Computer Vision and Image Understanding 117 (2013) 113–129

Contents lists available at SciVerse ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Joint estimation of segmentation and structure from motion

Luca Zappella [a,*], Alessio Del Bue [b], Xavier Lladó [c], Joaquim Salvi [c]

[a] Center for Imaging Science, Johns Hopkins University, 3400 North Charles Street Baltimore, MD 21218 USA
[b] Pattern Analysis and Computer Vision (PAVIS), Istituto Italiano di Tecnologia, Italy
[c] Architecture and Technology, University of Girona, 17071 Girona, Spain

ARTICLE INFO

Article history:
Received 19 May 2011
Accepted 13 September 2012
Available online 27 October 2012

Keywords:
Structure from motion
Multi-body structure from motion
Motion segmentation
Sparsity

ABSTRACT

We present a novel optimisation framework for the estimation of the multi-body motion segmentation and 3D reconstruction of a set of point trajectories in the presence of missing data. The proposed solution not only assigns the trajectories to the correct motion but it also solves for the 3D location of multi-body shape and it fills the missing entries in the measurement matrix. Such a solution is based on two fundamental principles: each of the multi-body motions is controlled by a set of metric constraints that are given by the specific camera model, and the shape matrix that describes the multi-body 3D shape is generally sparse. We jointly include such constraints in a unique optimisation framework which, starting from an initial segmentation, iteratively enforces these set of constraints in three stages. First, metric constraints are used to estimate the 3D metric shape and to fill the missing entries according to an orthographic camera model. Then, wrongly segmented trajectories are detected by using sparse optimisation of the shape matrix. A final reclassification strategy assigns the detected points to the right motion or discards them as outliers. We provide experiments that show consistent improvements to previous approaches both on synthetic and real data.

1. Introduction

The inference of the 3D position of freely moving objects is one of the most important tasks in computer vision. In complex scenarios, where several bodies rigidly move, it is first necessary to classify the motion of different objects before performing any other reconstruction task. In particular, *Motion Segmentation* (MS) from feature trajectories consists of segmenting the trajectories generated by points that belong to the surface of moving objects. MS is a low-level task and it is a fundamental step for other higher level tasks such as 3D reconstruction. In particular, uncalibrated *Structure from Motion* (SfM) is often required for several applications. Given one object that moves throughout a video sequence and given its 2D tracked features, the aim of SfM is to recover both the 3D coordinates of the points (up to a scale factor), and the motion description of the whole structure for each frame (up to an arbitrary initial rotation). The input of SfM algorithms is a trajectory matrix $\mathbf{W}_{2F\times P}$, where F is the number of frames of the sequence and P is the number of tracked points. The trajectory matrix contains for each frame the position in 2D of each of the tracked points. The result of SfM algorithms is the motion matrix $\mathbf{M}$, which describes rotation and translation of the tracked points in every frame, and the shape matrix $\mathbf{S}$, which contains the 3D position of each of the P features, such that $\mathbf{W} = \mathbf{MS}$ [1]. Fig. 1 provides an example of SfM: given different views of the same book, a SfM algorithm can recover the 3D structure of the book and the position of the camera for each of the views.

Very often SfM algorithms assume that only the trajectories of one single object are stored in $\mathbf{W}$. However, most of the time the feature tracker algorithm provides trajectories that belong to different objects with different motions. Hence, in order to apply a canonical single-body SfM algorithm, the MS problem should be solved first, so that trajectories that follow the same motion (i.e. trajectories that belong to the same object) are grouped together. As an example, Fig. 2 shows how MS can be used as a pre-processing step for SfM. If the segmentation is correct, SfM can recover the 3D structure and the motion of the objects, however, SfM requires that no outliers are present in the trajectory matrix of the segmented objects. This means that the MS algorithm has to provide a perfect segmentation in order to obtain an accurate 3D reconstruction. In Fig. 3 one of the trajectories of $\mathbf{W}_2$ is wrongly classified as belonging to $\mathbf{W}_3$, this mistake could likely lead to poor reconstructions of the 3D shape generated from $\mathbf{W}_3$. On the other hand, the motion description $\mathbf{M}_3$ should still be reliable in spite of the error, in fact the motion is influenced by the all of the points in $\mathbf{W}_3$, and if the majority of them are correctly classified the influence of one error plays a minor role.

In recent years, MS field has experienced a fast advance. The results of such a progress are new algorithms with very low misclassification rates. Nevertheless, SfM requires no error at all, as

* Corresponding author. Tel.: +1 410 516 6736; fax: +1 410 516 4557.
E-mail address: zappella@cis.jhu.edu (L. Zappella).

1077-3142/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2012.09.004

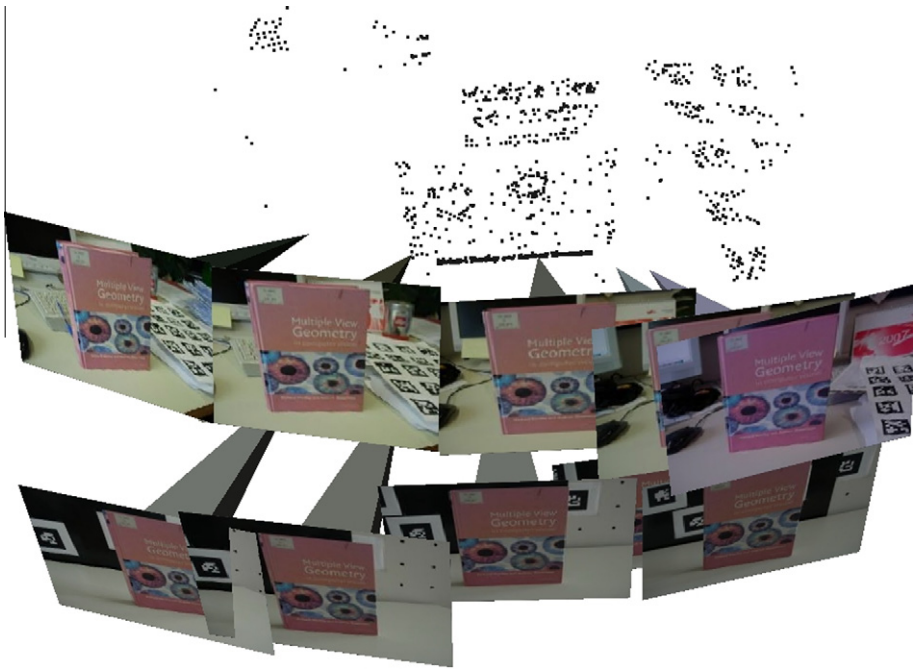


Fig. 1. Example of SfM: different views of the same object are taken and the position of each point is stored in a trajectory matrix $\mathbf{W}$, then a SfM algorithm can recover the 3D structure of the object, and the position of the camera for each view by factorizing $\mathbf{W}$ into the structure matrix $\mathbf{S}$ and the motion matrix $\mathbf{M}$ such that $\mathbf{W} = \mathbf{M}\,\mathbf{S}$ (figure taken from [2]).

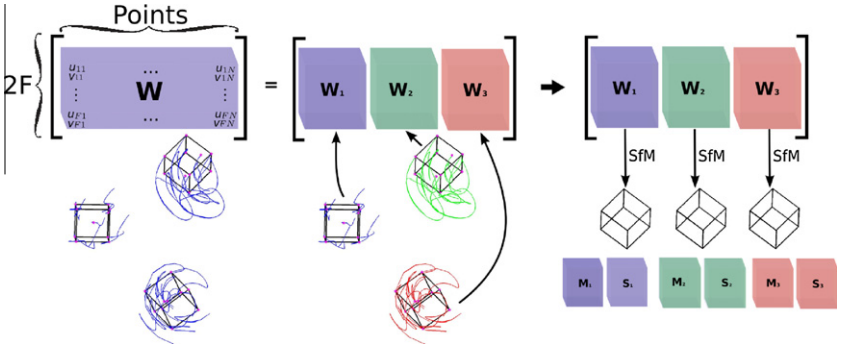


Fig. 2. An example of motion segmentation as a pre-processing step of SfM. When the trajectory matrix $\mathbf{W}$ is correctly segmented into the trajectory matrices $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$, then the 3D reconstruction is perfect (given that the motions are not degenerate).

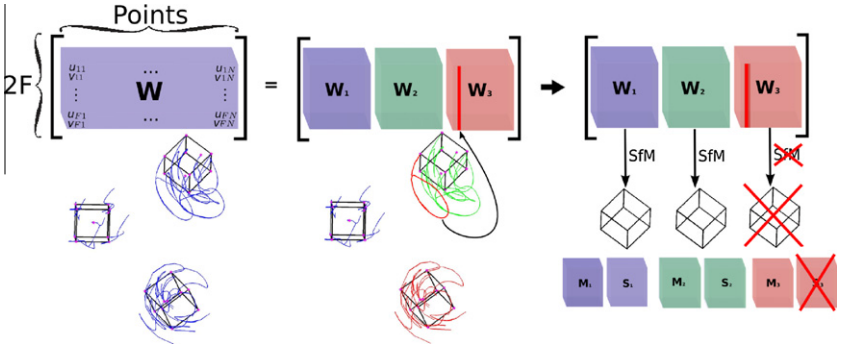


Fig. 3. When the segmentation result has even only one single error, like the trajectory of $\mathbf{W}_2$ wrongly associated to $\mathbf{W}_3$, the 3D reconstruction $\mathbf{S}_3$ may not be correct. On the other hand, the motion description $\mathbf{M}_3$ is still reliable as it is influenced by all of the points of $\mathbf{W}_3$ and only one single error plays a minor role.

shown in the previous example. At the same time, single-body SfM has also progressed rapidly, while multi-body SfM has been proved to be a much more challenging task [3]. Given the scenario just described, it seems sensible to think about an algorithm able to solve the multi-body SfM by exploiting good, but not yet perfect, MS solutions, and robust single-body SfM algorithms. The

proposed approach, Joint Estimation of Segmentation and Structure from motion (JESS), is a generic framework that can be applied to correct the initial result of *any* MS technique. JESS is an optimisation algorithm which, not only assigns the trajectories to the correct motion (MS), but it also solves for the 3D location of multi-body shape (SfM) and fills the missing entries, caused by occlusions or failure of the tracking system, of the trajectory matrix. Such a solution is based on two fundamental principles widely used in SfM but never applied, so far, to MS: the multi-body motion is subject to a set of *metric constraints* given by the specific camera model, and the shape matrix that describes the 3D shape is generally *sparse*. JESS iteratively enforces these constraints in three stages. First, given an initial segmentation, metric constraints are used to estimate the 3D metric shape and to fill the missing entries according to a scaled orthographic camera model. Then, wrong segmentations are detected using sparse optimisation of the multi-body shape matrix. A final reclassification strategy assigns the detected trajectories to the right motion or discards them as outliers.

Being JESS an iterative algorithm that exploits SfM constraints it is subject to few assumptions. The first assumption is that the motion of the different bodies is independent. The second assumption is that the initialisation of the iterative procedure (i.e. the initial amount of misclassified points) is not far from the correct solution. We have empirically shown that JESS can perform well when the initial MS error is not higher than 10% of the total number of points. This assumption should not be restrictive as it has been shown that current MS algorithms, such as [5,6,8,24] to mention a few, have performances with average misclassification rates that are below 5%. Nevertheless, we report in this paper some cases with a higher initial misclassification rate and we observe that JESS could reduce the misclassification rate also when the initial error ranges between 10% and 40%.

We present different synthetic experiments in order to evaluate the performance of JESS in terms of segmentation and 3D reconstruction when using: different ratios of noise, different levels of misclassified points in the initial segmentation, and different levels of missing data. An exhaustive evaluation on the real sequences of the Hopkins155 database [4] and on 12 additional sequences presented by the same authors in [5] is also included, comparing favourably our performance with two of the best state of the art MS methods dealing with missing data [6,7]. On both synthetic and real databases we show that JESS outperforms a RANSAC-based [9] algorithm, showing also that the problem is not trivial. The performance of JESS in the case of non-missing data are also shown when the initial MS is provided by one of the best performing algorithms [8] on the Hopkins155 database. Finally, JESS is also tested on a real sequence that contains two known datasets traditionally used for SfM evaluation. Differently from the Hopkins155 database, this last sequence contains longer rotational and translational motions. JESS Matlab source code is publicly available at http://eia.udg.edu/~zappella. This paper is a major extension of a preliminary work presented in [10].

2. Previous works

Ever since the first single-body SfM algorithm, proposed by Tomasi and Kanade [1], numerous improvements have been suggested in order to deal with rigid, articulated and also non-rigid objects [11–13]. Furthermore, other methods have been proposed in order to deal with missing data in the original 2D feature trajectories [14,17].

