Pattern Recognition 44 (2011) 454–470

Contents lists available at ScienceDirect

Pattern Recognition

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

Enhanced Local Subspace Affinity for feature-based motion segmentation

L. Zappella [a,*], X. Lladó [a], E. Provenzi [b], J. Salvi [a]

[a] Institut d'Informàtica i Aplicacions, Universitat de Girona, Girona, Spain
[b] Departamento de Tecnologías de la Información y las Comunicaciones, Universitat Pompeu Fabra, Barcelona, Spain

ARTICLE INFO

Article history:
Received 9 July 2009
Received in revised form
22 July 2010
Accepted 10 August 2010

Keywords:
Motion segmentation
Manifold clustering
Model selection
Cluster number estimation

ABSTRACT

We present a new motion segmentation algorithm: the Enhanced Local Subspace Affinity (ELSA). Unlike Local Subspace Affinity, ELSA is robust in a variety of conditions even without manual tuning of its parameters. This result is achieved thanks to two improvements. The first is a new model selection technique for the estimation of the trajectory matrix rank. The second is an estimation of the number of motions based on the analysis of the eigenvalue spectrum of the Symmetric Normalized Laplacian matrix. Results using the Hopkins155 database and synthetic sequences are presented and compared with state of the art techniques.

1. Introduction

Motion segmentation aims to identify moving objects in a video sequence. It is a key step for many computer vision tasks such as robotics, inspection, metrology, video surveillance, video indexing, traffic monitoring, structure from motion, and many other applications. The importance of motion segmentation is evident from reviewing its vast literature. However, the fact that it is still considered a "hot" topic also testifies that there are many problems that have not yet been solved.

Based on their main underlying technique, motion segmentation strategies could be classified into the following groups: image difference, *statistical*, optical flow, wavelets, layers, and manifold clustering.

Image difference: image difference techniques are some of the simplest and most used for detecting changes. They consist in thresholding the pixel-wise intensity difference of two consecutive frames [1,2]. Despite their simplicity they provide good results being able to deal with occlusions, multiple objects, independent motions, non-rigid, and articulated objects. The main problems of these techniques are the high sensitivity to noise and to light changes, and the difficulty to deal with moving cameras and temporary stopping, which is the ability to deal with objects that may stop temporarily and hence be mistaken as background.

Statistical: statistical theory is widely used in motion segmentation. Common statistical frameworks applied to motion segmentation are Maximum A Posteriori Probability [3,4], Particle Filter [5] and Expectation Maximization [6]. Statistical approaches use mainly dense-based representations; this means that each pixel is classified, in contrast to feature-based representation techniques that classify only some selected features. This group of techniques works well with multiple motions and can deal with occlusions and temporary stopping. In general they are robust, as long as the model reflects the actual situation, but they degrade quickly as the model fails to represent reality. Moreover, most of the statistical approaches require some kind of a priori knowledge.

Wavelets: these methods exploit the ability of wavelets to analyse the different frequency components of the images, and then study each component with a resolution matched to its scale [7,8]. Wavelet solutions seem to provide overall good results but are limited to simple cases (such as translations in front of the camera).

Optical flow (*OF*): OF can be defined as the apparent motion of brightness patterns in an image sequence. Like image difference, OF is an old concept greatly exploited in computer vision and used also for motion segmentation [9–11]. OF, theoretically, can provide useful information to segment the motions. However, OF alone cannot deal with occlusions or temporary stopping. Moreover, in its simple version it is very sensitive to noise and light changes.

Layers: the key idea of layer based techniques is to divide the image into layers with uniform motion. Furthermore, each layer is associated with a depth level and a "transparency" level that determines the behaviour of the layers in the event of overlaps. Recently, new interest has arisen for this technique [12,13]. Layers are probably the most natural solution for occlusions. The main drawback is the level of complexity of these algorithms and the typically large number of parameters that have to be tuned.

* Corresponding author.
E-mail address: zappella@eia.udg.edu (L. Zappella).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.08.015

Manifold clustering: these techniques aim at defining a low-dimensional embedding of the data points (trajectories in motion segmentation) that preserves some properties of the high-dimensional data set, such as geodesic distance or local relationships. This class of solutions, usually based on feature points, can easily tackle temporary stopping and provides overall good results. A common drawback to all these techniques is that they perform very well when the assumptions of rigidity and independence of the motions are respected, but problems arise when one of these assumptions fails. The intense work done on manifold clustering for motion segmentation led to satisfactory performances, which make these solutions appealing. However, more work has to be done in order to have a motion segmentation algorithm that is completely automatic and independent from a priori knowledge.

In this paper we present the Enhanced Local Subspace Affinity (ELSA), a motion segmentation algorithm based on manifold clustering. ELSA is inspired by the Local Subspace Affinity (LSA) [14,15] technique introduced by Yan and Pollefeys. In contrast to LSA, ELSA is able to automatically tune its most sensitive parameter and it does not require previous knowledge of the number of motions. Such a result is achieved thanks to two improvements. The first is a new model selection technique called Enhanced Model Selection (EMS). EMS is able to adjust automatically to different noise conditions and different number of motions. A preliminary version of EMS was first presented in [16]. The second improvement introduced in this paper is an estimation of the number of motions based on finding, dynamically, a threshold for the eigenvalue spectrum of the Symmetric Normalized Laplacian matrix. By doing so, the final segmentation can be achieved by any spectral clustering algorithm without requiring any a priori knowledge about the number of motions. For all the other parameters we propose a fixed value that we use in all our experiments, showing that even without tuning them, they lead to good results in most of the cases. If one wants, all the parameters could be manually tuned in order to achieve even better performance but we were not interested in obtaining "the best result" but rather in having a good behaviour in the majority of cases without requiring manual tuning. A full source code implementation of ELSA is available at http://eia.udg.edu/~zappella.

The rest of the paper is structured as follows. In Section 2 we review the state of the art focusing on manifold clustering techniques. In particular, in Section 3, LSA [14,15] is analysed in detail. Our new proposed algorithm ELSA is presented in Section 4. The experimental results, shown in Section 5, are computed on the Hopkins155[1] database [17], which is a reference database for motion segmentation. We use also noise perturbed versions of the Hopkins155 database in order to test the behaviour of our algorithm with different noise levels. Moreover, to test the behaviour with more than 3 motions we use a synthetic database with 4 and 5 motions and controlled noise conditions. The results of ELSA are compared with LSA in order to test the new EMS. Furthermore, ELSA is compared with the recently proposed Agglomerative Lossy Compression (ALC) algorithm [18] which is, to the best of our knowledge, the best performing manifold clustering algorithm without a priori knowledge. In Section 6 conclusions are drawn, and future work is discussed.

2. Manifold clustering state of the art

This section provides a complete review on manifold clustering algorithms applied to motion segmentation. A comprehensive review on different motion segmentation techniques can be found in [19].

In general, manifold clustering solutions consist of clustering data that has common properties by, for example, fitting a set of hyperplanes to the data. Frequently, when the ambient space is very big they project the original data set into a smaller space. Most solutions assume an affine camera model, however, it is possible to extend them to the projective case by an iterative process as shown in [20].

Manifold clustering comprises a large number of different techniques, and a further classification can help in giving some order. Manifold clustering can be divided into: iterative solutions, *statistical* solutions, Agglomerate Lossy Compression (ALC), factorization solutions, and subspace estimation solutions. The techniques revised here are summarised in Table 1, which offers a compact at-a-glance overview of the manifold clustering category.

An *iterative* solution is presented in [21] where the RANdom SAmple Consensus (RANSAC) algorithm is used. RANSAC tries to fit a model to the data by randomly sampling n points, computing the residual of each point to the model and counting the number of inliers, which are those points whose residual is below a threshold. The procedure is repeated until the number of inliers is above a threshold, or enough samples have been drawn. Another iterative algorithm called "K-Subspaces Clustering" is presented in [22] for face clustering, however, the same idea could be adopted to solve the motion segmentation problem. K-Subspaces can be seen as a variant of K-means. K-Subspaces iteratively assigns points to the nearest subspace, updating each subspace by computing the new basis that minimises the sum of the square distances to all the points of that cluster. The algorithm ends after a predefined number of iterations. With a different strategy, the authors of [23] propose a subspace segmentation algorithm based on a Grassmannian minimisation approach. This technique consists in estimating the subspace with the maximum consensus (MCS), defined as the maximum number of data points that are inside the subspace. Then, the algorithm is recursively applied to the data inside the subspace in order to look for smaller subspaces included within it. The MCS is efficiently built by a Grassmannian minimisation problem.

