation of motions. In: IEEE Workshop on Motion Understanding. pp. 179–186 (1991)
2. Costeira, J., Kanade, T.: A multibody factorization method for independently moving objects. International Journal of Computer Vision pp. 159–179 (1998)
3. Han, M., Kanade, T.: Reconstruction of a scene with multiple linearly moving objects. International Journal of Computer Vision 53, 285–300 (2000)
4. Kanatani, K.: Motion segmentation by subspace separation and model selection. In: CVPR. vol. 2, pp. 586–591 (2001)
5. Zelnik-Manor, L., Irani, M.: Degeneracies, dependencies and their implications in multi-body and multi-sequence factorization. In: CVPR. pp. 287–293 (2003)
6. Yan, J., Pollefeys, M.: A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: ECCV06 (2006)
7. Vidal, R., Hartley, R.: Motion segmentation with missing data using powerfactorization and GPCA. In: CVPR (2004)
8. Tomasi, C., Kanade, T.: Shape and motion from image streams: a factorization method. Full report on the orthographic case (1992)
9. Buchanan, A., Fitzgibbon, A.: Damped newton algorithms for matrix factorization with missing data. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2, 316–322 (2005)
10. Golub, G., Van Loan, C. (eds.): Matrix Computations. The Johns Hopkins Univ. Press, Baltimore, MD (1989)
11. Chen, P., Suter, D.: Recovering the missing components in a large noisy low-rank matrix: Application to SFM. IEEE Transactions on PAMI vol. 26 (2004)
12. Weiss, Y.: Segmentation using eigenvectors: a unifying view. In: International Conference on Computer Vision (1999)
13. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on PAMI (2000)
14. Kanatani, K.: Statistical optimization and geometric inference in computer vision. Philosophical transactions: Mathematical, physical and engineering sciences 356, 1303–1320 (1998)
15. Ma, Y., Soatto, J., Kosecká, J., Sastry, S.: An invitation to 3D vision: From images to geometric models. Springer, Heidelberg (2004)