Several attempts have been made also to directly solve the multi-body SfM problem. Most of them tried to compute the MS solution intrinsically, by exploiting epipolar geometry and mixing algebraic and statistical tools [3,18–20]. The main limitation of these methods is the sensitivity to noise and outliers. Moreover, to the authors knowledge, the works presented in [3,21] are the only multi-body reconstruction approaches that are able to deal with missing data. The main idea of [21] is to enforce two-view constraints between consecutive frames and to use a model selection strategy to perform the segmentation. However, this strategy can lead to under and over segmentation results if the model selection is not properly fed with the right candidates. As stated in [3] a practical multi-body SfM algorithm which can handle realistic sequences is still missing. The authors of [3] tried to solve the problems that affect [21] by a post-processing step that employs a splitting and merging strategy. As the authors of [3] further discuss, there are still several open issues which need to be addressed. In particular they need to manually adjust the parameters of their algorithm in order to deal with different conditions (light changes, camera speed, etc.). To summarise, while the effort to solve the single-body SfM problem has led to robust algorithms able to deal with a variety of conditions, the multi-body SfM remains an open problem.

Meanwhile, research in the MS field has also advanced significantly. Different strategies have been used to tackle MS as described in [22]: image difference, statistics, wavelets, optical flow, layers and manifold clustering to cite a few. Recently, the Hopkins155 database [4] has become a standard benchmark for the evaluation of MS techniques. A few algorithms [7,22–24,8] reported low misclassification rates on the Hopkins155 database, which testifies the progresses of MS algorithms. However, most of the MS methods assume that feature trajectories are visible throughout the whole sequence and they do not deal with outliers introduced by a wrong association of the tracker. Recently, some approaches [5,6,25,26] have also tackled this issue by providing promising results.

The large amount of successful single-body SfM algorithms, the robustness reached by MS approaches and the weaknesses of the multi-body SfM frameworks have tightened the relationship between MS and SfM. If the multi-body SfM problem is to be solved, a successful MS algorithm has to be applied as a preprocessing step in order to feed the single-body SfM algorithms with the trajectories of one object at a time. So far, this is how MS and SfM have been related to each other. However, SfM theory provides some constraints that MS algorithms have never exploited. Specifically, the *metric constraints* that have to be satisfied by each estimated motion, and the fact that the shape matrix that describes the 3D shape of the multi-body case is *sparse*. The aim of this work is to bring the MS and SfM problems closer by trying to use SfM constraints in order to solve the MS problem. We have developed an iterative bilinear optimisation strategy that, using the SfM constraints, corrects an initial (and possibly erroneous) solution given by any MS algorithm. Furthermore, our algorithm achieves a 3D multi-body reconstruction and it fills the missing entries according to an orthographic camera model. These constraints are particularly effective in the presence of missing data, since metric constraints are the key to obtain effective matrix completion of the 2D trajectories as demonstrated in [17]. Hence, an initial segmentation is exploited to solve the multi-body SfM problem, which, in turn, provides unexploited constraints to correct the segmentation. Once a stop condition is verified, a reclassification strategy can take place in order to reclassify the points identified as MS errors. JESS contributes towards the challenging direction of merging the problems of MS and SfM.

In the following section the theory for the single and multi-body SfM cases are developed. During this analysis the two constraints used in JESS, the metric constraints and the sparsity of the multi-body shape matrix, will be highlighted.

3. SfM with missing data

The bilinear SfM problem with missing data is introduced now for the single object case. The solution to this problem provides the metric constraints given by an orthographic camera model. This formulation will then be extended to the multi-body case, and the sparsity constraint will be shown explicitly.

3.1. Single-body SfM with missing data

Consider a set of P_n point trajectories extracted from a single object n that rigidly moves in F frames. By stacking each image trajectory in a single matrix $\mathbf{W}_n$ of size $2F \times P_n$, it is possible to express the global motion and the 3D shape matrices of the single object in a bilinear form as:

$$\mathbf{W}_n = \mathbf{M}_n \mathbf{S}_n = \begin{bmatrix} \mathbf{R}_{1n} & \vec{t}_{1n} \\ \vdots & \vdots \\ \mathbf{R}_{Fn} & \vec{t}_{Fn} \end{bmatrix} \begin{bmatrix} X_1 & \dots & X_n \\ Y_1 & \dots & Y_n \\ Z_1 & \dots & Z_n \\ 1 & \dots & 1 \end{bmatrix}, \tag{1}$$

where $\mathbf{M}_n$ is the $2F \times 4$ motion matrix and $\mathbf{S}_n$ is the $4 \times P_n$ shape matrix in homogeneous coordinates. Each frame-wise element $\mathbf{R}_{fn}$, for $f = 1, \dots, F$, is a 2×3 orthographic camera matrix that has to satisfy the metric constraints of the model (i.e. $\mathbf{R}_{fn}\mathbf{R}_{fn}^T = \mathbf{I}_{2\times 2}$). The 2-vector $\vec{t}_{fn}$ represents the 2D translation of the rigid object. We also introduce the registered $\overline{\mathbf{W}}_n$ measurement matrix such that $\overline{\mathbf{W}}_n = \mathbf{W}_n - \vec{t}\,\vec{1}_{P_n}^T$, where $\vec{1}_{P_n}^T$ is a vector of P_n ones and $\vec{t} = [\vec{t}_1^T, \dots, \vec{t}_F^T]^T$. Thus, one of the bilinear factors includes a set of non-linear constraints given by the camera matrix, i.e. $\mathbf{M}_n$ resides in a specific *motion manifold* [17].

In the case of missing data, for example due to occlusions or interrupted image tracks, we define the binary mask matrix G_n of size $2F \times P_n$ such that a 1 represents a known entry and a 0 denotes a missing one. In order to solve for the bilinear components, and thus the SfM problem, the equivalent optimisation problem [14] can be defined as[1]:

$$\begin{aligned} &\text{minimise} && \|\mathbf{G}_n \odot (\mathbf{W}_n - \mathbf{M}_n\mathbf{S}_n)\|^2 \\ &\text{subject to} && \mathbf{R}_{fn}\mathbf{R}_{fn}^T = \mathbf{I}_{2\times 2}, \quad f = 1, \dots, F. \end{aligned} \tag{2}$$

This problem requires not only the estimation of the camera motion $\mathbf{M}_n$ and the 3D shape $\mathbf{S}_n$, but it also needs the imputation (filling) of the missing entries in $\mathbf{W}_n$. The reader is referred to Buchanan and Fitzgibbon's work [14] for an extensive review of the approaches able to solve the missing data SfM problem.

3.2. Multi-body SfM with missing data

If the 2D image tracks belong to a set of N independently moving objects, it is still possible to formalise the problem in bilinear form. For the moment, the segmentation of each image trajectory is considered as given. Thus, by grouping the measurement in a single $\mathbf{W}$ it is possible to write:

$$\mathbf{W} = [\mathbf{W}_1 | \mathbf{W}_2 | \dots | \mathbf{W}_N], \tag{3}$$

where $\mathbf{W}_n \in \mathbb{R}^{2F \times P_n}$, for $n = 1, \dots, N$, is the trajectory matrix that contains only the P_n points of motion n (i.e. $P = \sum_{n=1}^{N} P_n$). Consequently, the $2F \times 4N$ aggregate motion matrix $\mathbf{M}$ and the $4N \times P$ aggregate shape matrix $\mathbf{S}$ are written as:

$$\mathbf{M} = [\mathbf{M}_1 | \mathbf{M}_2 | \dots | \mathbf{M}_{N-1} | \mathbf{M}_N] \quad \text{and} \quad \mathbf{S} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{0}_2 & \dots & & \mathbf{0}_N \\ \mathbf{0}_1 & \mathbf{S}_2 & \ddots & \dots & \mathbf{0}_N \\ \vdots & \mathbf{0}_2 & \ddots & \mathbf{0}_{N-1} & \vdots \\ \mathbf{0}_1 & \vdots & \ddots & \vdots & \mathbf{0}_N \\ \mathbf{0}_1 & & \dots & \mathbf{0}_{N-1} & \mathbf{S}_N \end{bmatrix}, \tag{4}$$

so that:

$$\mathbf{W} = \mathbf{W}\mathbf{S}. \tag{5}$$

It is now possible to note from Eq. (4) that the aggregate shape matrix $\mathbf{S}$ is remarkably sparse. In fact, note that each of the $\mathbf{0}_n$, for $n = 1, \dots, N$, is a matrix of size $4 \times P_n$ full of zeros.

Finally, the optimisation problem with missing data is defined as:

$$\begin{aligned} &\text{minimise} && \|\mathbf{G} \odot (\mathbf{W} - \mathbf{M}\mathbf{S})\|^2 \\ &\text{subject to} && \mathbf{R}_{fn}\mathbf{R}_{fn}^T = \mathbf{I}_{2\times 2}, \quad f = 1, \dots, F, \\ & && \quad\quad n = 1, \dots, N, \end{aligned} \tag{6}$$

where matrix $\mathbf{G}$ of size $2F \times P$ defines the overall missing entries mask. The solution of this problem not only requires the estimation of the bilinear components, but it also needs the classification of each 2D point to the correct moving body (MS step).

4. The JESS algorithm

In principle, each strategy which decreases the reprojection error of Eq. (6) is appropriate to the task. However, the bilinear formulation $\mathbf{W} = \mathbf{M}\mathbf{S}$ has a wide set of solutions, which greatly increases the chances of falling into a local minima. Just to give an evaluation of the solution space, if $\widehat{\mathbf{M}}$ and $\widehat{\mathbf{S}}$ are a solution corresponding to a minimum of Eq. (6), any non-singular matrix $\mathbf{Q}_{4N\times 4N}$ could be interposed between the factors, such as $\mathbf{W} = \widehat{\mathbf{M}}\mathbf{Q}\mathbf{Q}^{-1}\widehat{\mathbf{S}} = \widetilde{\mathbf{M}}\widetilde{\mathbf{S}}$, providing another valid solution. In the case that some of the image trajectory points are missing, the problem becomes NP-hard [15,16].

For this reason it is necessary to impose the correct constraints to the problem in order to reduce the solution space and to avoid erroneous reclassifications of the points. The flow of the proposed algorithm is shown in Fig. 4. To summarise, we propose an iterative method that exploits the initial segmentation provided by any MS algorithm (MS block in Fig. 4). Then, in an alternating fashion, both the metric constraints and the sparsity of the 3D structure matrix $\mathbf{S}$ are imposed (respectively SfM and Sparse blocks in Fig. 4). These constraints are then exploited in order to iteratively correct the segmentation (Error detection block in Fig. 4). The key idea is that misclassified points tend to disobey the motion model to which they are assigned and, therefore, they predominantly contribute to the final reprojection error. The question now is how to define an approach that can select the points that contribute the most to the error and remove them. Optionally, the algorithm could then be extended so that, once the segmentation is corrected, the removed points may be re-assigned to the proper group or discarded in the case that they are outliers (Reclass. block in Fig. 4). In the following sections each step is described in more detail.

4.1. Multi-body metric constraints

As already shown in Eq. (1), each motion is subject to the respective constraints given by the chosen camera model. Specifically, the matrix $\mathbf{M}$ cannot assume arbitrary values but it lies on a particular *motion manifold* [17]. When missing data affect the measurements, this constraint can be used both to design specific

[1] Symbol $\odot$ denotes the Hadamard product, which is the entry-wise product between matrices.

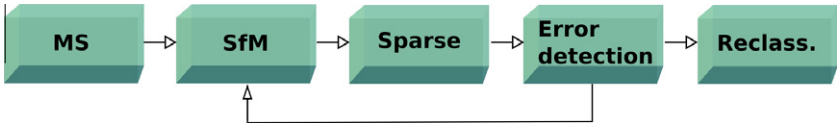


Fig. 4. Summary of the JESS algorithm.

optimisation algorithms, and to reduce the chance of computing solutions that correctly minimise the reprojection error but that lead to inaccurate 3D reconstructions [17]. In such regard, a SfM problem has to be solved for each of the N registered measurement matrices $\mathtt{W}_n$, that can possibly contain missing data. Note that this step not only finds $\mathtt{S}_n$ and $\mathtt{M}_n$ by enforcing the metric constraints, but it also fills in the missing entries of $\mathtt{W}_n$. Instead, previous methods for filling missing entries tend to eliminate the trajectories that are incomplete [6]. So, when high percentages of missing data are present, the measurement matrix will be severely decimated thus leading to an ambiguous or unsolvable problem. Hence, these approaches first estimate the segmentation, then they fill the missing data. Even in algorithms, where there is no removal, like in [7,27], they typically do not consider manifold constraints, which were shown in [17] to provide better resilience to higher ratio of missing data.

For the SfM step the Bilinear Augmented Lagrangian Multipliers (BALM) [28] method was used. BALM is a locally converging optimisation algorithm that has the property of enforcing the exact metric constraints and it shows particular robustness to high ratio of missing data. Besides, it is a generic bilinear optimiser, thus it could be used for non-rigid and articulated SfM problems as well.