Iterative solutions are in general robust to noise and outliers, and they provide good solutions if the number of clusters and the dimensions of the subspaces are known. This a priori knowledge can be clearly seen as their limitation as this information is not always available. Moreover, they require an initial estimation and are not robust against bad initializations and hence, are not guaranteed to converge.

The authors of [24] use a *statistical* framework for detecting degeneracies of a geometric model. They use the geometric Akaikes information criterion (AIC) defined in [25] in order to evaluate whether two clouds of points should be merged or not. Another statistic based technique is presented in [26]. This work analyses the geometric structure of the degeneracy of the motion model, and suggests a multi-stage unsupervised learning scheme, first using the degenerate motion model and then using the general 3D motion model. The authors of [27] extend the Expectation Maximization algorithm proposed in [28] for the single object case, to multiple motions and missing data. In [29] the same authors further extend the method incorporating non-motion cues (such as spatial coherence) into the M-step of the algorithm.

Statistical solutions have more or less the same strength and weaknesses of iterative techniques. They can be robust against noise whenever the statistic model is built taking the noise explicitly into account. However, when noise is not considered, or is not properly modeled, their performances rapidly degenerate. As previously mentioned statistical approaches are robust as long as the model reflects the actual situation.

A completely different idea is the basis of [18], which uses the *Agglomerative Lossy Compression* (ALC) algorithm [30].

[1] Available at http://www.vision.jhu.edu.

Table 1
Summary of the examined techniques with respect to the most important attributes.

Manifold clustering								
Iterative								
Fishler et al. [21]	F			✓	✓	I	RA	C
Ho et al. [22]	F			✓	ON	✓	✓	CD
da Silva et al. [23]	F			✓	O	✓	✓	X
Statistical								
Kanatani et al. [24]	F			✓	OI	I	R	
Sugaya et al. [26]	F			✓	✓	I	R	C
Gruber et al. [27]	F	✓		✓	ON	I	R	X
Gruber et al. [29]	F	✓	✓	✓	ON	I	R	X
ALC								
Rao et al. [18]	F	✓		✓	OI	I	R	
Factorization								
Costeira et al. [32]	F			✓	OI	I	R	
Ichimura et al. [33]	F			✓	OI	I	R	
Zelnik-Manor et al. [34]	F			✓	✓	✓	RA	CD
Zhou et al. 2003 [35]	F	✓		✓	✓	✓	RA	CD
Subspaces								
Vidal et al. [36]	F	✓		✓	I	✓	R	C
Yan et al. [14,15]	F			✓	OI	✓	✓	CDX
Julià et al. [38]	F	✓		✓	OI	✓	R	C
Goh et al. [40]	F			✓	OI	✓	R	CD
Vidal et al. [37]	F	✓		✓	OI	✓	R	C
Goh et al. [42]	F			✓	OI	✓	R	CD
Chen et al. [39]	F			✓	ON	✓	✓	CD
Elhamifar et al. [43]	F	✓		✓	✓	✓	✓	D
Our new proposal: ELSA	F			✓	✓	✓	✓	
Features (F)/dense (D)								
Occlusion or missing data								
Spatial continuity								
Temporary stopping								
Robustness (O, Outliers; N, noise; I, Initialization; ✓ all)								
Dependency (I, independent; D, dependent; ✓ all)								
Kind (R, rigid; N, non-rigid; A, articulated; ✓ all)								
Prior knowledge (C; clusters number; D, subspace dimension; X, other; T, training)								

In *feature-based* methods, the objects are represented by a limited number of salient points, while *dense-based* methods compute a pixel-wise motion. *Spatial continuity* means that the information provided by the neighbourhood of a point is taken into account. *Temporary stopping* is the ability to deal with objects that stop temporarily. *Missing data* are the ability to deal with the lack of information caused by occlusions or appearing and disappearing features. *Robustness* groups together the ability to deal with all the problems caused by noise, by initialization (for simplicity whenever the initialization is not required the algorithm is considered robust against initialization) and outliers. *Dependency* is the ability to deal with independent motions (the pairwise intersection of the motion subspaces is the zero vector), and dependent motions (the pairwise intersection of the motion subspaces is not empty). *Kind* is the ability to deal with rigid motion (the trajectories generated by the points of a rigid object form a linear subspace of dimensions no more than 4 [31]), non-rigid (the trajectories generated by the points of a non-rigid object can be approximated by a combination of k weighted key basis shapes, and they form a linear subspace of dimension no more than $3k+1$ [44,45]), and articulated (when it is composed of two dependent motions connected by a link). *Prior knowledge* summarises the fact that some sort of knowledge is required: number of motions, dimension of the generated subspaces, any other form of a priori knowledge, or training step.

This technique consists in minimizing a cost function by grouping trajectories. Roughly speaking, the cost function is equal to the amount of information required to represent each manifold, given a solution for the segmentation.

ALC provides a connection between coding theory and space representation. It performs extremely well with a variety of motions. However, it suffers from the curse of dimensionality problem. Furthermore, the algorithm depends on a parameter that has to be tuned per each sequence depending on the number of motions and the amount of noise. Although the tuning can be automated by trying many different values and choosing the solution with the lowest cost, this process is highly time-consuming.

Factorization techniques are based on the approach introduced by Tomasi and Kanade in [31] to recover structure and motion using features tracked through a sequence of images. In [32] the framework of Tomasi and Kanade is used to build a symmetric matrix, called "shape interaction matrix", whose size is equal to the number of tracked features. The shape interaction matrix has, among other properties, zero entries if the two indexes represent features belonging to different motions and non-zero otherwise. Hence, the algorithm focuses on finding the permutation of the shape interaction matrix that gives a block diagonal structure. In [33] the authors, having estimated the rank r of the trajectory matrix, perform the QR decomposition of the shape interaction matrix in order to select some convenient features and perform an initial segmentation among those. Finally, the remaining features are segmented by using the orthogonal projection matrix. The two previous factorization techniques assume that the motions are independent. In [34] the authors study the degeneracy in case of dependent motion. They propose a factorization method that consists in building an affinity matrix by using only the dominant eigenvector and estimating the rank of the trajectory matrix by studying the ratios of the eigenvalues. In [35] a hierarchical factorization method for recovering articulated hand motion under

weak perspective projection is presented. They consider each part of the articulated object as independent, and they use any technique able to deal with missing data to fill the gaps in the trajectory matrix. In a second step, they guarantee that the extremities of consecutive objects are linked in the recovered motion.

Factorization techniques are based on a very simple and elegant framework. However, they are particularly sensitive to noise and cannot deal very well with outliers. Moreover, most of the techniques assume rigid and independent motions.

The last category of manifold clustering is the *subspace estimation* techniques. The work presented in [36] is one of them. First, exploiting the fact that trajectories of rigid and independent motion generate subspaces at maximum of dimension 4, they project these trajectories onto a five-dimensional space using PowerFactorization. Afterwards, the Generalized Principal Component Analysis (GPCA) is used to fit a polynomial of degree n, where n is the number of subspaces (i.e. the number of motions), to the data and estimate the basis of the subspaces using the derivatives of the polynomial. More recently, the same authors extend in [37] the above-mentioned framework using RANSAC to perform the space projection in order to deal with outliers. Another well-known technique is the Local Subspace Affinity (LSA) [14,15]. LSA is able to deal with different type of motion: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. The key idea is that different motion trajectories lie in different subspaces. Thus, the subspaces are estimated and an affinity matrix is built using a measure based on principal angles. The final segmentation is obtained by clustering the affinity matrix. One of the main limitation of LSA is that a full trajectory matrix without missing data is assumed. In [38] a technique similar to LSA is presented in order to deal with missing data. The idea is to fill the missing data so that the final matrix will have a frequency spectrum similar to the one of the input matrix. When a full trajectory matrix is obtained an affinity matrix is built and a cluster algorithm based on normalized cuts is applied in order to provide the segmentation. In [39] the authors propose a generalization of LSA called Spectral Curvature Clustering (SCC). SCC differs from LSA for two main reasons. The first reason is related to the affinity measure, as SCC uses polar curvature while LSA uses principal angles. The second reason is how they select the points used to estimate the local subspaces: SCC uses an iterative random sampling, while LSA uses the nearest neighbours. A completely different strategy is presented in [40] where, starting from the Locally Linear Embedding algorithm [41], they propose the Locally Linear Manifold Clustering algorithm (LLMC). With LLMC the authors try to deal with linear and non-linear manifolds. The same authors extended this idea to Riemannian manifolds [42]. They project the data from the Euclidean space to a Riemannian space and reduce the clustering to a central clustering problem. More recently in [43] a new way for describing the subspaces called Sparse Subspace Clustering (SSC) was presented. The authors exploit the fact that each point (in the union of subspaces) can be described with a sparse representation with respect to the dictionary composed of all the points. By using l_1 optimisation, and under mild assumptions, they estimate the subspaces and they build a similarity matrix which is used to obtain the final segmentation by spectral clustering.

