

Simultaneous Motion Segmentation and Structure from Motion

Luca Zappella
IliA University of Girona
zappella@eia.udg.edu

Alessio Del Bue
IIT Genova
alessio.delbue@iit.it

Xavier Lladó
IliA University of Girona
llado@eia.udg.edu

Joaquim Salvi
IliA University of Girona
qsalvi@eia.udg.edu

Abstract

*This paper presents a novel approach to simultaneously compute the motion segmentation and the 3D reconstruction of a set of 2D points extracted from an image sequence. Starting from an initial segmentation, our method proposes an iterative procedure that corrects the misclassified points while reconstructing the 3D scene, which is composed of objects that move independently. This optimization procedure is made by considering two well-known principles: firstly, in multi-body Structure from Motion the matrix describing the 3D shape is **sparse**, secondly, the segmented 2D points must give a valid 3D reconstruction given the **rotational metric constraints**. Our formulation results in a bilinear optimization where sparsity and metric constraints are enforced at each iteration of the algorithm. The final result is the corrected segmentation, the 3D structure of the moving objects and an orthographic camera matrix for each motion and each frame. Results are shown on synthetic sequences and a preliminary application on real sequences of the Hopkins155 database is presented.*

1. Introduction

The inference of the 3D position of moving objects in a scene is one of the most important tasks in Computer Vision. In complex scenarios where several bodies rigidly move, it is first necessary to cluster the motion belonging to different objects before performing any other reconstruction task. In particular, *Motion Segmentation* (MS) from feature trajectories consists of segmenting the trajectories that move with different motions throughout a video sequence. MS is a low-level task and it is a fundamental step for any further motion analysis. Its importance is denoted by the active research within this field since the beginning of computer vision to date. Different strategies have been used to tackle MS as described in [17]: image difference, statistics,

wavelets, Optical Flow, Layers and Manifold clustering to cite a few. Recently the Hopkins155 database [14] has become a standard benchmark for the evaluation of MS techniques. A few algorithms [7, 9, 18] reported low misclassification rates on the Hopkins155 database which testifies that MS algorithms are becoming more reliable.

Once a segmentation is available with image trajectories assigned to each object, other higher level tasks such as 3D reconstruction can take place. 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). Numerous techniques have been proposed to solve the SfM problem, one of the most successful approaches has been the Tomasi and Kanade’s factorization algorithm [12] developed in the early 90’s. The key idea of their method is to express the geometric invariants present in the data as a bilinear model of its 3D shape and motion components. Tomasi and Kanade’s algorithm extracts these components globally by using the whole information contained in the trajectory matrix of the moving shape. The algorithm was later extended to work with more general camera models for rigid objects [4, 10, 13] and more recently to deal with non-rigid objects [2]. All these techniques share a common assumption: there is only one object moving in the scene. In order to perform multiple reconstruction, it is necessary to rely on a MS technique that feeds the SfM algorithm with one object at a time or to develop a different framework for multi-body SfM.

1.1. Related works on multi-body SfM

The early attempts to solve multi-body SfM tried to tackle the problem using algebraic approaches [5]. After algebraic approaches, which are very sensitive to noise, re-

searchers started to take into account the geometry of the problem and to exploit the epipolar geometry constraints, like in [8]. Pure epipolar geometry constraints however, are also sensitive to noise and to outliers. A simplified situation using only 2-views and noise-free sequences is presented in [16]. Extension of epipolar geometry to the 3-views case mixed with algebraic tools can be found in [15]. In [11] a scheme for n -views multi-body SfM with a perspective camera has been presented. The method mixes two-view epipolar geometry with statistical considerations that are used to link the motions found in the two-views throughout the sequence.

In summary, most of multi-body SfM algorithms so far tried to solve the MS intrinsically with the SfM. Mainly they tried to exploit epipolar geometry mixing algebraic or statistical tools. The main limitation has always been the presence of noise and outliers. Nowadays the advantage we have, besides the knowledge gained thanks to the previous studies, is that MS algorithms provide rather good solutions. Therefore, this initial segmentation can be exploited to help to solve the multi-body SfM problem, which in turn can provide unexploited constraints to correct the segmentation.