Therefore, an initial segmentation is used to divide the trajectory matrix into the N trajectory matrices $\mathtt{W}_1, \ldots, \mathtt{W}_N$. The BALM algorithm is applied for each matrix $\mathtt{W}_N$ providing the motion matrix $\mathtt{M}_n$ and the structure matrix $\mathtt{S}_n$. At the end of this step the aggregate matrices described in Eq. (4) can be created. Clearly, as the initial segmentation may contain errors this result needs to be corrected. The correction is performed by exploiting the sparse constraint of the aggregate shape matrix $\mathtt{S}$, as explained in the following section.

4.2. Sparsity of matrix $\mathtt{S}$

The sparse pattern of the aggregate shape matrix can be used in order to estimate matrix $\mathtt{S}$ which best satisfies such a constraint. In order to make this problem tractable, the ℓ_1 norm can be used as a surrogate for sparsity. In such terms, the optimisation problem becomes solving, for each point p in $\mathtt{S}$, a *basis pursuit denoising* problem [29]:

$$\underset{\vec{s}_p}{\text{minimise}} \frac{1}{2}\|\vec{w}_p - \mathbf{M}\,\vec{s}_p\|_2^2 + \tau\|\vec{s}_p\|_1, \tag{7}$$

where $\tau \in \mathbb{R}^+$ is a *regularisation parameter*, $\vec{w}_p$ is the registered trajectory vector for the point p, and the $3N \times P$ matrix $\overline{\mathbf{S}} = [\vec{s}_1 \mid \ldots \mid \vec{s}_P]$ represents the collection of non-homogenous 3D coordinates. Accordingly, the $2F \times 3N$ matrix $\mathtt{M}$ is the motion matrix less the translation vector at each frame. Each $3N$-vector $\vec{s}_p$ contains $3(N-1)$ zeros, thus the image trajectory $\vec{w}_p$ can be described by a (small) subset of the 3D points. The sparse optimisation is initialised so that $\overline{\mathbf{S}} = \mathbf{S}$, where $\mathbf{S}$ is the aggregate shape matrix obtained at the end of the previous SfM step. Note that $\overline{\mathbf{S}}$, which results from the sparse optimisation, may not satisfy the 3D metric structure of the objects. However, the result of the minimisation can be used to detect points that do not comply with the previously estimated metric constraints.

For the sparse minimisation step the Sparse Reconstruction by Separable Approximation (SpaRSA) algorithm [30] was used. SpaRSA is able to solve large-scale optimisation problems efficiently and it requires only the tuning of the regularisation term τ (Eq. (7)). The parameter τ was empirically tuned on 50% of the sequences of the synthetic database. In all of the experiments the following τ values were used: for two motions $\tau = 0.9$, for three motions $\tau = 1.6$, for four motions $\tau = 2.5$ and for five motions $\tau = 3.2$.

At the end of this step it is possible to define two different reconstructed trajectory matrices, the one obtained after imposing the metric constraints:

$$\widetilde{\mathbf{W}} = [\mathbf{M}_1\mathbf{S}_1 | \ldots | \mathbf{M}_N\mathbf{S}_N], \tag{8}$$

and the one obtained by optimising the reprojection error while imposing the sparsity constraint:

$$\widehat{\mathbf{W}} = [\mathbf{M}_1\overline{\mathbf{S}}_1 | \ldots | \mathbf{M}_N\overline{\mathbf{S}}_N]. \tag{9}$$

4.3. Identifying candidate errors

The intuition behind JESS is that misclassified points tend to disobey the motion to which they are wrongly assigned. Therefore, in order to reduce the effects of the association with the wrong motion, during the sparse optimisation their 3D coordinates will tend to change more than those of other points. In order to identify the candidate errors the trajectory matrices $\widetilde{\mathbf{W}}$, obtained after the SfM stage, and $\widehat{\mathbf{W}}$, obtained after the sparse minimisation stage, are compared. Since SfM and sparse optimisation perform arbitrary normalisations on $\widetilde{\mathbf{W}}$ and $\widehat{\mathbf{W}}$, these two matrices have to be registered. Specifically, the image centroids have to be evaluated in a unique reference space, and the mean distance of all of the points from the origin is imposed to be equal to $\sqrt{2}$.

After registration, the 2D distance between $\widetilde{\mathbf{W}}$ and $\widehat{\mathbf{W}}$, for each point p and for each frame f is computed. Two measures to identify the candidates are used: (a) the point p_a with the highest 2D reprojection difference for any of the F frames and (b) the point p_b with the highest mean 2D reprojection difference over all the F frames. Therefore, at each iteration the points p_a and p_b are removed from the trajectory matrix.

4.4. Stop condition

At each iteration, the candidate misclassified points are removed until a stop condition is verified. Once the candidate errors are removed the algorithm can iterate again from the beginning until a stop condition is verified. Note that a valid stop condition could be to fix the number of points that JESS will remove (assuming that the average behaviour of the used MS algorithm is known). Note also that, if desired, the last step of the algorithm can reclassify all the removed points that were considered candidate errors, so that no information is lost.

Nevertheless, JESS can be run with another stop condition for those cases where there is no prior information about the expected misclassification rate of the MS algorithm. The most intuitive condition is to use the reprojection error of Eq. (6): when the error decreases below a threshold, then the algorithm should stop. However, tests showed that such an error could have a non increasing behaviour when the number of errors in the segmentation increases. On the other hand, it was also noted that when the segmentation is correct the reprojection error tends to become stable. Hence, one useful condition is that the difference in the

reprojection error between one iteration and the following has to be less than a threshold for a fixed amount of iterations. Specifically, it was empirically chosen that the reprojection error has to be smaller than 5×10^{-7} for at least three consecutive JESS iterations. Moreover, this first condition was associated with a second one that is: the 2D reprojection difference of the candidate point p_a (or p_b) has to be smaller than a threshold (in all of the experiments such a threshold was set to 0.5, which was empirically found to be a good overall threshold). When both conditions are satisfied the algorithm can terminate. The conditions are purposely very strict as it is preferable to perform more iterations and remove as many errors as possible rather than to stop the algorithm too early and leave even a single error in the segmentation. However, to avoid the algorithm running indefinitely a maximum number of iterations can also be set.

4.5. Reclassification

Once the stop condition is satisfied, all of the removed points can be reclassified to the correct motion by exploiting the NSI similarity measure [24]. Moreover, such a measure can also be used to detect outliers which, by definition, do not belong to any of the motions.

A summary of JESS is shown in Algorithm 1. Starting from an initial motion segmentation, the main building blocks of JESS are the computation of SfM with missing data, which enforces the metric constraints, the sparse minimisation, which detects candidate errors, and the reclassification step, which enables the reassignment of the detected misclassified points to the correct motion.

Algorithm 1. JESS

1: Compute an initial MS, arrange $\mathbf{w}$ as in Eq. (3) and build $\overline{\mathbf{W}}$.
2: **repeat**
3: $\forall$ motion $n = 1, \ldots, N$ compute SfM: $\widetilde{\mathbf{W}}_n = \mathbf{M}_n\mathbf{S}_n$.
4: Perform sparse minimisation, Eq. (7), and obtain $\bar{\mathbf{S}}$.
5: Compute $\widetilde{\mathbf{W}}$ as in Eq. (8), and $\widehat{\mathbf{W}}$ as in Eq. (9).
6: Register the two metrices to their respective centroids.
7: $\forall$ points $p = 1, \ldots, P$ and $\forall$ frames $f = 1, \ldots, F$ compute 2D distance $Dist(p,f)$ between $\widetilde{\mathbf{W}}$ and $\widehat{\mathbf{W}}$.
8: Find $p_a = \max_p(Dist(f,p)) \forall f = 1, \ldots, F$.
9: Find $p_b = \max_p\left(\sum_{f=1}^{F} Dist(f,p)/F\right)$.
10: Remove p_a and p_b (if $p_b \neq p_a$) from $\overline{\mathbf{W}}$.
11: **until** stop condition satisfied
12: Reclassify removed points

5. Experiments

We have evaluated the following different features of our approach: the validity of our algorithm independently from the stop condition, the results using the stop condition, the ability of the algorithm to deal with missing data, the performance of the reclassification strategy and the quality of the final 3D reconstruction. For simplicity, whenever JESS is stopped before the reclassification strategy it will be called JESS-R, whereas the name JESS will be used to refer to the complete algorithm (including the reclassification step).

While testing the efficacy of JESS, we also compared its results with a RANSAC-based [9] algorithm in order to detect outliers (i.e. misclassified points) in a given segmentation. The RANSAC-based algorithm is summarised in Algorithm 2. The desired probability of selecting a set of inliers was set to 99%, the number of iterations required per each object was computed following [9] and assuming that the number of errors were equally distributed among the objects. We tested different minimum number of points that had to be randomly selected in order to compute the SfM model (from 4 to 10 points) and we present here the best results (obtained with 4 points per object). Once the best SfM model was found the reprojection error of each point was computed and used in order to detect outliers. Specifically, the given a sequence with ($\mathcal{M}$) initially misclassified points $3 \times \mathcal{M}$ points with the highest reprojection error (in any given frame) and the $3 \times \mathcal{M}$ with the highest mean reprojection error (over all the frames) were considered outliers (i.e. misclassified points).

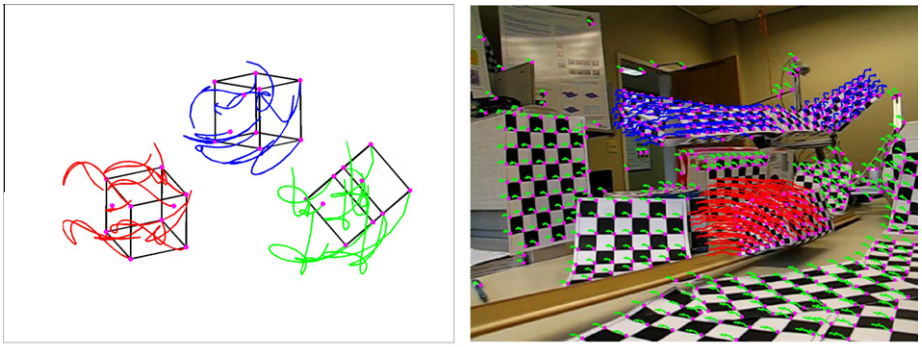
Algorithm 2. Refining initial MS with RANSAC

1: Compute an initial MS for N motions, arrange $\mathbf{w}$ as in Eq. (3) and build $\overline{\mathbf{W}}$.
2: **for all** motion $n = 1, \ldots, N$ **do**
3: Compute number of required RANSAC iterations as in [9]
4: **for all** required RANSAC iterations **do**
5: Build $\widetilde{\mathbf{W}}_n^4$ by randomly selecting 4 trajectories from $\widetilde{\mathbf{W}}_n$.
6: Compute SfM: $\widetilde{\mathbf{W}}_n^4 = \mathbf{M}_n^4\mathbf{S}_n^4$.
7: Compute structure for all points: $\mathbf{S}_n = \mathbf{M}_n^{4\dagger}\widetilde{\mathbf{W}}_n$.
8: Compute $\widehat{\mathbf{W}}_n = \mathbf{M}_n^4\mathbf{S}_n$.
9: Compute reprojection error between $\widehat{\mathbf{W}}_n$ and $\widetilde{\mathbf{W}}_n$.
10: **end for**
11: Identify $\widehat{\mathbf{W}}_n$ with minimum reprojection error
12: $\forall$ points $p = 1, \ldots, P_n$ and $\forall$ frames $f = 1, \ldots, F$ compute 2D distance $Dist(p,f)$ between $\widetilde{\mathbf{W}}_n$ and $\widehat{\mathbf{W}}_n$.
13: **for** $3 \times \mathcal{M}$ iterations **do**
14: Remove points as in JESS steps 8,9,10.
15: **end for**
16: **end for**

The experiments were performed on different datasets that are now described in detail.

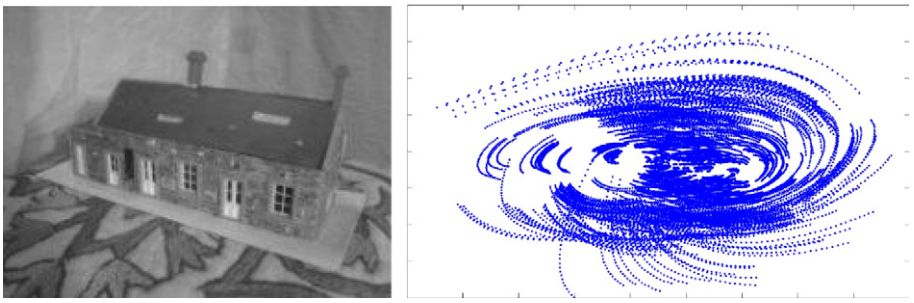
Synthetic sequences, full and missing data. The proposed synthetic sequences contained a set of moving cubes, with 56 tracked features each, that randomly rotate and translate. The database included different sequences of 50 frames each with a varying number of independent motions and noise. Specifically, we tested 10 randomly generated motions with 2, 3, 4 and 5 independently moving objects (cubes) giving a total of 40 sequences. Gaussian noise with zero mean and standard deviation of 0, 0.5, 1, 1.5 and 2.0 pixels was added to each sequence (for a total of 200 sequences). We ran different tests with an increasing number of initially misclassified points (randomly selected) starting with the simplest case of 1 misclassified point per sequence in order to check algorithmic convergence. The remaining tests were performed with higher number of misclassified points: 1%, 2%, 3%, 4%, 5% and 10% of the total number of points in each sequence. Moreover, in order to simulate occlusions in one of the tests we randomly removed 10% of the data of each sequence. An example of a synthetic frame (which, for clarity, is plotted with just few tracked features) is shown in Fig. 5a.