Subspace estimation techniques can deal with intersection of the subspaces and do not need any initialization. However, all these techniques suffer from common problems: curse of dimensionality, a weak estimation of the number of motions and of the dimension of the subspaces. The curse of dimensionality is mainly solved in two ways: projection onto smaller subspaces or random sampling. Conversely, estimation of the number of motions and of the subspace dimension remain two practical open issues.

Manifold clustering, specifically the group of subspace estimation, seems to be a good candidate for further investigation. This group of techniques already provides a good performance in terms of segmentation quality and it is naturally extended to structure from motion algorithms. A quick glance at Table 1 indicates that the price to pay for dealing with different kinds of motion and with dependent motions is a higher a priori knowledge (in particular about the dimension of the generated subspaces). Another common limitation that should be overcome is the required knowledge about the number of motions in the scene.

Among the subspace estimation techniques reviewed, the Local Subspace Affinity method proposed by Yan and Pollefeys [14,15] appears a promising approach. As already concluded by Tron and Vidal in [17], for the case of non-missing data LSA performs better than GPCA and a RANSAC based technique. It is able to segment different types of motion (independent, dependent, articulated, rigid, non-rigid, degenerate and non-degenerate) as well as to deal with a limited amount of mismatches in the tracked features. Furthermore, it is based on a relatively simple framework that can be easily extended to include more information and, despite its good performance, there is room for improvement, as explained in the next section.

3. Local Subspace Affinity (LSA)

The general idea of LSA is that trajectories that belong to different motions lie on different subspaces. Thus, the segmentation can be obtained by grouping together all the trajectories that generate similar subspaces. Following this idea, LSA estimates the subspace generated by each trajectory and builds an affinity matrix. The affinity measure between two trajectories is given by the distance between the two generated subspaces. The final segmentation is obtained by clustering the affinity matrix.

3.1. The algorithm

LSA can be summarised in six fundamental steps:

1. build a trajectory matrix $\mathbf{W}_{2f\times p}$, where f is the number of frames of the input sequence and p is the number of tracked feature points;
2. estimate the rank of $\mathbf{W}_{2f\times p}$; this step is accomplished by a model selection (MS) technique inspired by the work of Kanatani [46]:
$$r = \underset{r}{\operatorname{argmin}}\left(\frac{\lambda_{r+1}^2}{\sum_{i=1}^{r}\lambda_i^2} + kr\right), \tag{1}$$
λ_i being the i-th singular value of $\mathbf{W}_{2f\times p}$, and k a parameter that depends on the noise of the tracked point positions: the higher the noise level is, the larger the k should be [14];
3. project every trajectory, which can be seen as a vector in $\mathbb{R}^{2f}$, onto an $\mathbb{R}^r$ unit sphere by singular value decomposition (SVD) and truncation to the first r components of the right singular vectors;
4. exploiting the fact that in the new space (global subspace) most points and their closest neighbours lie in the same subspace, the local subspaces generated by each trajectory, and its nearest neighbours (NNs), are computed; this result can be achieved by SVD, hence the estimation of the local subspace dimension is required again in order to truncate the SVD result; such an estimation can be done by the same model selection technique illustrated in formula (1);
5. compute an affinity matrix $\mathbf{A}$, where the affinity is a function of the distance between the local subspaces in terms of principal angles;

6. cluster $\mathbf{A}$ in order to have the final motion segmentation; any clustering technique could be used, the authors suggest Normalized Cuts [47].

3.2. Problems

LSA provides an elegant solution for the manifold clustering problem, however, it is possible to identify some weaknesses of the algorithm.

The first weakness is that LSA is able to deal only with full trajectories, no missing data are allowed (step 1). This is an important limitation, however, some authors have already pointed out possible solutions [38].

A second weakness is the fact that MS, formula (1), relies on a parameter k which has to be tuned depending on the noise level and the number of motions (steps 2 and 4). Hence, MS requires previous knowledge regarding the amount of noise and the number of motions. Moreover, it is very sensitive to the parameter k, to the extent that Tron and Vidal, in their implementation of LSA [17], claim that they had problems in finding a value for k that was good for all the sequences of the Hopkins155 database. Therefore, in [17] the authors avoided rank estimations and preferred to fix the space size. They chose to fix the global space dimension to $4n$ (step 2), where n is the number of motions. They also fixed the dimension of the local subspaces generated by each trajectory to 4 (step 4). By doing so, only the number of motions is required. However, the ability of LSA to deal with different types of motion is greatly reduced, in fact only rigid and non-degenerate motions ensure a rank equal to 4 [31].

Another problem consists in the fact that Normalized Cuts does not provide a reliable instrument for estimating the number of clusters (in the case of motions segmentation, a cluster is equivalent to a motion). Shi and Malik in [47] suggest to use the Cheeger constant [48] or the cost of the last cut as an indication of whether or not it is worth to continue the cutting process. However, these two constants are highly sensitive to noise and they require the use of thresholds that may change depending on the input sequence. This is the reason why most of the available implementations of Normalized Cuts assume the final number of clusters as known data.

4. Enhanced Local Subspace Affinity (ELSA)

Our proposed ELSA fixes the weaknesses of LSA as it finds automatically a good value for k without requiring any a priori knowledge, and introduces a robust estimation of the number of motions. In Section 4.1 the Enhanced Model Selection technique for rank estimation is presented and in Section 4.2 a robust number of motions estimation is described.

4.1. Enhanced Model Selection (EMS)

EMS automatically finds a good k value for formula (1), so that the amount of noise and the number of motions are not required as a priori knowledge. As formula (1) is used in steps 2 and 4 of LSA, the estimated k for the global space (step 2) is called k_g, while the k for the local subspaces (step 4) is called k_s.

EMS is first applied to step 2 for the rank estimation of the matrix $\mathbf{W}_{2f\times p}$. For the moment, the dimension of the local subspace generated by each trajectory is fixed to 4 (assuming the motions are rigid), the extension of EMS to step 4 is explained later. The key idea of EMS lies in the relationship between the rank of $\mathbf{W}_{2f\times p}$ estimated by MS (formula (1)), and the computed affinity matrix $\mathbf{A}$ (step 5). To offer a pictorial understanding of such a relationship, an example is shown in Fig. 1(a)–(f), where the affinity matrices, obtained after estimating the rank with different k_g values, are shown. A more rigorous explanation will be presented in Section 4.1.1.

The sequence used in this example has three rigid motions (maximum rank is 12). When the rank of $\mathbf{W}_{2f\times p}$ is estimated using an inappropriate k_g value, the affinity matrix does not provide any useful information, as in Fig. 1(a) and (b) (rank overestimated) and Fig. 1(d)–(f) (rank underestimated). The best affinity matrix, visually speaking and knowing how many motions are in the scene, is obtained with $k_g=10^{-7.5}$, which gives an estimated rank of 10, Fig. 1(c).

The relationship between the chosen k_g and the computed affinity matrix clarifies why a wrong choice of k_g can greatly affect the final segmentation. It is crucial for LSA to have a good rank estimation of the trajectory matrix in order to perform a correct segmentation. On the other hand, this relationship is giving some information: by looking at the affinity matrix it is possible to assess the accuracy of the rank estimation. In Section 4.1.1 we analyse why there is such a relationship, and in Section 4.1.2 we explain that, without knowing the number of motions and without assuming any order of the tracked features, the entropy can be used as a measure of the "reliability" of the affinity matrix, and hence of the estimated rank. Moreover, in Section 4.1.3 a speed up algorithm is proposed in order to quickly obtain an affinity matrix with high entropy, avoiding an exhaustive search among a big range of k_g values. Finally, the extension of EMS for the estimation of the size of the local subspaces is presented in Section 4.1.4.

4.1.1. Affinity matrix as a function of the estimated rank

Before getting into details about how the relationship between the affinity matrix and the estimated rank r (and hence the k_g value) could be exploited, let us analyse more deeply the behaviour of the affinity matrix in relation to the estimated rank r. As there are many factors that influence the final affinity matrix it is easier to start with a simplified problem and extend it later to the real case. Assume for the moment that there is no noise in the tracked feature positions, and that the ground truth set of NNs of each trajectory is known. Also, assume that the local subspaces generated by trajectories of different motions are orthogonal.