1.2. Our contribution

Our proposed approach defines a novel optimization procedure in order to remove the errors of an initial segmentation and simultaneously solve the multi-body 3D SfM. We present a generic framework that can be applied to correct the result of any MS algorithm and goes in the direction of merging the problems of MS and SfM. Given an initial segmentation, we develop an iterative bilinear optimization algorithm that, at each iteration, enforces two constraints that arise from SfM and multi-body SfM theory: the *metric constraints* given the camera projection matrices and the *sparsity* of the matrix that contains the 3D shapes of all the moving objects. Thanks to this optimization we are able to correct the errors for the initial segmentation and solve the multi-body SfM problem.

The remainder of the paper is organized as follows. In section 2 we introduce the background and the notation of the SfM framework. Our proposed bilinear MS and SfM approach is then presented in section 3. The summary of the algorithm and some implementation choices are explained in section 4. The experiments on synthetic and real data sets which validate our approach are shown in section 5. Finally, conclusions are given in section 6.

2. Background

2.1. Single-body SfM

The first step of SfM consists of storing in the trajectory matrix $\bar{W} \in \mathbb{R}^{2F \times P}$, the coordinates of the P 2D features tracked throughout a video sequence of F frames. From

SfM theory we know that, given one single motion, it is possible to factorize the trajectory matrix into a motion matrix and a shape matrix. The shape matrix is defined up to a scale factor, therefore, \bar{W} can be normalized as follows:

$$\bar{W} = (W - T)/\gamma, \quad (1)$$

where $T \in \mathbb{R}^{2F \times P}$ is a translation matrix that registers the coordinates to the image centroid¹ and $\gamma \in \mathbb{R}^+$ is the scale factor computed as the sum of the maximum x and y values in $(W - T)$. At this point it is possible to compute the camera motion and 3D shape such that:

$$\bar{W} = MS, \quad (2)$$

where $S \in \mathbb{R}^{3 \times P}$ is the shape matrix and $M \in \mathbb{R}^{2F \times 3}$ is the motion matrix further defined as:

$$M = \begin{bmatrix} R_1 \\ \vdots \\ R_F \end{bmatrix}, \quad (3)$$

R_f , for $f = 1, \dots, F$, being orthographic camera matrices of size 2×3 .

The common principle underlying all the previously mentioned SfM techniques is that in order to have a valid 3D reconstruction the *metric constraints* have to be enforced, which means that:

$$R_f R_f^T = I, \quad \forall f = 1, \dots, F, \quad (4)$$

where I is the identity matrix of size 2×2 .

2.2. Extension to multi-body SfM

Let us extend equation (2) to the multi-body case in presence of N motions, assuming for the moment that the correct segmentation is known. The trajectory matrix \bar{W} is now defined as:

$$\bar{W} = [\bar{W}_1 | \bar{W}_2 | \dots | \bar{W}_N], \quad (5)$$

where $\bar{W} \in \mathbb{R}^{2F \times P}$ while $\bar{W}_n \in \mathbb{R}^{2F \times P_n}$, for $n = 1, \dots, N$, is the trajectory matrix that contains only the points of motion n , and P_n is the number of points that belong to motion n , therefore $P = \sum_{n=1}^N P_n$. Equation (2) holds, but the structure of the matrices involved changes. Specifically, the aggregate motion matrix $M \in \mathbb{R}^{2F \times 3N}$ is composed as follows:

$$M = [M_1 | \dots | M_N] = \begin{bmatrix} R_{11} & \dots & R_{1N} \\ \vdots & \ddots & \vdots \\ R_{F1} & \dots & R_{FN} \end{bmatrix}, \quad (6)$$

where R_{fn} , for $f = 1, \dots, F$ and $n = 1, \dots, N$, is a 2×3 orthographic camera matrix that has to satisfy the metric