Real sequences, full and missing data. The Hopkins155 database [4] contained 104 checkerboard sequences, 38 traffic sequences and 13 other sequences (among which are sequences with articulated motions). We have also used the 12 additional checkerboard sequences (in presence of missing data due to occlusions) pub-



(a) *Synthetic* (b) *2RT3RC*

Fig. 5. Examples of synthetic and real sequences.



(a) First frame (b) Trajectories

Fig. 6. Example of the House sequence.

lished in [31] by the same authors. For simplicity we will refer to this additional set as the Hopkins12 database. An example of a real frame from Hopkins155 is shown in Fig. 5b.

The Hopkins155 and Hopkins12 databases are established benchmarks in the MS community, however, as far as we know they have never been used for SfM as the amount of motion contained in the sequences is barely enough to satisfy the SfM constraints. It is well known, in fact, that the quality of the reconstruction provided by SfM algorithms is greatly affected by degenerate motions. We refer the reader to [32,33] for a deeper analysis of the relationship between SfM and degenerate motion sequences. It is important, however, to note that as JESS relies on the ability to impose those constraints, these database are particularly challenging. In order to assess the performance of JESS in cases where the SfM constraints are satisfied we have a created another data set, the *House and Hotel*.

House and Hotel. These two sequences are well known in the SfM community. Frame examples of the two sequences are shown in Figs. 6 and 7. The two sets of trajectories were unified in a unique sequence and translated apart so that the whole set of points could be seen as a new real sequence that contains two moving objects. The new sequence is composed of 30 frames, the House object contains 672 points while the Hotel object contains 133 points for a total of 805 trajectories.

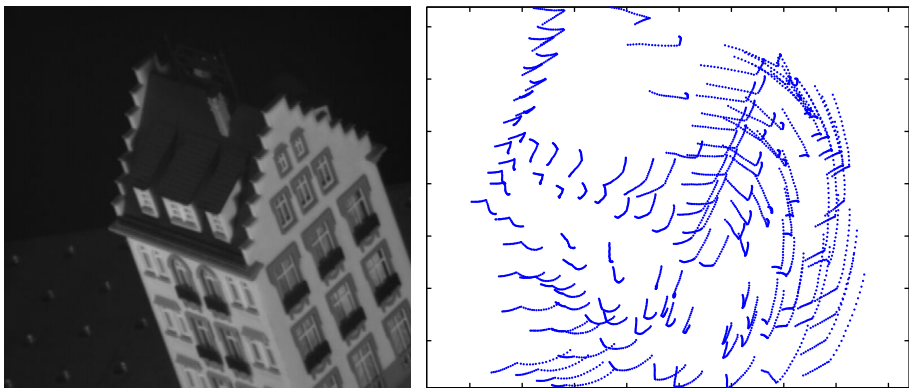
5.1. JESS with a fixed number of iterations

The first set of experiments evaluated the ability of JESS-R to converge to the correct segmentation and compared the results of JESS-R with those obtained by using a RANSAC-based algorithm. Accordingly, JESS-R was run for a fixed number of iterations without imposing a stop condition. The number of iterations was $3 \times \mathcal{M}$, where $\mathcal{M}$ was the number of initially misclassified points. The RANSAC-based solution was run as explained in Algorithm 2 and similarly to JESS-R the number of iterations performed to detect outliers was set to $3 \times \mathcal{M}$.

In Fig. 8 it is possible to observe the percentage of identified errors (i.e. how many of the initially misclassified points were identified as a percentage of all of the original misclassified points, a line with $\bigcirc$ symbol identifies JESS-R results while a line with $\times$ identifies RANSAC results) for each different number of motions and level of noise. The results show a very robust behaviour of JESS-R against different numbers of motion. JESS-R is robust also in the presence of an increasing amount of noise and initial errors: the percentage of error detection is 100% for almost all of the cases. Also, it is worth to stress the importance of the tests with only 1 misclassified point, in fact these tests showed that even in the presence of only 1 error, JESS-R was always able to detect it (among all of the $56 \times N$ points of each sequence) within only 3 iterations, showing the validity of the algorithm.

The RANSAC-based solution performs equally well to JESS-R only in presence of two objects and with only 1 or 1% of initially misclassified points. The other experiment setups show how the RANSAC-based solution is degraded by any of the aspects that make the sequences more challenging: number of objects, number of initially misclassified points and noise level. In summary, JESS-R outperforms RANSAC, its error detection being, in some setups, even 60% higher than the one of RANSAC (four motions, 4% of initially misclassified points and σ of the noise equal to 1.0).

The same experiment with the same settings was repeated for JESS-R on the real sequences of the Hopkins155 database, where initial misclassified points were randomly selected. Averaged results are shown in Fig. 9. Similarly to the results on the synthetic database, the real test demonstrates the stability of the behaviour



(a) First frame (b) Trajectories

Fig. 7. Example of the Hotel sequence.

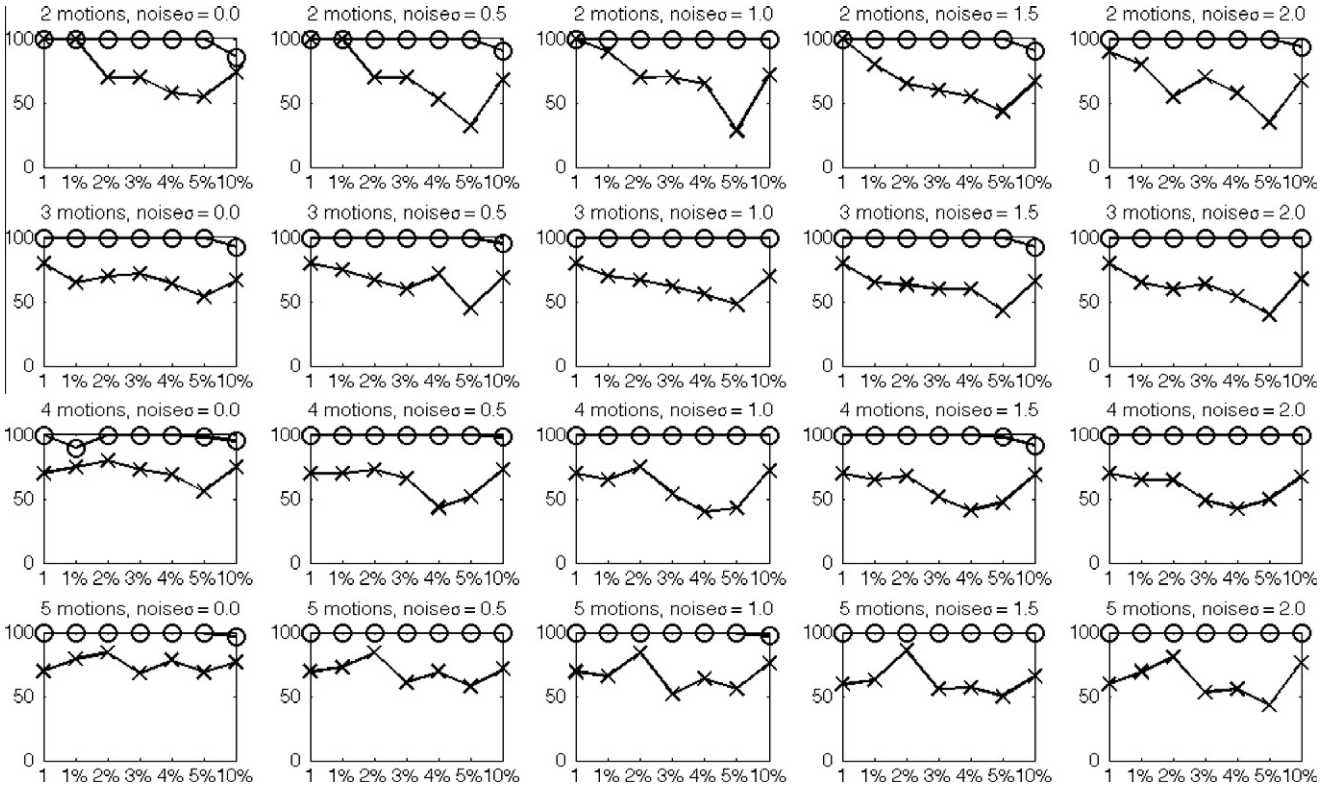


Fig. 8. Average results of JESS-R (○) and RANSAC (×) with a fixed amount of iterations applied to the synthetic database that contains different numbers of motion (from 2 to 5). On the *x*-axis the initial amount of misclassified points is shown, on the *y*-axis the percentage of detected errors is shown. ○ and × are the percentages of the removed misclassified points over all the misclassified points.

of JESS-R against the initial misclassification rate: the percentage of error detection is stable between 70% and 80%. Even in this experiment RANSAC performs similarly to JESS-R for 1 misclassified point and 1% of misclassified points, while when the amount of misclassification increases JESS-R is on average around 10% more accurate than RANSAC.

This first set of tests shows that the problem is not trivial as a RANSAC-based solution was not able to detect most of the initial misclassified points even in the synthetic experiment setups. On the other hand, it shows that JESS-R is able to greatly reduce the misclassification rate of the input sequences. On the synthetic database correction is almost perfect while on the challenging sequences of the Hopkins155 database any of the initial misclassification rates is reduced, at least, by 70%.

5.2. Stop condition

The aim of the second set of experiments was to verify the proposed stop condition (Section 4.4). In order to stop the algorithm in the case that the stop condition is never verified, we set the maximum number of iterations to be $(x + 3)\%$ of the points of the sequence, x being the percentage of initial misclassification.

Results on the synthetic database are shown in Fig. 10. The detection of the errors is very similar to the case with a fix amount of iterations. In this set of experiments also the amount of false positives, as a percentage of all of the points of the sequence, are reported (a line with × symbol). False positives are points that were removed by JESS-R even if they were correctly classified. As shown in Fig. 10 the amount of false positives ranges between

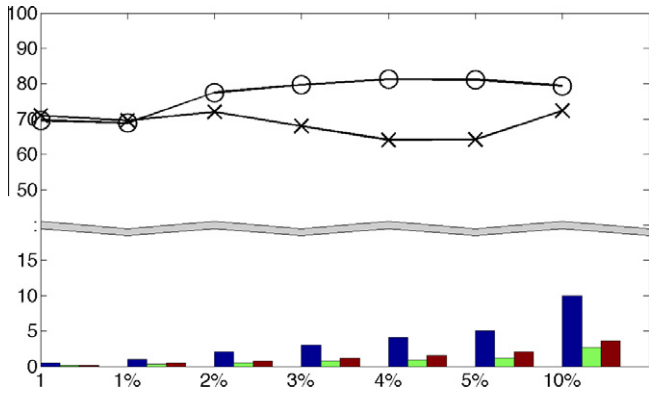


Fig. 9. Average results of JESS-R (○) and RANSAC (×) applied to the Hopkins155 database. On the *x*-axis the initial amount of misclassified points is shown, on the *y*-axis the percentage of detected errors over all the misclassified points is shown. On the bottom of the plot the bars show from left to right and for each percentage of misclassified points: initial misclassification, misclassification after removal of errors by JESS-R, misclassification after removal and reclassification by JESS.

5% to 15%. An almost perfect error detection behaviour testifies that the condition is not usually satisfied until all of the errors have been removed. On the other hand, a small amount of false positives indicates that the algorithm terminates the loop not too long after detection of the last error. All of the removed points, including the false positives, will be reassigned to the correct motion by the reclassification strategy.

Results on the Hopkins155 database are shown in Fig. 11. The results are very stable and show only a slight decreasing trend when the misclassification rate increases. This suggests that the stop condition may be too strict in the presence of high misclassification rates and with dependent motions. Nevertheless, in the case of an error rate not higher than 3%, JESS-R is able to reduce the initial misclassification rate by 70%, even when a tuning process for the stop condition is avoided. With a higher amount of initially misclassified points, the error detection is reduced of about 60% for initial misclassification rates of 4% and 5%, and of more than 50% for an initial misclassification rate of 10%. Similarly to the results on the synthetic database, the amount of false positives is stable: in the case of the Hopkins155 database is around 4%.

5.3. Missing data

This set of experiments was performed on synthetic and real sequences with missing data in order to demonstrate the algorithm performance on such challenging cases. Tests on the synthetic and the Hopkins155 databases were performed with 10% of missing data (randomly selected) and using the proposed stop condition. Fig. 12 shows the results of JESS-R on the synthetic database. The behaviour of JESS-R is very robust and the percentage of error detection is almost always 100% and never less than 80%. On the Hopkins155 database JESS-R is able to keep an error detection rate of approximately 60% for all of the amounts of initial misclassification tested, as shown in Fig. 13.