The affinity between two generic trajectories α and β depends on the distance between their local subspaces, which in LSA is computed through the principal angles. Let us recall that the principal angles between two subspaces $S(\alpha)$ and $S(\beta)$ are defined recursively as a series of angles $0\le\theta_1\le,\ldots,\le\theta_M\le\pi/2$, where $M=\min\{rank(S(\alpha)),rank(S(\beta))\}$ [49]:

$$\cos(\theta_1)=\max_{u\in S(\alpha),v\in S(\beta)} u^Tv=u_1^Tv_1, \tag{2}$$

while

$$\cos(\theta_k)=\max_{u\in S(\alpha),v\in S(\beta)} u^Tv=u_k^Tv_k,\quad \forall k=2,\ldots M, \tag{3}$$

$$\text{s.t.}:\quad \|u\|=\|v\|=1,\quad u^Tu_j=0,\quad v^Tv_j=0\ \ \forall j=1,\ldots,k-1.$$

The vectors $u_1,\ldots,u_k$ and $v_1,\ldots,v_k$ are the principal vectors. In the ideal case and with a perfect rank estimation, the principal angles θ_i between two local subspaces generated by trajectories of the same motion should be 0. On the other hand, when α and β belong to different motions, θ_i should be close to $\pi/2$. Let us now discuss the behavioural trend of the principal angles as a function of r (the rank of the global subspace) in the cases of under and overestimation.

The analysis in the case of underestimation is quite easy to develop. In fact, in this case the components of the lost

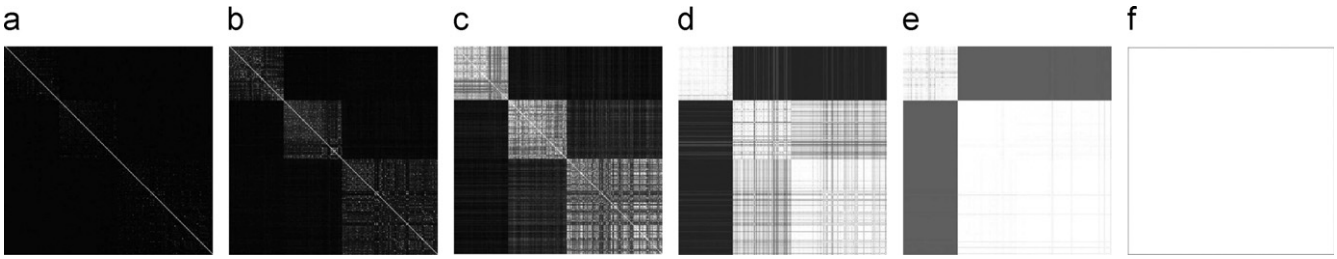

Fig. 1. Affinity matrices computed with different k_g values; real sequence 1R2RC from the Hopkins155 database, theoretical maximum rank of $\mathbf{W}$ is 12; black is minimum affinity, white is maximum affinity and r is the estimated rank. (a) $k_g=10^{-12}$; $r=57$; (b) $k_g=10^{-10}$; $r=21$; (c) $k_g=10^{-7.5}$; $r=10$; (d) $k_g=10^{-6}$; $r=6$; (e) $k_g=10^{-5}$; $r=5$; (f) $k_g=10^{-4}$; $r=4$.

dimensions are projected onto the remaining dimensions, artificially forcing the two local subspaces towards each other to the point when they collapse onto exactly the same local subspace. As a consequence, two local subspaces tend to become closer as r decreases.

On the other hand, the behaviour of principal angles when r is an overestimation is less intuitive. In fact, in [50] it is explained that the problem of computing principal angles and principal vectors when the rank of the global space is overestimated is an ill-posed problem. To the authors' knowledge, the most helpful mathematical result for the case under analysis is presented in [51]. The authors study the probability density function (pdf) of the largest principal angle between two subspaces chosen from a uniform distribution on the Grassmann manifold of p-planes embedded in $\mathbb{R}^n$. They show that, when n is appreciably bigger than p (precisely $n>2p-1$), the pdf is close to zero for small angles and rapidly increases to reach a global maximum in $\pi/2$.

The resemblance between this abstract mathematical situation and the practical case under analysis is represented by the fact that, when the rank is overestimated, the extra component added to the trajectory vectors (in the global space) are sampled from basis vectors of the null space of $\mathbf{W}$. In fact, the projection onto the global space is done as follows. The matrix $\mathbf{W}_{2F\times N}$ is decomposed by SVD as: $\mathbf{W}=\mathbf{UDV}^T$. Hence, if the rank of $\mathbf{W}$ is r_{real}, the first r_{real} columns of $\mathbf{V}$ correspond to the basis of the row space of $\mathbf{W}$ whereas the remaining $N-r_{real}$ columns of $\mathbf{V}$ correspond to the basis of the null space of $\mathbf{W}$. In this new global subspace the first r_{real} components of each row i of $\mathbf{V}$ (for $i=[1,N]$) represent the trajectory i in the global subspace. In Fig. 2(a) the meaning of each column and row of $\mathbf{V}$ is summarised in the case that the estimated rank of $\mathbf{W}$ is $r_{est}=r_{real}$. However, if the estimated rank of $\mathbf{W}$ is $r_{est}>r_{real}$, each trajectory i will be represented by its true r_{real} components (taken from the first r_{real} components of row i of matrix $\mathbf{V}$) plus $r_{est}-r_{real}$ extra components that are taken from row i of the $r_{est}-r_{real}$ basis of the null space of $\mathbf{W}$. These extra components are unrelated to the trajectory i, hence they are random with respect to that trajectory. This second case is summarised in Fig. 2(b). Note that the projection of the trajectories onto the global space is *oblique*, for this reason the components that exceed the real rank are not eliminated and they act as random values.

Finally, note that all of the reasoning holds true even in the case of no noise and it does not involve the selection of the NNs. Hence, when r_{est} is sufficiently big the work of [51] applies to the case under analysis. Therefore, the higher the overestimation, the closer the resemblance to a uniform distribution.

From the result presented in [51], an *overall increasing behaviour* of the principal angles as a function of the estimated rank can be inferred.[2] To support this inference, in Fig. 3(a)–(d) we present the trend of the largest principal angle of synthetic sequences with 2 rigid motions (hence the maximum rank is 8) and no noise. As the test is performed using the first part of the LSA algorithm the nearest neighbours (NNs) are estimated as explained in step 4 of the LSA summary (Section 3.1). For each sequence four angles are compared: two angles between trajectories of the same motion (blue lines) and two angles between trajectories of different motions (red lines). These results are just a few samples of a large number of experiments performed on the whole synthetic database (see Section 5.1 for a detailed description of the 240 synthetic sequences) with 2, 3, 4 and 5 rigid motions and an increasing noise level and on the Hopkins155 database. All the experiments show the same pattern, i.e. when the rank is very small all principal angles tend towards zero, while when the rank is heavily overestimated all of the angles tend towards $\pi/2$. For simplicity, only the largest principal angle is shown in the examples. However, the smaller angles also follow the same trend. These experiments confirm the overall increasing behaviour inferred from [51]. Moreover, from Fig. 3(a) to (d) it is possible to appreciate that when the estimation of the rank is close to the correct rank, then the principal angles between local subspaces generated by trajectories of different motions are higher than those between local subspaces generated by trajectories of the same motion.

From now on, we will refer to the behaviour of the principal angles θ_i with respect to r as the function $\theta_i(r)$. Let us now analyse how this behaviour reflects on the affinity value between two fixed generic trajectories α and β, where the affinity is defined as

$$A(r)=e^{-\sum_{i=1}^{M}\sin^2(\theta_i(r))}. \tag{4}$$

In the ideal case and with a perfect rank estimation, the affinity between trajectories of the same motion is maximum (i.e. 1), whereas the affinity between trajectories of different motions is minimum (i.e. close to 0). Similarly to what was done for the principal angles, it is interesting to understand the global behaviour of the function $A(r)$. In order to do so, we study the first derivative of $A(r)$:

$$\frac{dA(r)}{dr}=-e^{-\sum_{i=1}^{M}\sin^2(\theta_i(r))}\sum_{i=1}^{M}2\sin(\theta_i(r))\cos(\theta_i(r))\theta_i'(r). \tag{5}$$

All of the functions appearing in the derivative (5) are non-negative for all values of r, except for $\theta_i'(r)$ (the first derivative of $\theta_i(r)$). However, it has been shown that $\theta_i(r)$ is overall increasing, so $\theta_i'(r)\geq 0$ for the majority of the values of r. The presence of the minus sign implies that $dA(r)/dr\leq 0$ for the majority of the values of r, i.e. $A(r)$ has an overall decreasing behavior. Specifically, when r is an underestimation all the affinity values tend to the maximum value, whereas when r is an overestimation they tend to the minimum value. Fig. 3(i)–(l) shows the affinity values of the same pairs of trajectories used in Fig. 3(a)–(d), confirming the results of the analysis just performed.