¹The matrix T is given by $T = \bar{t} \bar{1}_P^T$ where $\bar{1}_P$ is a vector of P ones. Thus the $2F$ -vector \bar{t} contains the 2D shape centroid at each frame.

constraints (i.e. $RR^T = I$), equation (4). From equations (2) and (6) it follows that the aggregate shape matrix $S \in \mathbb{R}^{3N \times P}$ is:

$$S = \begin{bmatrix} S_1 & 0 & \dots & 0 \\ 0 & S_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & S_N \end{bmatrix}, \quad (7)$$

where S_n , for $n = 1, \dots, N$, is the shape matrix of the motion n and it has size $3 \times P_n$. As can be seen from equation (7), the aggregate shape matrix S is highly sparse.

3. Bilinear MS and SfM

Despite the fact that the mathematical model for SfM can be extended to multi-body SfM, it cannot be directly used in order to find a unique solution. However, if one can start from an already good, but not necessarily perfect, segmentation then the *metric constraints* of each of the motion matrices, and the *sparsity* structure of the aggregate shape matrix could be exploited in order to correct the segmentation and perform a multi-body SfM.

We propose an iterative method that exploits the initial segmentation provided by any MS algorithm. The proposed approach is an alternation approach that first imposes the metric constraints, given the camera matrix, and then enforces the sparsity of the 3D structure matrix S . In the following we are describing each step in more detail.

3.1. Enforcing metric constraints

Starting from an initial segmentation, the matrices M_n and S_n , for $n = 1, \dots, N$, can be computed independently for each motion by any SfM algorithm. Clearly, the motion matrix and the shape matrix obtained, called \tilde{M}_n and \tilde{S}_n , will not be the exact M_n and S_n (i.e. the real motion and shape matrices), nevertheless, if it is true that the majority of the points P_n belong to motion n , \tilde{M}_n and \tilde{S}_n are good initializations for further refinements.

We define:

$$\tilde{M} = [\tilde{M}_1 | \dots | \tilde{M}_N], \quad (8)$$

and

$$\tilde{S} = \begin{bmatrix} \tilde{S}_1 & 0 & \dots & 0 \\ 0 & \tilde{S}_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \tilde{S}_N \end{bmatrix}, \quad (9)$$

as being the initial values, and the bilinear form is now:

$$\tilde{w} = \tilde{M}\tilde{S}. \quad (10)$$

At this stage we have a first multi-body SfM solution where the metric constraints have been enforced for each motion by applying a single-body SfM algorithm.

3.2. Exploiting S sparsity

It is at this step that we exploit the sparse structure of S (S being the ground truth of the aggregate shape matrix). In fact, we want to refine the initial solution provided by \tilde{M} and \tilde{S} so that $\|\tilde{w} - \tilde{M}\tilde{S}\|_2$ is minimized but, in order to limit the solution space, we want also to impose that the final \hat{S} has to be sparse. As the majority of points in each motion n should be correctly segmented, it is reasonable to assume that $\tilde{M} \approx M$ (M being the ground truth motion matrix), therefore, it is easier to focus, at this step, only on \tilde{S} . The refining problem can be formulated as a *basis pursuit denoising* problem [3]:

$$\min_{\hat{S}} \|\hat{S}\|_1 \text{ subject to } \|\tilde{w} - \tilde{M}\hat{S}\|_2 \leq \epsilon, \quad (11)$$

where $\epsilon \in \mathbb{R}^+$ and the initial value for $\hat{S} = \tilde{S}$. At the end of the sparse optimization step we define:

$$\hat{w} = \tilde{M}\hat{S}. \quad (12)$$

The minimization problem of equation (11) focuses on imposing the sparsity of \hat{S} and on keeping the reprojection error small, but it does not enforce the metric constraints as in the previous SfM step. Hence, it cannot be expected that $\hat{S} = S$. Our assumption is that because of the sparse minimization the misclassified points will experience a bigger change, with respect to the correctly classified ones, in their 3D position. Such change should allow the misclassified points to obey the (metric) motion enforced in the previous step. In general we observed that the columns of the matrix \hat{S} associated to a misclassified point will vary more than the columns of a correctly classified point. The main issue now is to find a correct criteria to identify such points.