A further test on real sequences was performed on the Hopkins12 database. In this case missing data were not simulated but were due to occlusions, while misclassified points were given by the errors of the GPCA [31] and SSC [6] motion segmentation algorithms[2]. The results are summarised in Table 1. The amount of missing data varied from 0.96% to 22.20% of the total points, with an average rate of 8.33%, while initial misclassification was between 0% and 48.6%. In this test the imposed maximum number of iterations (in case the stop condition is not satisfied before) was equal to 10% of the points of the sequence (clearly much less than the initial misclassification of some of the sequences). Considering all of the sequences with an initial error rate not higher than 10%, JESS-R is able to detect the errors and decrease the initial misclassification in the majority of the cases.

When the initial misclassification level is above 10% results are less significant for two reasons: average MS algorithms error rate is nowadays much smaller than 10%, and since JESS is an iterative algorithm it cannot be expected to converge if the initialisation is far from the correct solution. Therefore, those cases are not considered in the following discussion. The misclassification rate of JESS-R applied to GPCA did not improve the initial rate (excluding of course the cases when the initial misclassification was already 0%) only in three sequences (*oc1R2RC_g23* whose misclassification remained constant, *oc2R3RCRT_g13* whose misclassification became worse by 0.32%, and *oc2R3RCRT_g23* whose misclassification became worse by 1.07%). In all of the three cases this is due to the fact that the stop condition was prematurely verified. Similarly, when JESS-R is applied to the results of SSC, only in one case JESS-R did not improve the initial segmentation (*oc1R2RC_g13*, whose misclassification became worse by 0.01%).

Overall, the results of these tests showed that JESS-R can deal successfully also with sequences that contain missing data. Also when JESS-R was tested on the Hopkins12 and Hopkins155, which are composed by sequences that are not ideal for the application of SfM constraints, JESS-R was able to reduce the initial misclassification rate in most of the cases. Moreover, in those few case where the misclassification was not improved, the error introduced by JESS-R was only marginal.

5.4. Reclassification strategy

All of the results discussed until here have concerned the detection and removal of segmentation errors. If it is required, once the segmentation has been improved, the removed points can be reintroduced using a reclassification strategy, as explained in Section 4.5. The reclassification strategy on the synthetic database with a fixed number of iterations shows a success rate, on average, of 99.99%. When the stop condition is used, and more points are removed, the success rate remains very high: 99.95%. The same result was confirmed also with the missing data, with a success rate of 99.97%.

Results of the reclassification strategy on the Hopkins155 database are shown on the bottom of the plots of Figs. 9, 11 and 13. The first bar shows the initial misclassification, the second presents the misclassification after removal of the points, and the third gives the misclassification after the reclassification strategy. Often the misclassification before and after the reclassification remains the same (i.e. the reclassification works perfectly in most of the cases), only in few tests the error rate after the reclassification is slightly increased. This small increment that happens in some cases also testifies that the SfM constraints used by JESS, and never exploited before, can solve some of the cases where rules (like NSI) used in classical motion segmentation algorithms would fail.

Overall, these results confirm that if the segmentation is mostly correct the reclassification strategy is able to reclassify the removed points (both errors and false positives) correctly. The same test was also applied in the case of missing data on the Hopkins12 database and the results are shown in the JESS column of Table 1. Even in this case, it is possible to appreciate that error rates before and after the reclassification are very similar. Moreover, in none of the relevant cases (i.e. sequences whose initial misclassification rate was below 10%) JESS had a worse misclassification than GPCA,

[2] GPCA and SSC implementations available at `vision.jhu.edu`; SSC parameters used were: $\tau = 0.01$, subspace size equal to 4, cluster step performed by Random Walks [34].

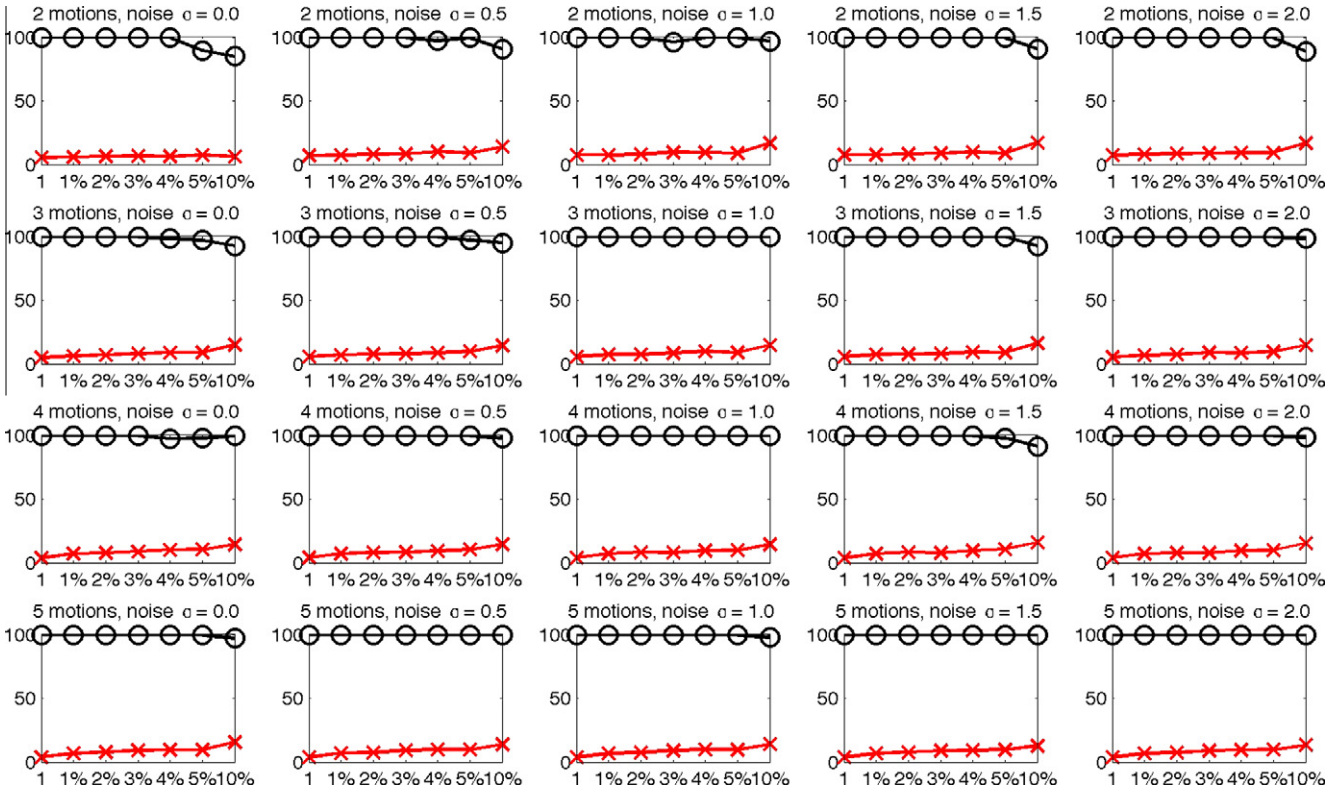


Fig. 10. Average results of JESS-R with stop condition applied to the synthetic database that contains different numbers of motion (from 2 to 5). On the *x*-axis the initial amount of misclassified points is shown, on the *y*-axis the percentage of detected errors is shown. ○ is the percentage of the removed misclassified points over all the misclassified points. False positives as a percentage of the total amount of points of each sequence are shown with an ×.

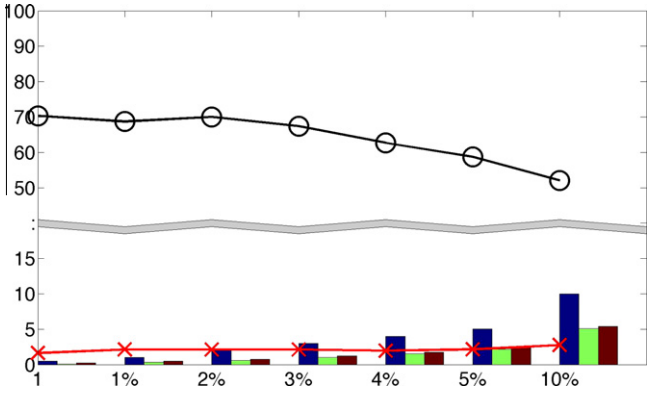


Fig. 11. Average results of JESS with stop condition applied to the Hopkins155 database. On the *x*-axis the initial amount of misclassified points is shown, on the *y*-axis the percentage of detected errors over all the misclassified points is shown (JESS-R). False positives as a percentage of the total amount of points of each sequence are shown with an ×. On the bottom of the plots the bars show from left to right and for each percentage of misclassified points: initial misclassification, misclassification after removal of errors by JESS-R, misclassification after removal and reclassification by JESS.

and only in one case (*oc1R2RCT_23*) JESS had a worse misclassification than SSC by only 0.23%. In all of the remaining cases the final misclassification is always equal or smaller than the one provided by the tested MS algorithms.

As far as the computational time is concerned, on the whole Hopkins155 database with 1 error per sequence JESS (including reclassification) required on average approximately 20 s per sequence (Matlab implementation on Quad-Core @ 2.4 GHz, with 16 GB RAM). Note that the stage of the algorithm that was most time consuming was the sparse optimisation. This time could be shortened by adopting high performance implementations of sparse optimisation on Graphic Processing Units [35].

5.5. JESS applied to real MS algorithms

In this section JESS is applied on the initial segmentation provided by one of the most recent and successful MS algorithms on the Hopkins155 database: the Automatic Subspace Affinity (ASA) [8]. ASA is not able to deal with missing data, however, it was chosen also because is one of the few that does not require tuning of any parameter, therefore, it can be directly applied to the tested sequences. As for the previous tests, results are presented before the reclassification step (JESS-R), which is an optional step, and after the reclassification step (JESS).

In Fig. 14 the misclassification rate of ASA, JESS-R and JESS are shown. Note that JESS-R improves the performance of ASA both with two (from 0.96% to 0.76%) and three motions (from 2.23% to 1.85%). Also JESS improves the misclassification rate of ASA, however, some of the detected errors are wrongly reintroduced by the reclassification strategy, this leads to a slightly higher error rate than JESS-R. Nevertheless, the error rate of JESS is never worse than the one of ASA (from 0.96% to 0.86% with two motions, while for three motions the error remains constant).

Note that the average initial misclassification rates provided by ASA were already small, hence, it is not possible to note big improvements in terms of average error rate. Nevertheless, some improvement can be found and a deeper analysis will reveal that the most important contribution is hidden to the information provided by the average misclassification rate. The results provided by

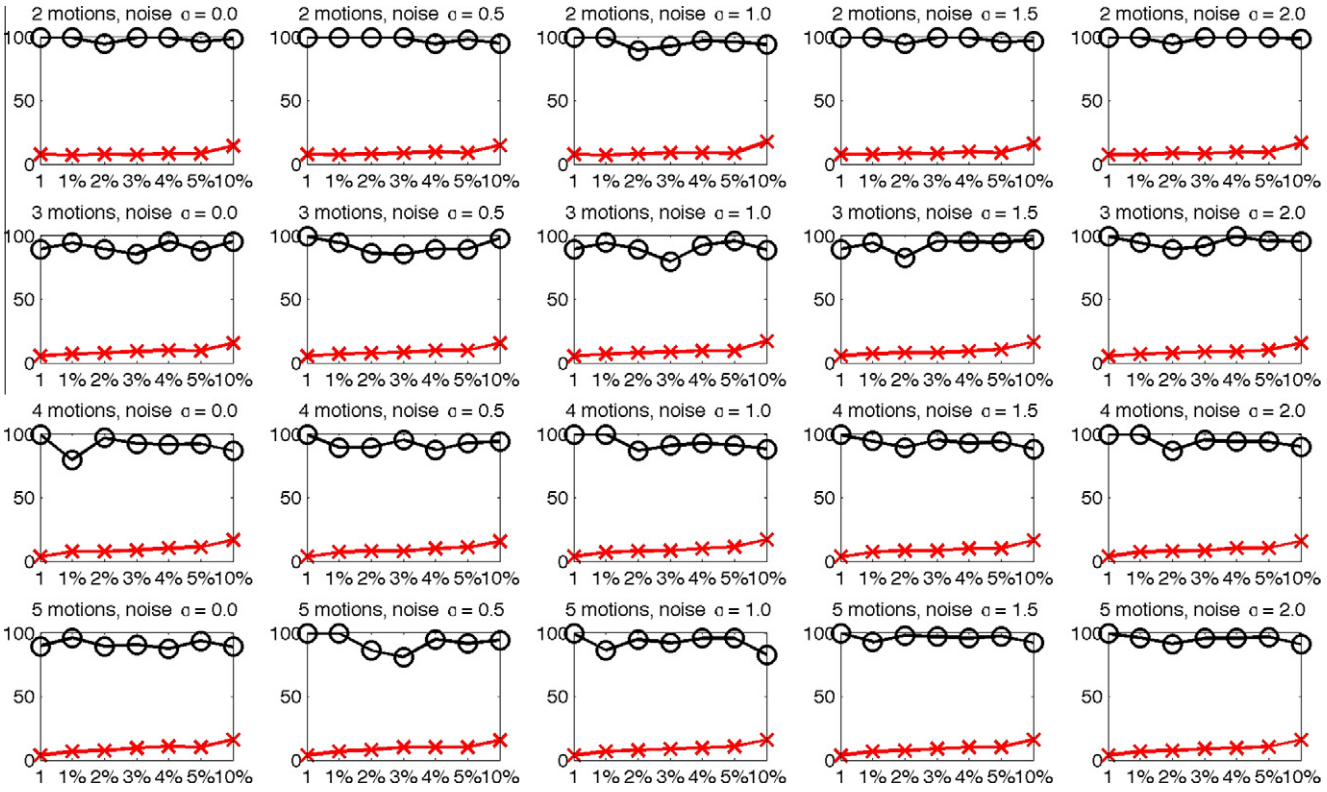


Fig. 12. Average results of JESS-R with stop condition and 10% of missing data applied to the synthetic database that contains different numbers of motion (from 2 to 5). On the *x*-axis the initial amount of misclassified points is shown, on the *y*-axis the percentage of detected errors is shown. ○ is the percentage of the removed misclassified points over all the misclassified points. False positives as a percentage of the total amount of points of each sequence are shown with an ×.