[2] To the authors' knowledge, no information is known about the precise analytical behaviour of such functions.

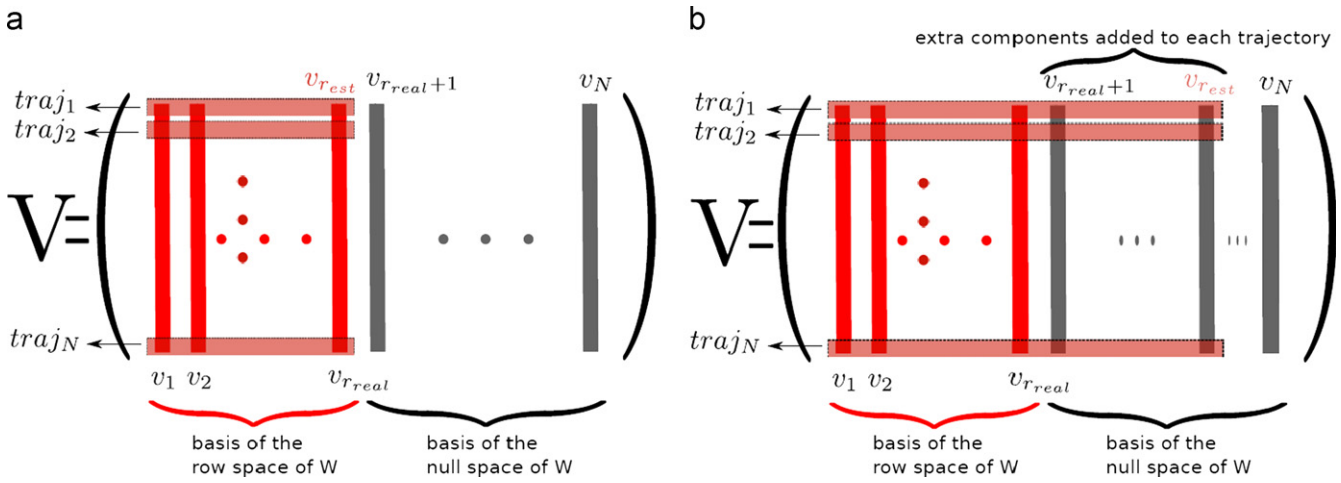

Fig. 2. Pictorial description of what happens when the rank of the global space is estimated. $traj_i$ stands for the ith trajectory. (a) Ideal case; (b) real case.

So far we have considered the case without noise and with perfectly orthogonal local subspaces. The effect of the presence of noise and the estimation of the NNs is that there may be oscillations in the functions of the principal angles (as in Fig. 3(e)–(h)), and so, potentially, also in the affinity functions. However, we will show later in this subsection why this, in turn, does not lead to big oscillations of the affinity values (as in Fig. 3(m)–(p)).

The last simplification was to consider orthogonal local subspaces. Usually, in real sequences the local subspaces are not perfectly orthogonal. The effect of non-perfect orthogonality is that, even in the overestimation cases, some pairs of trajectories may have low affinity but not exactly equal to zero. The effects of non-perfect orthogonality and of the estimation of the NNs can be seen in Fig. 3(a)–(d): despite the fact that there is no noise, it is possible to notice small oscillations in the $\theta_i(r)$.

Therefore, the main consequence of moving from an ideal to a real situation is that the $\theta_i(r)$ may have wider oscillations. However, such oscillations rarely lead to oscillations of the affinity values. In fact, the oscillations of the principal angles may be compensated by the sum involved in the computation of the affinity, formula (4) (especially if the oscillation affects one of the smallest angles). Moreover, the highly non-linear behaviour of the decreasing exponential used to define the affinity, tends to smooth the effect of small changes. Even in the worst case scenario, it is very unlikely that all the affinities between the pairs of trajectories oscillate in correspondence to the same estimation of the rank. Hence, it is highly probable that the trend of $A(r)$ remains overall decreasing. Let us stress that Fig. 3(m)–(p) testify that, even when the presence of noise induces considerable oscillations in the $\theta_i(r)$, the affinities are not dissimilar to those of the case without noise (Fig. 3(i)–(l)).

Summarising, even without the assumptions made at the beginning, the affinity between every pair of trajectories is maximum when the rank of the global subspace is highly underestimated and is minimum when the rank is highly overestimated. In between, the affinities have a decreasing trend. Specific pairs may present oscillations, but the majority of the affinities remain, overall, decreasing functions of r.

4.1.2. How to choose a good affinity matrix

Now that the relationship between the affinity and the estimated rank has been clarified, it is necessary to find a measure of the "reliability" of the affinity matrix. The ideal criterion for the selection of the affinity would be to choose the one that minimises the final misclassification rate. However, the ground truth of the segmentation is not always known. In real cases a convenient criterion could be to choose a high contrasted affinity matrix, because the higher the contrast the higher the quantity of information that can be used to compare the trajectories. A well known measure of contrast, and of quantity of information, is the entropy [52]:

$$E(\mathbf{A}(r)) = -\sum_{i=0}^{1} h_{\mathbf{A}(r)}(i)\log_2(h_{\mathbf{A}(r)}(i)), \tag{6}$$

where $h_{\mathbf{A}(r)}(i)$ is the histogram count in the bin i (in our experiments we use 256 bins). As stated in the previous section, when r is an underestimation, all of the affinities tend to be clustered around 1, leading to a very low entropy value. As r approaches the correct rank, the affinities between trajectories of the same motion tend to diverge from those between trajectories of different motions, hence the entropy increases. Note that it is not possible to exclude that theoretically the entropy function can have oscillations, but regardless we are more interested in its overall behaviour. The more r is increased, becoming an overestimation, the more the affinities tend to converge around 0, consequently the entropy decreases again. Fig. 3(q)–(t) shows the trend of the entropy for the same synthetic sequences used to show $\theta_i(r)$ (Fig. 3(e)–(h)) and $A(r)$ (Fig. 3(m)–(p)) in the presence of noise. Similarly, Fig. 4 shows the entropy trend of the real sequence 1R2RC (Hopkins155 database) used to compute the affinity matrices shown in Fig. 1.

Notice that the entropy would not be maximum in the case of a perfect affinity matrix with only two values (maximum and minimum). However, such a situation is extremely rare in real sequences. Also in synthetic sequences with no noise a perfect affinity matrix is obtained only when the motions are completely independent. In practical cases, without having a clear indication about which affinity matrix is the most suitable, the highest entropy choice has the interesting property of discarding uniform (and thus useless) affinity matrices. In the examples shown in Fig. 3(q)–(t) the highest entropy always corresponds to a rank value where the affinities between trajectories of the same motion are higher than those between trajectories of different motions (Fig. 3(m)–(p)). In the example of Fig. 4 the maximum corresponds to the affinity matrix of Fig. 1(c), which is the one corresponding to the rank estimation closest to the real rank. We call the Enhanced Model Selection (EMS) this new way of estimating the rank as the one that leads to the affinity matrix with the maximum entropy.

As stated before, it is not possible to ensure that the affinity matrix with the maximum entropy corresponds to a perfect