3.3. Identifying candidate errors

In order to identify the candidate errors we compare the trajectory matrices \tilde{w} , equation (10), and \hat{w} , equation (12). First we align the 2D data and bring them to a common scale, since sparse and SfM optimisations perform arbitrary normalization of the trajectory matrix. Hence, we translate and normalize the 2D homogeneous points of \tilde{w} and \hat{w} so that the centroid is at the origin, and their mean distance from the origin is $\sqrt{2}$. After this normalization step, the 2D distance between \tilde{w} and \hat{w} , for each point n and for each frame f , can be computed. We define two measures to identify the candidates: 1) the point $p1$ with the highest 2D reprojection difference for any of the F frames, 2) the point $p2$ with the highest mean 2D reprojection difference over all the F frames. Most of the time $p1 = p2$, however, occasionally they may be different. The candidate point(s) can now be removed from \tilde{w} and the process can start again to compute the SfM for each motion.

4. Implementation details

4.1. Summary of the algorithm

The three steps described in the previous sections (enforcing metric constraints, exploiting sparsity and identifying candidate errors) are summarized in the iterative Algorithm 1. The termination criteria depends on the specific application and on the *a priori* knowledge that may be available. In our experiments we wanted to evaluate the reliability of our proposal independently from the stop condition. Therefore, we allow the algorithm to iterate for a maximum of $3 \times \mathcal{X}$ times, where \mathcal{X} is the number of misclassified points. If after $3 \times \mathcal{X}$ iterations there is still at least one misclassified point we consider that sequence not corrected.

Algorithm 1 Simultaneous MS and SfM

- 1: Compute an initial MS, arrange \bar{w} as in equation (5).
 - 2: \mathcal{X} = number of misclassified points.
 - 3: *iterations* = 0. *done* = 0.
 - 4: **repeat**
 - 5: *iterations* = *iterations* + 1.
 - 6: \forall motion $n = 1, \dots, N$ compute SfM: $\tilde{w}_n \simeq M_n S_n$.
 - 7: Build \tilde{M} and \tilde{S} , as in equations (8) and (9).
 - 8: **if** *iterations* > $3 \times \mathcal{X}$ or sequence corrected **then**
 - 9: *done* = 1.
 - 10: **else**
 - 11: Compute \hat{S} via sparse minimization, equation (11).
 - 12: Compute $\tilde{w} = \tilde{M}\hat{S}$ and $\hat{w} = \tilde{M}\hat{S}$ and normalize them.
 - 13: \forall points $p = 1, \dots, P$ and \forall frames $f = 1, \dots, F$ compute 2D distance $D(p, f)$ between \tilde{w} and \hat{w} .
 - 14: Find $p1 = \max_p(D(f, p)) \forall f = 1 \dots F$.
 - 15: Find $p2 = \max_p(\sum_{f=1}^F D(f, p)/F)$.
 - 16: Remove $p1$ and $p2$ (if $p2 \neq p1$) from \bar{w} .
 - 17: **end if**
 - 18: **until** *done* == 1
-

Notice that the main three algorithms involved in our proposal, MS, SfM and sparse minimization, can be seen as three generic tools. In the following sections we will describe more in detail our specific implementation and the algorithms we have chosen for the SfM and the sparse minimization steps.

4.2. Structure from Motion step

For the SfM step we decided to use a Bilinear Augmented Lagrangian Multipliers (BALM) [6] method which has the property of enforcing exact metric constraints and the ability to deal with a high ratio of missing data. This last feature can be exploited for a further extension of our optimization algorithm. Besides, BALM is a generic bilinear optimizer, thus it could be used also for non-rigid and articulated SfM.