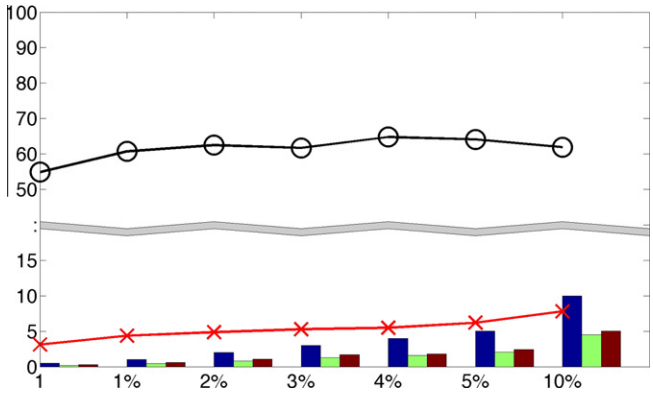


Fig. 13. Average results of JESS with stop condition and 10% of missing data applied to the Hopkins155 database. On the *x*-axis the initial amount of misclassified points is shown, on the *y*-axis the percentage of detected errors over all the misclassified points is shown (JESS-R). False positives as a percentage of the total amount of points of each sequence are shown with an ×. On the bottom of the plots the bars show from left to right and for each percentage of misclassified points: initial misclassification, misclassification after removal of errors by JESS-R, misclassification after removal and reclassification by JESS.

ASA were composed by many sequences with a small error rate and few sequences with a very high error rate. When the initialisation is good then JESS converges to the correct solution, however, the impact on the average misclassification rate is minor. On the other hand, when the initialisation of JESS contains a high initial misclassification rate, convergence of the algorithm is challenging. In fact, those few sequences that have a high misclassification rate, and therefore that contribute heavily to the final rate, do not satisfy the assumption of JESS-R/JESS. Table 2 offers a detail of the misclassification rates. Particularly interesting is that the checkerboard sequences are those where the correction has been less effective, whereas articulated and traffic sequences show a better correction rate. The reason for this difference relies probably on the fact that checkerboard sequences contain objects that perform very small motions, while in the other two groups the motions performed are bigger, and therefore, the ability of the SfM step to impose camera constraints is more effective. This result gives also more significance to the previous results of JESS shown on the Hopkins155 database and on Hopkins12 database, which is composed exclusively by checkerboard sequences.

Another interesting analysis comes from the study of the histograms of the misclassification rates shown in Fig. 15. From this histogram the effectiveness of JESS-R/JESS can be appreciated much more than from the overall misclassification rates. Note that JESS-R and JESS improved mainly the sequences with an initial rate below 5% (as expected), while for sequences with a higher initial rate the improvement was little. On the other hand, the number of sequences with no error at all increased from 101 of ASA to 128 of JESS-R (then to 115 of JESS).

This final test showed that, given an initialisation provided by a MS algoirthm, JESS-R/JESS can successfully correct errors of the segmentation. The reclassification strategy is able to reclassify correctly almost all of the removed points. However, in few cases the reclassification may fail. This proves that the constraints imposed by JESS are a key feature for improving the performances of classical MS algorithms. Moreover, note that when it is possible (i.e. when the remaining points after JESS-R removal are still sufficient for the desired task) the reclassification is not necessary. The reclassification strategy is also one of the steps where further

Table 1
Average results of JESS applied on the results of the GPCA and SSC algorithms on the Hopkins12 database. MD: Missing Data; GPCA/SSC: misclassification of GPCA or SSC algorithm; JESS-R: misclassification of the JESS algorithm before reclassification; JESS: misclassification of the complete JESS algorithm.

Name	MD (%)	GPCA (%)	JESS-R (%)	JESS (%)	SSC (%)	JESS-R (%)	JESS (%)
oc1R2RC	4.8	34.30	35.76	34.45	0.46	0.00	0.00
oc1R2RCT	4.5	12.36	9.67	11.82	2.36	0.22	1.27
oc1R2RCT_g12	10.1	4.33	0.00	2.16	0.00	0.00	0.00
oc1R2RCT_g13	5.2	3.52	1.02	1.64	3.05	1.28	1.41
oc1R2RCT_g23	1.0	3.16	0.00	0.00	0.00	0.00	0.23
oc1R2RC_g12	10.0	0.00	0.00	0.00	0.00	0.00	0.00
oc1R2RC_g13	6.0	2.24	0.00	0.00	0.41	0.42	0.41
oc1R2RC_g23	0.6	0.78	0.78	0.78	0.00	0.00	0.00
oc2R3RCRT	13.1	38.33	35.71	33.62	48.61	50.00	49.68
oc2R3RCRT_g12	22.2	4.94	4.29	3.70	0.00	0.00	0.00
oc2R3RCRT_g13	9.7	5.37	5.69	5.37	39.13	39.05	38.62
oc2R3RCRT_g23	12.7	9.97	11.04	9.71	44.09	47.32	41.73

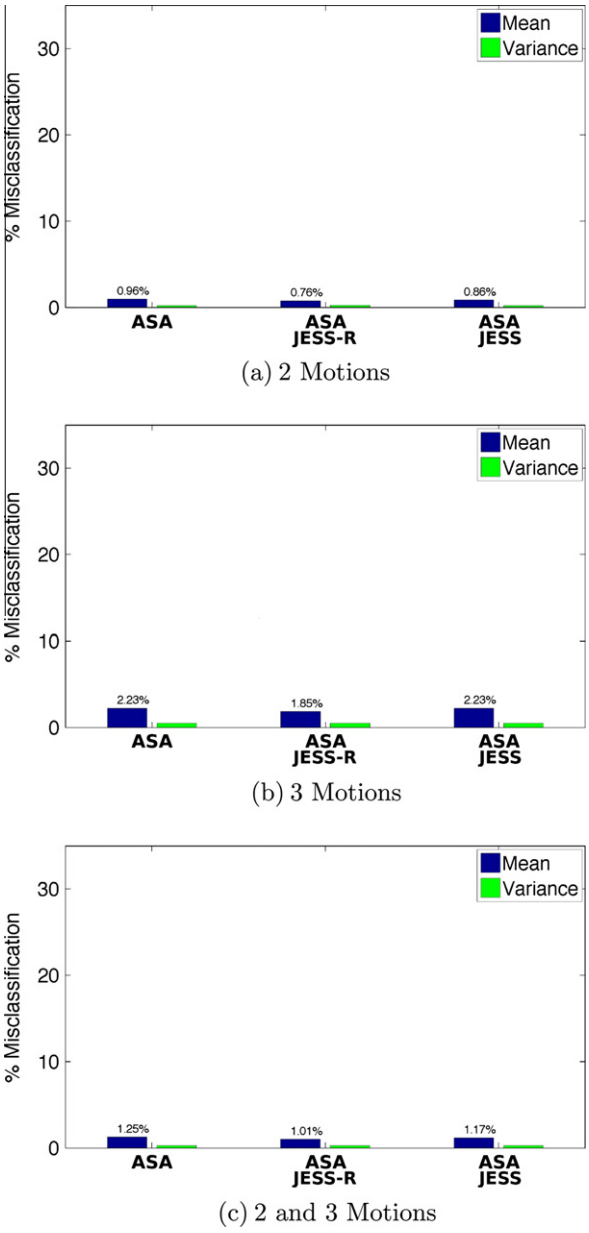


Fig. 14. Mean and variance misclassification rate of ASA before and after application of JESS.

Table 2
Misclassification rates on the Hopkins155 database of ASA with JESS-R (no reclassification) and JESS.

Method	%Avg	%Var	%Avg	%Var	%Avg	%Var	%Avg	%Var
Two motions	Check. (78)		Artic. (11)		Traffic (31)		All (120)	
ASA	1.00	0.32	1.75	0.10	0.57	0.01	0.96	0.22
ASA + JESS-R	0.96	0.33	1.47	0.16	0.00	0.00	0.76	0.23
ASA + JESS	1.04	0.32	1.92	0.10	0.03	0.00	0.86	0.22
Three motions	Check. (26)		Artic. (2)		Traffic (7)		All (35)	
ASA	2.41	0.65	3.72	0.28	1.11	0.03	2.23	0.49
ASA + JESS-R	2.39	0.67	1.19	0.03	0.03	0.00	1.85	0.51
ASA + JESS	2.68	0.64	3.19	0.20	0.27	0.00	2.23	0.49

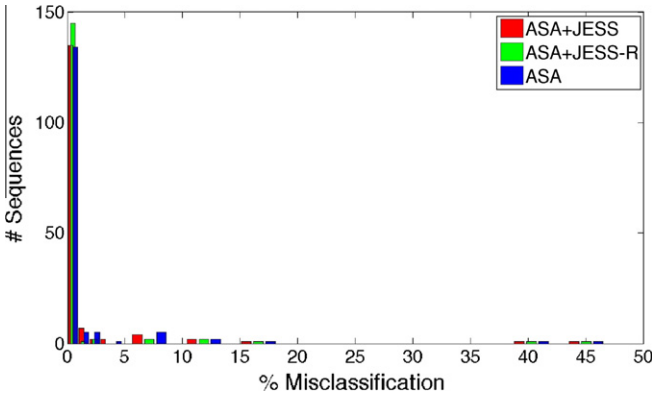


Fig. 15. Histogram of the misclassification rate of ASA with JESS-R and JESS on the Hopkins155 database; misclassification rates from 0% to 5% are sub-sampled with bins of 1%, misclassification rates greater than 5% are sub-sampled with bins of 5%.

investigation could lead to even better results. The final misclassification rate obtained combining ASA with JESS-R is one of the smallest in the state of the art of MS among techniques that do not require a tuning stage. More importantly, JESS produces one of the highest number of perfect segmentation (128 sequences over 156), which is essential for SfM to take place.

5.6. House and Hotel test

At the beginning of the section it was anticipated that the Hopkins155 and the Hopkins12 databases were not ideal test sets for JESS as they are designed for pure MS algorithms. As JESS merges MS with SfM, it is required that the tested sequences satisfy SfM constraints. The tests presented on GPCA, SSC and ASA showed this issue explicitly. In fact, the checkerboards sequences, which theoretically are among the easiest sequences in terms of MS, are those

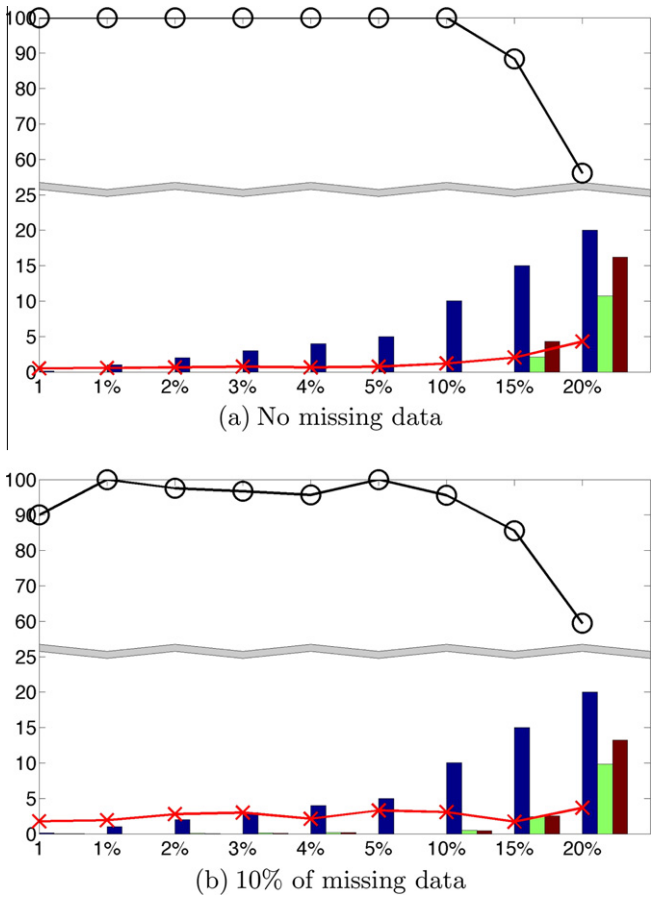


Fig. 16. Average results of JESS with stop condition applied to the House and Hotel sequence. On the *x*-axis the initial amount of misclassified points is shown, on the *y*-axis the percentage of detected errors over all the misclassified points is shown (JESS-R). False positives as a percentage of the total amount of points of each sequence are shown with an ×. On the bottom of the plots the bars show from left to right and for each percentage of misclassified points: initial misclassification, misclassification after removal of errors by JESS-R, misclassification after removal and reclassification by JESS.

in which the improvement gained with JESS is smaller. Now a case in which motions are suitable for SfM reconstruction is presented. The aim of this test is to verify the behaviour of JESS when SfM constraints can be satisfied.