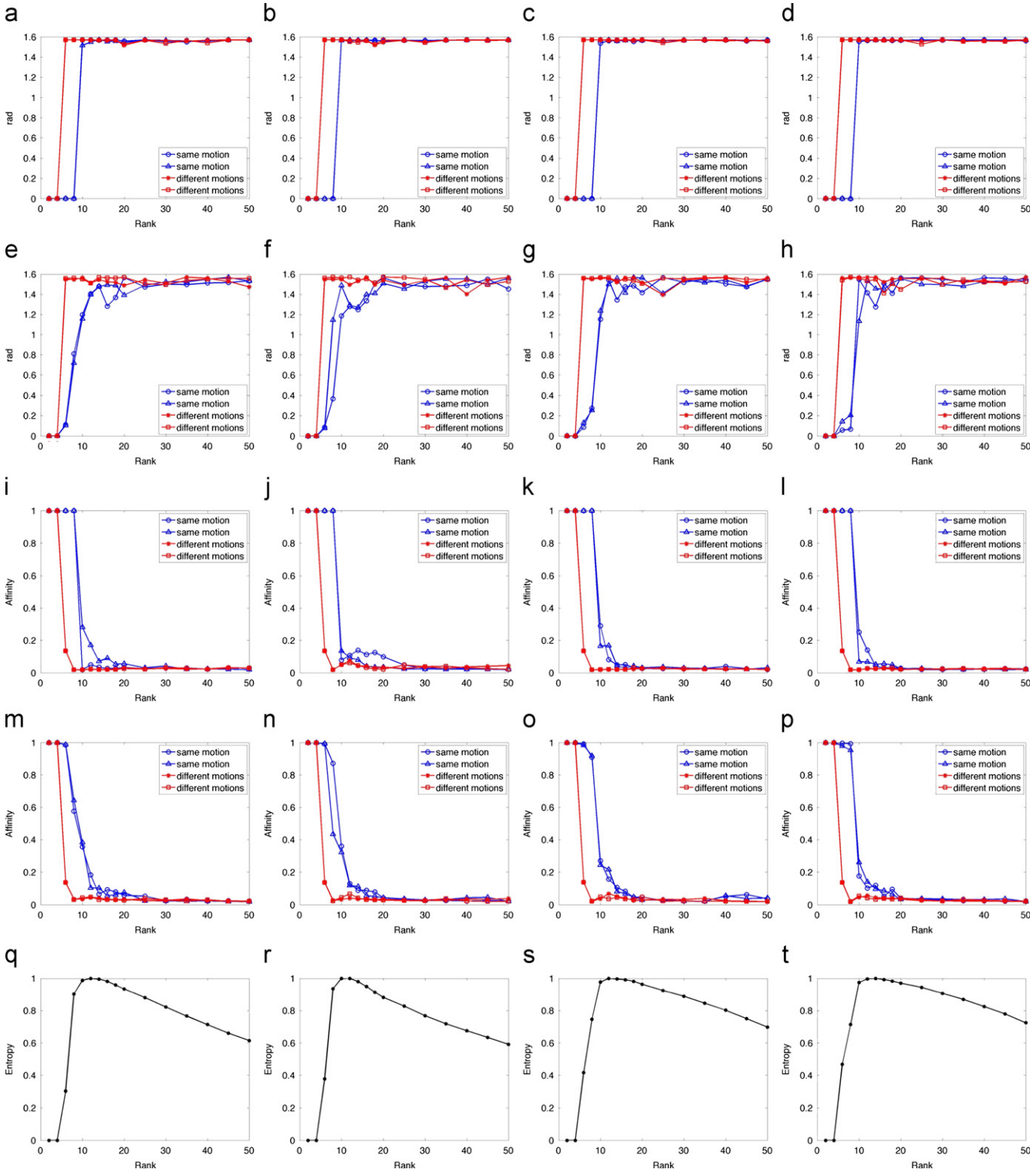

Fig. 3. Trend of the largest principal angle ((a)–(h)), and of the affinity ((i)–(p)), between two pairs of trajectories of the same motion (blue triangles and circles) and two pairs of trajectories of different motions (red asterisks and squares). The trajectories are randomly taken from synthetic sequences (first and third row sequences with no noise, second and fourth rows of same sequences with Gaussian noise $\sigma = 0.5$) with two rigid motions, hence the maximum rank is 8 (see Section 5.1 for more details about the synthetic database; NNs are estimated). Last row shows the entropy trend of $\mathbf{A}(r)$ related to the sequences with Gaussian noise $\sigma = 0.5$. (a) $\theta_1(r)$ seqA $\sigma = 0$; (b) $\theta_1(r)$ seqB $\sigma = 0$; (c) $\theta_1(r)$ seqC $\sigma = 0$; (d) $\theta_1(r)$ seqD $\sigma = 0$; (e) $\theta_1(r)$ seqA $\sigma = 0.5$; (f) $\theta_1(r)$ seqB $\sigma = 0.5$; (g) $\theta_1(r)$ seqC $\sigma = 0.5$; (h) $\theta_1(r)$ seqD $\sigma = 0.5$; (i) $A(r)$ seqA $\sigma = 0$; (j) $A(r)$ seqB $\sigma = 0$; (k) $A(r)$ seqC $\sigma = 0$; (l) $A(r)$ seqD $\sigma = 0$; (m) $A(r)$ seqA $\sigma = 0.5$; (n) $A(r)$ seqB $\sigma = 0.5$; (o) $A(r)$ seqC $\sigma = 0.5$; (p) $A(r)$ seqD $\sigma = 0.5$; (q) $E(\mathbb{A}(r))$ seqA; (r) $E(\mathbb{A}(r))$ seqB; (s) $E(\mathbb{A}(r))$ seqC; (t) $E(\mathbb{A}(r))$ seqD. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

estimation of the rank. Although is expected to provide a good estimation of the rank and contain enough information for a successful clustering step, the rank could be slightly over-estimated or underestimated. Clearly the perfect estimation is the most desirable outcome, however, even a small overestimation is acceptable. The least desirable outcome would be an

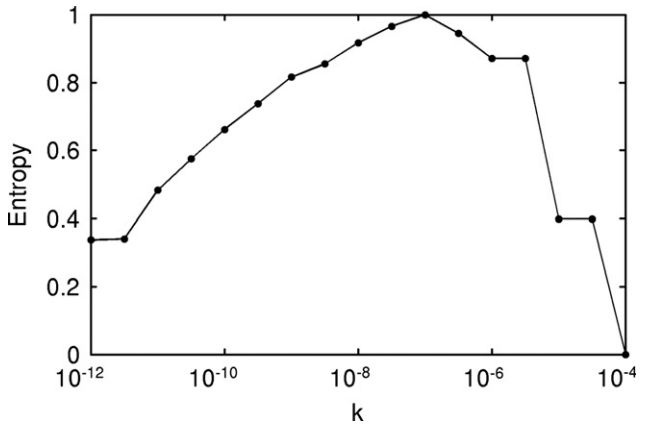

Fig. 4. Example of the entropy trend experienced with all the sequences used in the experiments. This specific example refers to the same sequence used to show the affinity matrices in Fig. 1.

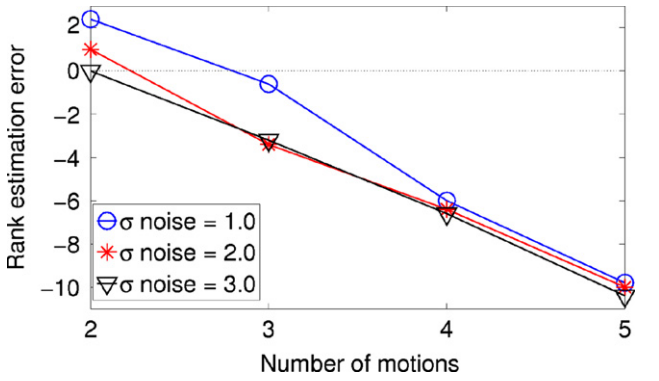

Fig. 5. Error of EMS rank estimation with different number of motions and noise levels. Each point is given averaging the errors over 10 different sequences.

underestimation as this corresponds to cutting important information. In Fig. 3(i)–(p) it can be appreciated how all the affinities go to 1 very quickly when the rank is underestimated.

In order to prevent a possible underestimation, or reduce its effects, it may be safer to increase the estimated rank by a small amount. Our experiments have shown that there is a correlation between the number of motions (or the amount of noise) and the position of the maximum entropy: the greater the number of motions (or the higher the noise) the lower the estimation of the rank. In Fig. 5 the error of the EMS rank estimation is shown in relation to the number of motions and the noise level. These results are computed on the synthetic database described in Section 5.1, each point in the plot corresponds to the average error over 10 synthetic sequences for each number of motion and each noise level. As can be seen, when the number of motions increases the error of the EMS rank estimation tends towards negative values. Similarly, the higher the noise the lower the rank estimation. We leave as a matter of further investigation a proper understanding of this behaviour. As a preliminary study, a possible way to correct the estimated rank could be to add 1 size for each motion to the previously EMS estimated rank. By adding only 1 size for each motion, even if the maximum was already an overestimation or a perfect estimate, the correction does not introduce too much random information, whereas if the maximum was an underestimation some important information is added.

EMS with correction is called EMS+, in Section 5 the performances between EMS and EMS+ are compared.

4.1.3. How to speed up the choice

We propose here a speed up technique that can be used in order to quickly find an affinity matrix with a high entropy. In order to have a fast but good estimation of the rank, we exploit the concavity of the overall entropy behaviour. As explained in the previous section, the entropy may have oscillations. However, by taking opportune safety measures it is possible to quickly find an affinity matrix with an entropy very close to the highest value. One can always renounce to speed in favor of a better estimation, however, in Section 5 we show that even with this approximated (but faster) choice good results are obtained.