4.3. Sparse minimization step

For the sparse minimization step we decided to use SALSA [1] for its faster performances compared with other sparse optimization tools. SALSA requires three main parameters to be tuned: ϵ , μ_1 and μ_2 .

The parameter ϵ defines the constraint of the optimization problem, equation (11). As explained in [1], ϵ has to be proportional to the noise variance, therefore, it can be automated once the noise variance of the tracked points has been estimated. Nevertheless, ϵ plays a fundamental role in the minimization problem, as too small an ϵ may cut away the real solution to the problem. Therefore, it is safer to provide an upper bound of the noise level rather than estimate it exactly. In our experiments we let SALSA estimate ϵ given an upper bound noise variance that is fixed to the equivalent of 5 pixels in the normalized space. We believe that 5 pixels is a reasonable limit after which the whole algorithm may become unreliable.

The other two parameters, μ_1 and μ_2 , are the weights of quadratic constraints as a result of the variable splitting algorithm used in SALSA. As there is no a-priori knowledge about these two parameter values we empirically found a combination that allows for good performances without the need to change them for each sequence. We performed extensive tests with 421 different combinations of μ_1 and μ_2 on our synthetic database and all the results that we present in this paper have been obtained with the same fixed combination: $\mu_1 = 0.1$ and $\mu_2 = 0.01$.

5. Experiments

The experiments were first performed on synthetic video sequences, followed by real sequences from the Hopkins155 database [14].

The synthetic sequences contain cubes, with 56 tracked features each, that randomly rotate and translate. Specifically, each synthetic sequence is 50 frames long. The whole database is composed by 80 different sequences with a varying number of independent motions and noise. In more detail, we tested 5 randomly generated motions with 2, 3, 4 and 5 independently moving objects (cubes) obtaining a total of 20 sequences. Gaussian noise with standard deviation of 0, 0.5, 1 and 1.5 pixels was added at each sequence. An example of a synthetic frame (which, for clarity, is plotted with just a few tracked features) is shown in Fig. 1(a).

We ran different tests with an increasing number of misclassified points (randomly selected). The basic test was performed with only 1 misclassified point per sequence. In this setup each experiment converged to the correct solution. This allowed us to verify that our algorithm was able to detect the misclassified point given the whole set of points in the sequence. The next tests were performed with a higher number of misclassified points: 1%, 2%, 3%,

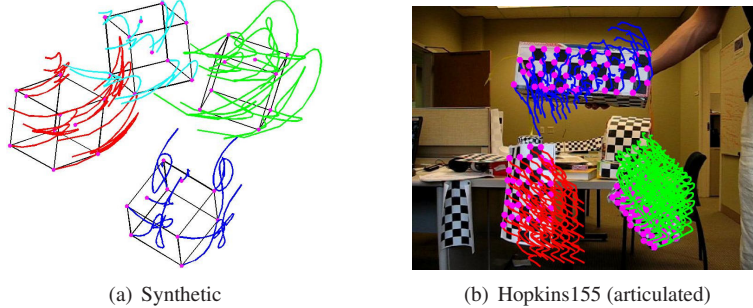


Figure 1. Example of an input frame and some trajectories of a synthetic and a real sequence.

4%, 5% and 10% of the total number of points in each sequence. In Fig. 2 it is possible to observe: (a) the percentage of error-free sequences that resulted from application of the algorithm (i.e. sequences whose misclassified points were all identified and removed, in the plots this is represented by a + symbol), (b) the percentage of identified errors (i.e. how many misclassified points were identified as a percentage of all the original misclassified points, in the plots a \circ symbol), (c) the percentage of false positives (i.e. how many points our algorithm removed even if they were correctly classified, in the plots a \times symbol). The results show that until a misclassification rate of 5% almost all of the sequences are completely corrected. Clearly the higher the noise, the number of motions and the percentage of misclassification, the higher the percentage of false positives tends to be. However, even when some sequences are not completely corrected, the high percentage of misclassified points identified (\circ) shows that our proposal is reliable and robust against noise and an increasing number of motions. For example, even in the case of an initial misclassification of 10% the percentage of detected errors in the segmentation is always above 90%, including the sequences with 5 motions and the highest noise level. To summarize, our algorithm is able to greatly reduce the initial misclassification rate of the input sequences even in the most challenging conditions. In fact, after the application of our algorithm the majority of the sequences become error-free.