Different tests were performed on the House and Hotel dataset. First JESS-R and JESS were tested with a variable amount of initial misclassification (up to 20% of the total amount of points) and no missing data. The results shown in Fig. 16a are the average results of 10 different runs; each run consisted of a different random selection of the misclassified points. Until a misclassification of 10% the error detection is perfect and the reclassification is able to reintroduce all of the removed points correctly. With 15% of initial misclassification the error detection becomes of 88.35% and therefore the misclassification rate goes from the 15% of the initialisation down to 4.2% with JESS-R, and finally to 8.6% with JESS. When the initial misclassification rate is of 20% around 56% of the errors are detected.

The second experiment was performed with 10% of missing data (randomly selected). Results are shown in Fig. 16b. As expected performances are slightly worse than the previous case, however, the trend of the error detection is still very similar.

In order to investigate further the ability to cope with missing data, another experiment was performed with an initial misclassification rate fixed to 10% and variable missing data from 0% to 40%. The average results of 10 runs are shown in Fig. 17. In this test it is

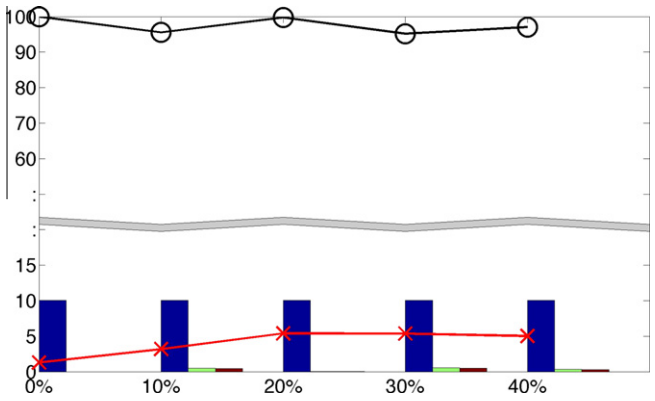


Fig. 17. Average results of JESS with stop condition applied to the House and Hotel sequence with an initial amount of misclassification equal to 10%. On the *x*-axis the amount of missing data, points is shown, on the *y*-axis the percentage of detected errors over all the misclassified points is shown (JESS-R). False positives as a percentage of the total amount of points of each sequence are shown with an ×. On the bottom of the plots the bars show from left to right and for each percentage of misclassified points: initial misclassification, misclassification after removal of errors by JESS-R, misclassification after removal and reclassification by JESS.

possible to appreciate the robustness of JESS with respect to missing data. In fact in all of the tests JESS is able to identify almost perfectly the 10% of initially misclassified points, remove them and reclassify them correctly.

Tests performed on the House and Hotel sequence confirmed the conclusions drawn from the analysis of the results on the Hopkins155 and Hopkins12. Specifically, if the SfM constraints are satisfied JESS is able to detect the initially misclassified points very effectively, even in presence of a high percentage of missing data. Nevertheless, even when SfM constraints cannot be fully satisfied, like in the Hopkins155 and the Hopkins12 database, JESS is still able to correct an initial misclassification, however, in this case a more relevant improvement cannot be expected.

Finally, it is worth to spend some comments on the quality of the 3D reconstruction of the House and Hotel sequence. In this case there is no ground truth for the 3D reconstruction, therefore, the reconstruction of the two buildings when there is no error in the segmentation is used as main reference. The 3D reconstruction of each of the two buildings is shown when the cloud of points of each object contained 10%, 5% and 0% of misclassified points. Only the points that belong to the correct object are then taken into account for the Procrustes analysis in order to align the reference reconstruction with the one obtained by JESS. The results are shown in Fig. 18–21. These results show how the presence of even only few misclassified points greatly affect the 3D reconstruction. Moreover, in these examples the misclassified points were removed from the analysis because the ground truth of the segmentation was known. However, in an unknown scenario it is not possible to rely on prior knowledge and therefore the ability of JESS to remove misclassified points is vital. For both buildings JESS was able to completely correct the segmentation and finally provide the reconstructions shown in Figs. 18–21d.

Note that the 3D error is always higher for the House sequence. This is due to the small amount of motion contained in this sequence (refer to Figs. 6 and 7 to compare respectively the House and the Hotel motions). This shows once again that if the motion is degenerate (in this case the amount of motion is not sufficiently long), metric constraints cannot be satisfied. In the case of the House and Hotel sequence the presence of a long motion of one of the two objects was sufficient for JESS to detect misclassified points. However, in cases when none of the motions are long enough, like for the Hopkins155 and Hopkins12 databases, the effect of imposing metric constraints in the optimisation is less effective.

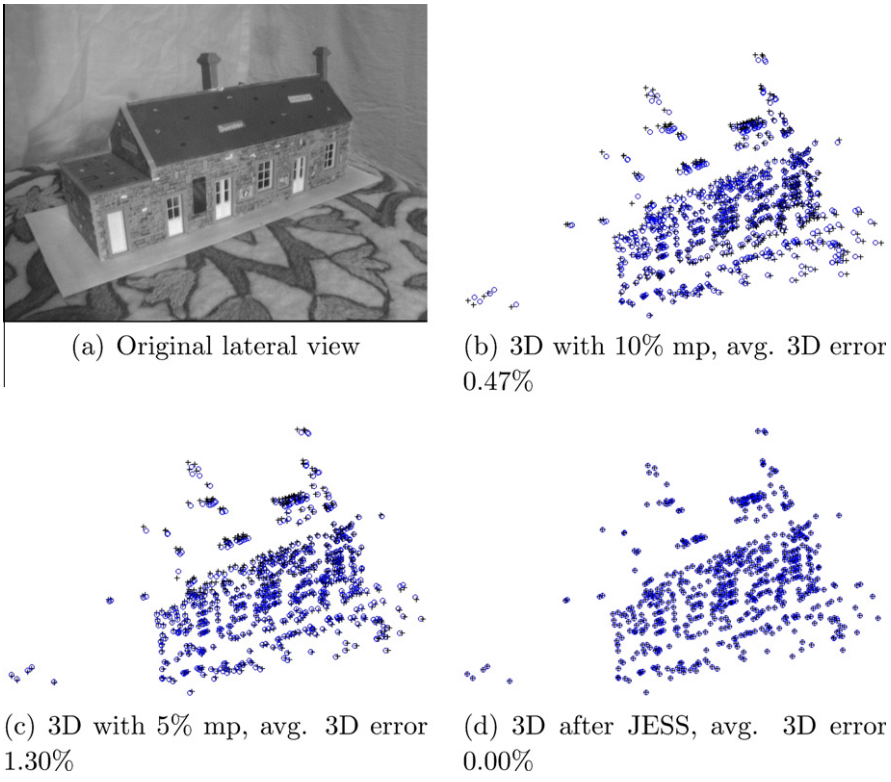


Fig. 18. 3D reconstruction of JESS on the House and Hotel sequence with different amount of misclassified points (mp). Metric size of House ground truth is $3.88 \times 6.25 \times 7.67$. Lateral view of House. ○ are the ground truth point positions, + are the 3D reconstructions. Errors shown as a percentage of the ground truth depth size.

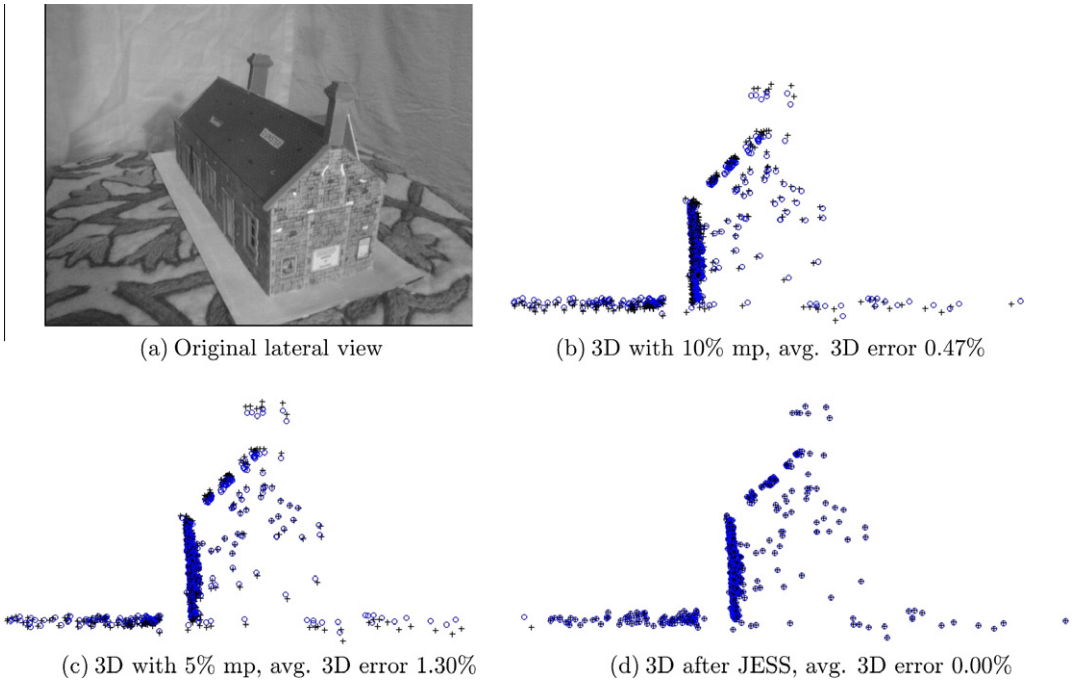
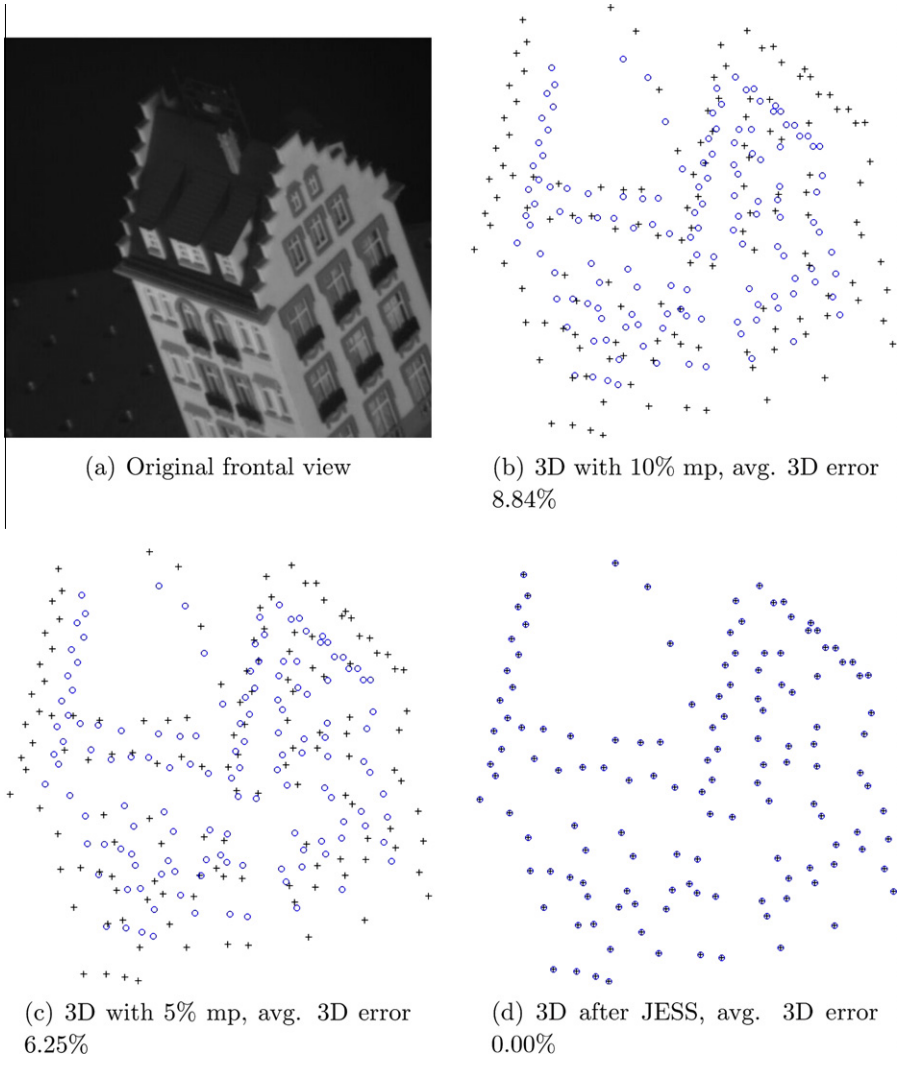


Fig. 19. 3D reconstruction of JESS on the House and Hotel sequence with different amount of misclassified points (mp). Metric size of House ground truth is $3.88 \times 6.25 \times 7.67$. Back view of House ○ are the ground truth point positions, + are the 3D reconstructions. Errors shown as a percentage of the ground truth depth size.