Assuming that there may be small oscillations in the entropy function, in order to avoid to select a local maximum it is sufficient to use a large sampling step (in our experiments the step $\Delta k_g = 1/\sqrt{10}$). Moreover, to establish the gradient of the entropy we do not choose only 2 samples but 3. By doing so, when a minimum is encountered we can extend the sampling towards the two extremes until a choice can be made. Once the gradient is established, we shift our search towards the increasing gradient repeating the sampling process until the maximum is found. We would like to remark that in our experiments when the entropy was sampled in the way just explained we have not found any fluctuations in its trend. However, when the entropy is computed with a finer sampling step, in some sequences a very small oscillation may be found close to the maximum values. On the whole Hopkins155 database an oscillation can be found only in the following sequences: 1RT2TC_g13, 2R3RTCRT, 2T3RCTP, articulated, cars1 and cars8. As shown in Fig. 6 the oscillations are very small and occur very close to the maximum entropy anyway. Hence, even if in these few cases our algorithm had selected a local maximum, the built affinity matrices would have had a very high entropy.

Of course, one can always argue that there may be a particularly unlucky situation where the combination of bad estimation of NNs and noise generates a very big oscillation, to the point that even with the kind of sampling just described, our algorithm would choose a local maximum. However, the oscillations are more likely to be around the correct rank, hence there is a chance that the selected affinity matrix can provide enough information anyway. Moreover, in such extreme conditions the assumptions of the LSA framework would probably be violated leading to bad segmentation even if the rank is perfectly estimated.

4.1.4. Size estimation of the local subspaces

So far we have assumed that the size of the local subspaces (step 4) was fixed to 4. Once the value k_g has been found equal to $1/10^x$, k_s can be set to $1/\sqrt{10^x}$. This corresponds to the choice made by the authors of LSA. In fact, Yan and Pollefeys explain in [14] that when detecting the rank of the local subspaces, due to a small number of samples, the noise level is higher, so it is desirable that $k_s > k_g$.

In summary, EMS and EMS+, by exploiting the relationship between the rank estimation of the trajectory matrix and the affinity matrix, are able to automatically tune the value of k for the global and local dimension estimation without requiring knowledge about the number of motions nor the amount of noise. Better rank estimations result in a better motion segmentation and, as EMS and EMS+ do not make any assumption regarding the types of motion, it can be used under any condition.

4.2. Number of motions estimation

Another weakness of LSA is related with the final segmentation of the affinity matrix. Yan and Pollefeys in [14] suggest to use

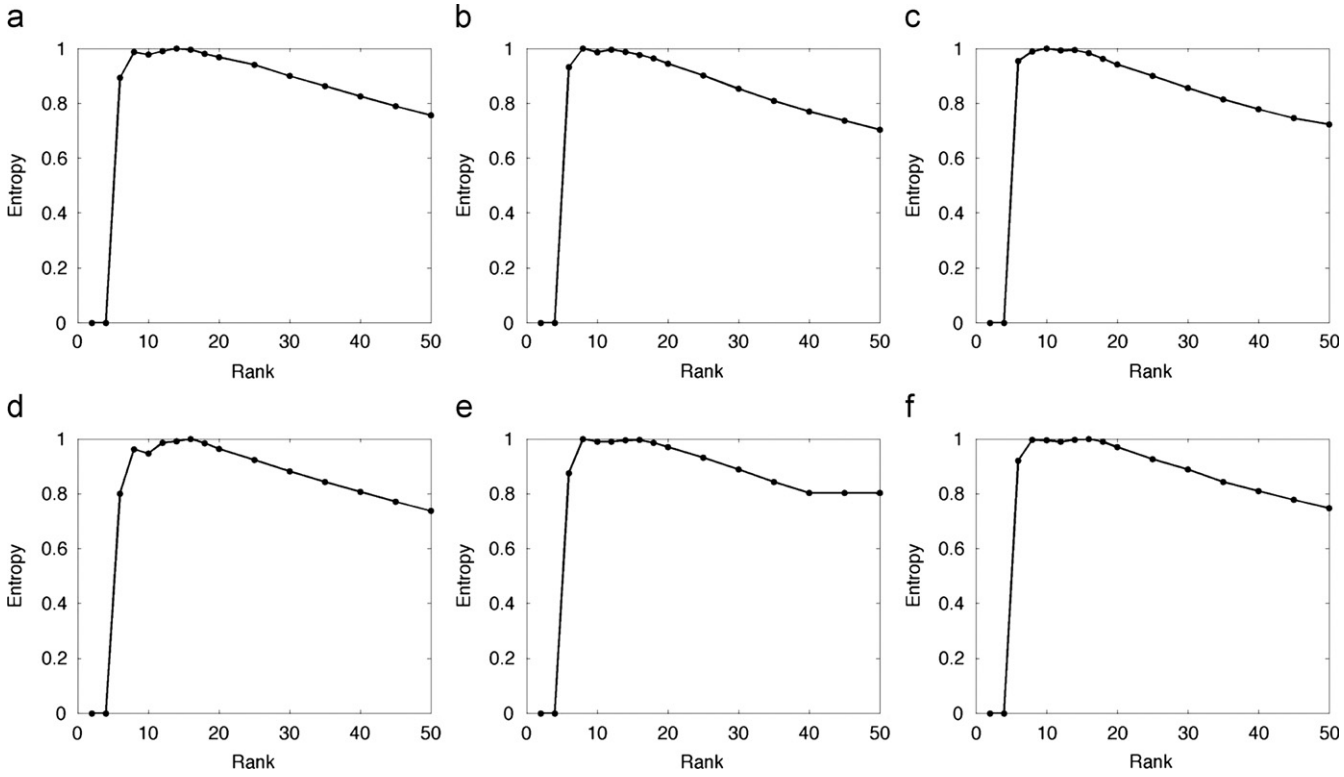

Fig. 6. The only entropy trend with oscillation found on the whole Hopkins155 database. In these plots the rank was sampled from 2 to 20 with a step of 2, from 20 to 50 with a step of 5. (a) 1RT2TC_g13; (b) 2R3RTCRT; (c) 2T3RCTP; (d) articulated; (e) cars1; (f) cars8.

Normalized Cuts [47]. Normalized Cuts is a good solution as long as the number of motions is known. When this information is not available, at every iteration the decision to terminate the process or not has to be taken. The authors of Normalized Cuts suggest to use the Cheeger constant [48] or the cost of the last cut in order to take this decision. The Cheeger constant and the cost of the last cut are both clues of how difficult it is to split the graph by removing a specific edge. When, after finding the minimum cut, one of these two values is high it means that it is not worth splitting the graph and the process can stop. Therefore, these indicators need a threshold to decide when the value is "high enough". The problem is that such a threshold is strongly influenced by the noise level and the number of motions. This explains why most of the Normalized Cuts implementations require to know in advance the number of motions.

Having tested the difficulty of using the Cheeger constant or the cost of the last cut, we try a different way, exploiting some spectral graph theory theorems. Specifically the following proposition.

Proposition 1 (*Number of connected components*). *Let G be an undirected graph with non-negative weights. Then, the multiplicity n of the eigenvalue* 0 *of the Laplacian matrix equals the number of connected components in the graph* [54].

In [47] it is shown that finding the minimum cut for splitting the graph, is equivalent to thresholding the values of the second smallest eigenvector of the Laplacian matrix $\mathbf{L}$:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{7}$$

where $\mathbf{A}$ is the adjacency matrix (specifically, in our case it is the affinity matrix), and $\mathbf{D}$ is a $p \times p$ diagonal matrix, p being the number of tracked features. Every entry $\mathbf{D}(i,i)$ contains the sum of the weights that connect node i to all of the others. Hence, matrix $\mathbf{L}$ and Proposition 1 could be used in order to estimate the number of motions. Proposition 1 refers to an ideal case where the eigenvalues that correspond to the connected components are exactly equal to 0 (which means no noise and fully independent motions). However, perturbation theory says that if there is not an ideal situation the last n eigenvalues are not equal to 0, nevertheless they should be very close to those of the ideal case [54]. Naturally, in motion segmentation, especially with real sequences, the ideal situation is not expected, but theoretically it should be possible to identify the threshold between the eigenvalues that correspond to the connected components and the remaining eigenvalues.