We also applied our algorithm (using the same input parameters) to the real sequences of the Hopkins155 database. As we did for the synthetic tests, we first applied our algorithm on the database where only 1 point per sequence was misclassified, then we extended the test using an initial misclassification rate of 1%. The Hopkins155 database (an example of a real frame is shown in Fig. 1(b)) contains 104 checkboard sequences, 38 traffic sequences and 13 other sequences (among which are sequences with articulated motions). As our algorithm assumes independence of the motions, the Hopkins155 database is a very challenging test. Nevertheless, with 1 misclassified point we are able to correct 77 sequences. When the misclassification rate rises

to 1% we are able to completely correct 61 sequences. It should be noted that for remaining sequences, which are not completely corrected, the initial misclassification rate is at least halved. This suggests that allowing the algorithm to perform more iterations could increase the total number of error-free sequences. Another way for improving the performance of our algorithm would be to tune the SALSA parameters μ_1 and μ_2 .

As far as the computational time is concerned, the majority of the resource is spent for the sparse minimization. On the whole Hopkins155 database with 1 error per sequence the algorithm required approximately 3 hours (Matlab implementation on Quad-Core @ 2.4GHz, with 16 GB RAM).

6. Conclusions

In this paper we have presented a new approach to perform motion segmentation and 3D reconstruction of 2D points extracted from an uncalibrated image sequence. Our method takes advantage of an initial segmentation to iteratively correct the misclassified points and compute the multi-body 3D structure of the analyzed scene. The proposed iterative bilinear optimization first enforces rotation metric constraints on the camera matrices, and then optimizes the reprojection error while maintaining the sparsity of the 3D shape matrix. Our proposal can be applied to any initialization provided by a standard MS algorithm.

The experiments on synthetic sequences and real data from the Hopkins155 database have shown the validity of our approach which was able to correct different amounts of segmentation errors. It should be noted that our approach does not require any parameter apart from the μ_1 and μ_2 that are needed by the SALSA sparse optimization tool, which were fixed for all of the experiments.

Future work will involve the study of a better automatic stopping condition to our iterative algorithm. Moreover, at present the points identified as errors are removed from the trajectory matrix, however, we are developing a re-classification strategy, which will allow reassignment of the removed points to the correct motion.