6. Conclusion

In this paper an optimisation framework for joint estimation of segmentation and structure from motion (JESS) has been presented. JESS is able to estimate the multi-body motion segmentation and the 3D reconstruction from point trajectories even in the presence of missing data. This approach starts from an initial segmentation and iterates to jointly include the metric constraints, given by an orthographic camera model, and the constraint that arises from the fact that the shape matrix that describes the multi-body 3D shape is generally sparse. The metric constraints are used to compute the 3D metric shapes and to fill the missing en-



(a) Original frontal view

(b) 3D with 10% mp, avg. 3D error 8.84%

(c) 3D with 5% mp, avg. 3D error 6.25%

(d) 3D after JESS, avg. 3D error 0.00%

Fig. 20. 3D reconstruction of JESS on the House and Hotel sequence with different amount of misclassified points (mp). Metric size of Hotel ground truth is $3.32 \times 3.80 \times 4.64$. Lateral view of Hotel. ○ are the ground truth point positions, + are the 3D reconstructions. Errors shown as a percentage of the ground truth depth size.

tries, while the sparse optimisation of the shape matrix detects and removes wrong assignments of the trajectories. A final optional step can be used to reclassify points initially misclassified. Note that while camera matrix and sparsity constraint have been previously used in SfM and multi-body SfM, they have never been used to solve the MS problem. Moreover, to these authors knowledge, JESS is unique in this field as there is no previous framework able to correct the results of a MS algorithm and, simultaneously, to compute the 3D structure of the moving objects. An alternative to JESS could be the RANSAC algorithm, however, we showed that JESS in general outperforms RANSAC. Moreover, when the initial misclassification rate increases the number of iterations required by RANSAC increases exponentially.

The experiments on short sequences (synthetic, Hopkins155 and Hopkins12) showed the validity of JESS in spite of the fact the algorithm has more potentiality when dealing with longer sequences (i.e. with longer motions). JESS was able to detect almost all of the initial errors also in the case of missing data. Furthermore, the fact that in some cases errors were correctly identified by JESS but then wrongly reclassified by the reclassification step, not only indicates room for possible improvements, but it also proves that the SfM constraints used by JESS can play a key role for those cases where traditional MS algorithms (like the reclassification algorithm) would fail. The House and Hotel sequence showed that for non degenerate motions, i.e. when SfM constraints can be satisfied, the efficiency of JESS is greatly improved compared to the cases with degenerate motion sequences.

In summary, it has been shown that JESS algorithm is an effective solution for the multiple-body SfM problem. Moreover, thanks to its modularity JESS can benefit from any improvement made to any of its main modules (MS, SfM, and sparse optimisation). The main limitations of JESS are in the ability of MS algorithms to provide a good initialization, and in the ability of single-body SfM algorithms to deal with degenerate motions.

In terms of future work, JESS could be improved by taking into account the specific nature of each motion. In fact, in the SfM step all the motions were treated as if they were rigid, even if in the Hopkins155 database there are some articulated and non-rigid motions. Therefore, by treating non-rigid and articulated motions properly the performance of JESS could be further improved. Moreover, as explained in Section 5.4 the performance in terms of computational time could be boosted by adopting implementations of sparse optimisation on Graphic Processing Units [35].

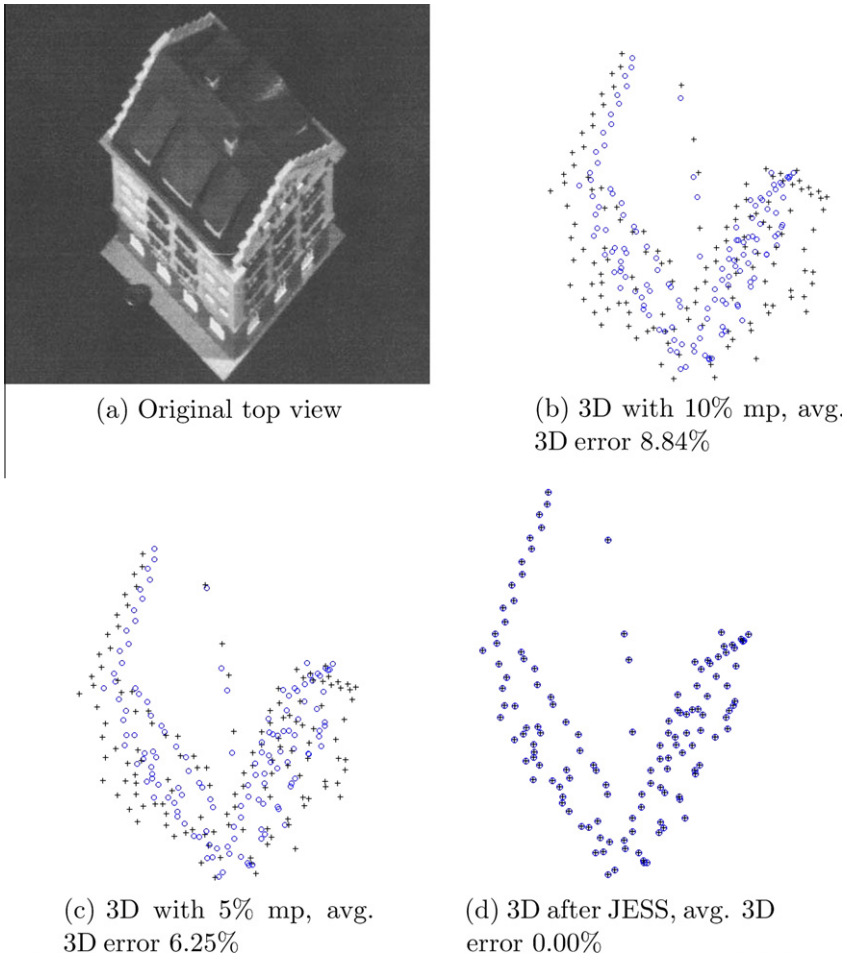


Fig. 21. 3D reconstruction of JESS on the House and Hotel sequence with different amount of misclassified points (mp). Metric size of Hotel ground truth is $3.32 \times 3.80 \times 4.64$. Top view of Hotel. ○ are the ground truth point positions, + are the 3D reconstructions. Errors shown as a percentage of the ground truth depth size.

Acknowledgments

This work has been supported by the Spanish Ministry of Science and Innovation projects CTM2011-29691-C02-02. L. Zappella was supported by the Catalan government scholarship 2009FI B1 00068. The authors would like to thank the reviewers for their valuable and crucial comments.

References

[1] C. Tomasi, T. Kanade, Detection and Tracking of Point Features, Technical Report, 1991.
[2] C. Zach, I.A., H. Bischof, What can missing correspondences tell us about 3d structure and motion? in: Proc. CVPR IEEE, 2008.
[3] K. Ozden, K. Schindler, L. Van Gool, Multibody structure-from-motion in practice, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1134–1141.
[4] R. Tron, R. Vidal, A benchmark for the comparison of 3-d motion segmentation algorithms, Proc. CVPR IEEE (2007) 1–8.
[5] R. Vidal, R. Tron, R. Hartley, Multiframe motion segmentation with missing data using PowerFactorization and GPCA, Int. J. Comput. Vision. 79 (2008) 85–105.
[6] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Proc. CVPR IEEE, 2009, pp. 2790–2797.
[7] R. Vidal, R. Hartley, Motion segmentation with missing data using PowerFactorization and GPCA, Proc. CVPR IEEE 2 (2004) 310–316.
[8] L. Zappella, E. Provenzi, X. Lladó, J. Salvi, Adaptive motion segmentation algorithm based on the principal angles configuration, in: Lect. Notes Comput. Sci. (ACCV), vol. 6494, 2011, pp. 15–26.
[9] M. Fischler, R. Bolles Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, in: Comm. of the ACM, vol. 24, 1981, pp. 381–395.
[10] L. Zappella, A. Del Bue, X. Lladó, J. Salvi, Simultaneous motion segmentation and structure from motion, in: Proc. of the IEEE Intern. Conf. on Motion and Video Computing, 2011, pp. 679–684.
[11] C. Bregler, A. Hertzmann, H. Biermann, Recovering non-rigid 3D shape from image streams, in: Proc. CVPR IEEE, 2000, pp. 690–696.
[12] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization method, Int. J. Comput. Vis. 9 (1992) 137–154.
[13] P. Tresadern, I. Reid, Articulated structure from motion by factorization, Proc. CVPR IEEE 2 (2005) 1110–1115.
[14] A.M. Buchanan, A. Fitzgibbon, Damped newton algorithms for matrix factorization with missing data, Proc. CVPR IEEE 2 (2005) 316–322.
[15] D. Nistér, F. Kahl, H. Stewénius, Structure from motion with missing data is NP-hard, IEEE ICCV (2007) 1–7.
[16] N. Gillis, F. Glineur, Low-rank approximation with weights or missing data is NP-hard, SIAM J. Matrix Anal. Appl. 32 (2011) 1149–1165.
[17] M. Marques, J. Costeira, Estimating 3d shape from degenerate sequences with missing data, Comput. Vis. Image Und. 113 (2009) 261–272.
[18] J. Costeira, T. Kanade, A multi-body factorization method for motion analysis, IEEE ICCV (1995) 1071–1076.
[19] A.W. Fitzgibbon, A. Zisserman, Multibody structure and motion: 3-d reconstruction of independently moving objects, in: Lect. Notes Comput. Sci. (ECCV), 2000, pp. 891–906.
[20] R. Vidal, R. Hartley, Three-view multibody structure from motion, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 214–227.
[21] K. Schindler, D. Suter, W. Wang, Hanzi, A model-selection framework for multibody structure-and-motion of image sequences, Int. J. Comput. Vis. 79 (2008) 159–177.
[22] L. Zappella, X. Lladó, E. Provenzi, J. Salvi, Enhanced local subspace affinity for feature-based motion segmentation, Pattern Recogn. 44 (2011) 454–470.
[23] F. Lauer, C. Schnorr, Spectral clustering of linear subspaces for motion segmentation, IEEE ICCV (2009) 678–685.
[24] N. Pinho da Silva, J. Costeira, The normalized subspace inclusion: Robust clustering of motion subspaces, IEEE ICCV (2009) 1444–1450.
[25] A. Cheriyadat, R. Radke, Non-negative matrix factorization of partial track data for motion segmentation, IEEE ICCV (2009) 865–872.

[26] S.R. Rao, R. Tron, R. Vidal, Y. Ma, Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories, Proc. CVPR IEEE (2008) 1–8.
[27] C. Julià, A. Sappa, F. Lumbreras, J. Serrat, A. Lpez, An iterative multiresolution scheme for SfM with missing data: Single and multiple object scenes, Image Vision Comput. 28 (2010) 164–176.
[28] A. Del Bue, J. Xavier, L. Agapito, M. Paladini, Bilinear modelling via Augmented Lagrange Multipliers (BALMs), IEEE Trans. Pattern Anal. Mach. Intell. 34 (8) (2012) 1496–1508.
[29] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1998) 33–61.
[30] S. Wright, R. Nowak, M. Figueiredo, Sparse reconstruction by separable approximation, IEEE Trans. Signal Process. 57 (2009) 2479–2493.
[31] R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (GPCA), IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1945–1959.
[32] J. Oliensis, The least-squares error for structure from infinitesimal motion, Int. J. Comput. Vis. 61 (2005) 259–299.
[33] P.H.S. Torr, A.W. Fitzgibbon, A. Zisserman, The problem of degeneracy in structure and motion recovery from uncalibrated image sequences, Int. J. Comput. Vis. 32 (1999) 27–44.
[34] L. Grady, Random walks for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1768–1783.
[35] S. Lee, S. Wright, Implementing Algorithms for Signal and Image Reconstruction on Graphical Processing Units, Technical Report, Computer Sciences Department, University of Wisconsin-Madison, 2008.