Acknowledgement. Spanish Ministry of Science projects

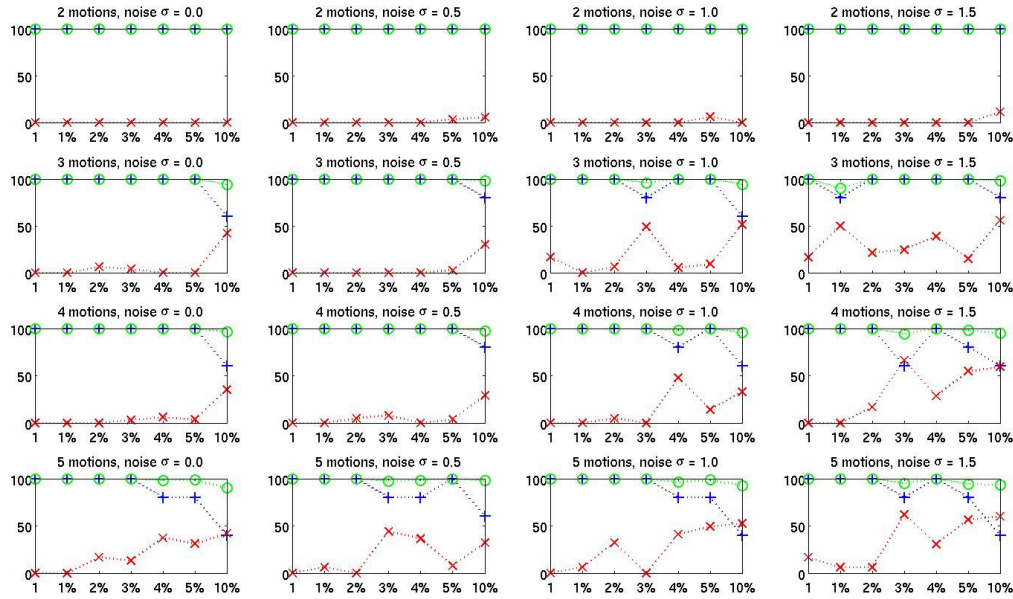


Figure 2. Results of our algorithm applied to the synthetic database. On the x -axis the initial amount of misclassified points. $+$ is the percentage of sequences whose segmentation was completely corrected. o is the percentage of the removed misclassified points over all the misclassified points. \times is the amount of false positive points as a percentage of all the removed points.

DPI2007-66796-C03-02 and DPI2008-06548-C03-03/DPI. L. Zappella is supported by the Catalan government scholarship 2009FI.B1 00068.

References

- [1] M. Afonso, J. Bioucas-Dias, and M. Figueiredo. An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Trans. on Image Proc.*, In Press. 4
- [2] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Proc. CVPR IEEE*, pages 690–696, June 2000. 1
- [3] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Jour. Scientific Comput.*, 20:33–61, 1998. 3
- [4] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(11):1098–1104, 1996. 1
- [5] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *IEEE I. Conf. Comp. Vis.*, pages 1071–1076, June 1995. 1
- [6] A. Del Bue, J. Xavier, L. Agapito, and M. Paladini. Bilinear factorization via augmented lagrange multipliers. In *ECCV*, volume 6314 of *Lect. Notes Comput. Sc.*, pages 283–296. Springer, 2010. 4
- [7] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Proc. CVPR IEEE*, volume 0, pages 2790–2797, 2009. 1
- [8] A. W. Fitzgibbon and A. Zisserman. Multibody structure and motion: 3-d reconstruction of independently moving objects. In *Lect. Notes Comput. Sc.*, pages 891–906, 2000. 2
- [9] F. Lauer and C. Schnrr. Spectral clustering of linear subspaces for motion segmentation. In *IEEE I. Conf. Comp. Vis.*, pages 678–685, 2009. 1
- [10] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. In *Lect. Notes Comput. Sc.*, volume 2, pages 97–108, 1994. 1
- [11] Schindler, Konrad, Suter, and David. Two-view multibody structure-and-motion with outliers through model selection. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(6):983–995, 2006. 2
- [12] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *Int. J. Comput. Vision.*, 9(2):137–154, 1992. 1
- [13] B. Triggs. Factorization methods for projective structure and motion. In *Proc. CVPR IEEE*, pages 845–851, 1996. 1
- [14] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. *Proc. CVPR IEEE*, pages 1–8, 2007. 1, 4
- [15] R. Vidal and R. Hartley. Three-view multibody structure from motion. *IEEE Trans. Pattern Anal. Machine Intell.*, 30(2):214–227, 2008. 2
- [16] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *Int. J. Comput. Vision*, 68(1):7–25, 2006. 2
- [17] L. Zappella, X. Lladó, E. Provenzi, and J. Salvi. Enhanced local subspace affinity for feature-based motion segmentation. *Pattern Recognition*, 44:454–470, 2011. 1
- [18] L. Zappella, E. Provenzi, X. Lladó, and J. Salvi. Adaptive motion segmentation algorithm based on the principal angles configuration. In *ACCV*, 2010. 1