Proposition 1 holds true also when using the eigenvalue spectrum of the Symmetric Normalized Laplacian matrix $\mathbf{L}_{sym}$ [54,55] instead of the Laplacian matrix $\mathbf{L}$:

$$\mathbf{L}_{sym} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}. \tag{8}$$

Fig. 7 shows the eigenvalues of $\mathbf{L}$ (first row) and of $\mathbf{L}_{sym}$ (second row) of a synthetic sequence with 3 motions and with an increasing noise level. The last three eigenvalues, which should suggest the number of motions, are plotted with a red filled circle. In the plots all the eigenvalues are normalized in order to allow an easier comparison between $\mathbf{L}$ and $\mathbf{L}_{sym}$. As can be seen, when the noise increases the difference between the red filled eigenvalues and the others decreases. However, the difference between the fourth to last and the third to last eigenvalues remains rather large in the $\mathbf{L}_{sym}$ spectrum, while it becomes really small in the $\mathbf{L}$ spectrum. In the next section the experimental results of the estimation of the number of motions using $\mathbf{L}$ and $\mathbf{L}_{sym}$ are compared.

As was previously stated, when the Cheeger constant or the cost of the last cut are used, the main problem is the choice of a threshold. The same problem is present when using the spectrum of the eigenvalues, regardless of the choice of $\mathbf{L}$ or $\mathbf{L}_{sym}$. However, while with the Cheeger or the cost of the last cut there is not much information that can be used in order to take such a decision, with the eigenvalues the information of the whole spectrum can be used. Nevertheless, the

threshold cannot be fixed as the noise greatly influences the distance between the eigenvalues, as shown in Fig. 7. In order to dynamically find a threshold for every case we try different techniques. In general, estimating the number of motions using the eigenvalue spectrum can be seen as a two class classification problem: class 1 is the class of the eigenvalues above the threshold while class 2 is the class of the eigenvalues below the threshold. The number of eigenvalues inside class 2 is the estimation of the number of motions. The first technique that we test is the Fuzzy C-Means clustering (FCM) [56]. This technique returns a probability of belonging to class 1 and to class 2 for every element of the set. The second technique is Otsu's method [57] which chooses the threshold to minimise the intra-class variance. As in this particular case it seems that the inter-class variance is also playing an important role, we try to find a trade-off between the intra- and the inter-class variance. We take inspiration from the Linear Discriminant Analysis (LDA) which minimises the intra-class variance while maximising the inter-class variance. With LDA the chosen threshold t is given as

$$\underset{t}{\operatorname{argmax}} \frac{Q_1(\mu_1(t)-\mu_{all})^2+(1-Q_1)(\mu_2(t)-\mu_{all})^2}{Q_1\sigma_1^2(t)+(1-Q_1)\sigma_2^2(t)}, \tag{9}$$

where μ_1 and σ_1 are the mean and the variance of class 1 given a certain threshold t, μ_2 and σ_2 are the mean and the variance of class 2, and μ_{all} is the mean of all the eigenvalues. In the original formulation of LDA, Q_1 is the probability of belonging to class 1. However, in this context this probability is unknown (knowing the probability means knowing already the number of motions). At the same time, not providing any weight for the two classes would mean that both classes are equally likely even though this is not true. Therefore, Q_1 should be seen as weight that has to favour class 1 over class 2.

The nominator of formula (9) measures the inter-class dissimilarity, whereas the denominator measures the intra-class dissimilarity. Therefore, choosing the threshold that maximises this ratio is like choosing the threshold that maximises the inter-class dissimilarity and minimises the intra-class dissimilarity.

In the next session, one of the experiments presented is about the estimation of the number of motions using the eigenvalue spectrum of $\mathbf{L}$ and of $\mathbf{L}_{sym}$, and thresholding them with FCM, OTSU and LDA.

5. Experimental results

5.1. Databases

In order to evaluate ELSA, we perform different tests with real sequences. The database used is the Hopkins155 which is a reference database for motion segmentation, composed of 155 real video sequences: 120 with 2 motions and 35 with 3 motions. An example of two real sequences are shown in Fig. 8(a) and (b). Inside the Hopkins155 database there are different types of sequences: checkboards, traffic and articulated/non-rigid. The checkboard is the main group (104 videos) thus it is likely that the type and the amount of noise inside the database does not change much as most of the sequences are taken in the same environment. For the purpose of testing the ELSA with bigger noise changes, we create another six databases derived from the Hopkins155 adding random Gaussian noise, with standard deviations of 0.5, 1, 1.5, 2, 2.5 and 3 pixels, to the tracked point positions. The original database plus the six derived from it compose a bigger database with 1085 video sequences.

Besides the Hopkins155 database, we also use a synthetic database. Specifically, synthetic sequences composed of 50 frames, with rotating and translating cubes. Each cube has 56 tracked features. An example of a synthetic frame (for plotting reasons with just a few tracked features) is shown in Fig. 8(c). Similarly to what we

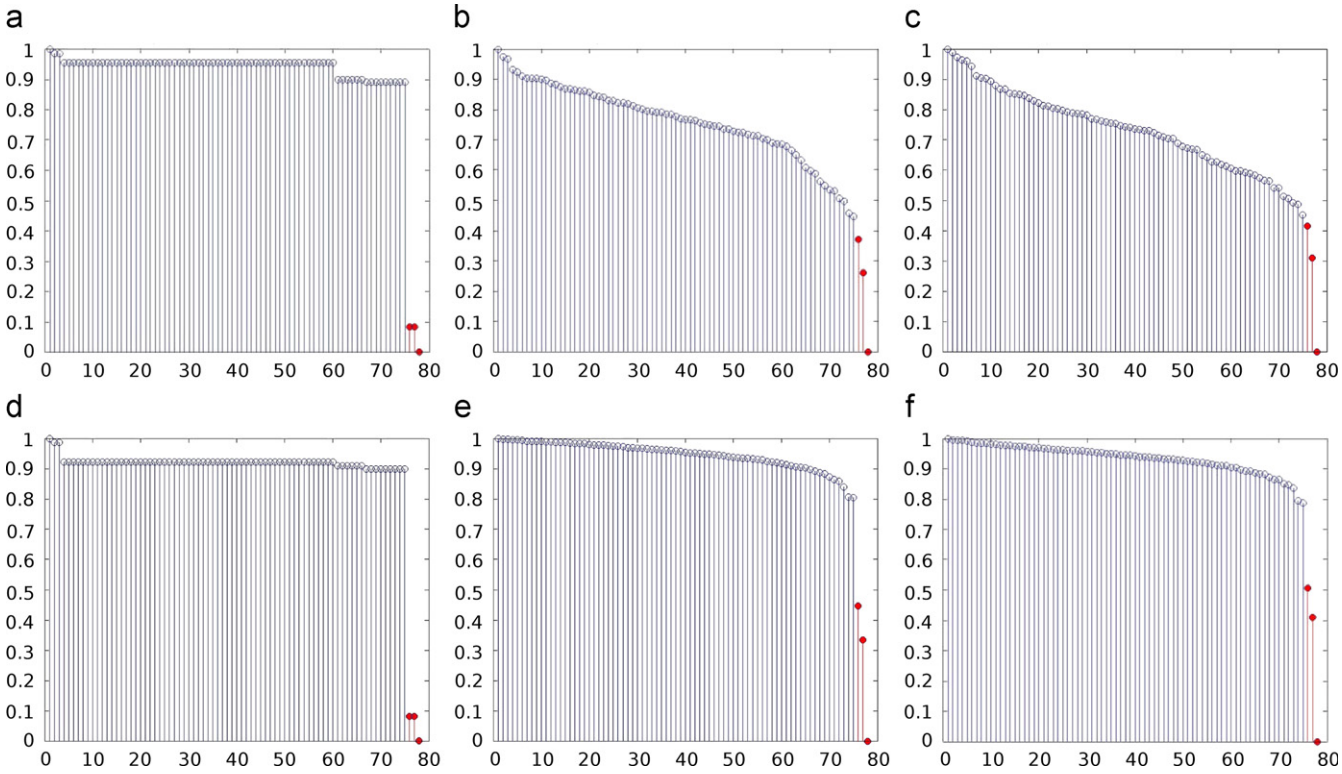

Fig. 7. Eigenvalues spectrum of $\mathbf{L}$ (fist row) and of $\mathbf{L}_{sym}$ (second row) for a synthetic sequence (similar to the one in Fig. 8(c)) with 3 rotating and translating cubes and increasing Gaussian noise level. (a) $\mathbf{L}$, $\sigma_{noise}=0.0$; (b) $\mathbf{L}$, $\sigma_{noise}=1.5$; (c) $\mathbf{L}$, $\sigma_{noise}=3.0$; (d) $\mathbf{L}_{sym}$, $\sigma_{noise}=0.0$; (e) $\mathbf{L}_{sym}$, $\sigma_{noise}=1.5$; (f) $\mathbf{L}_{sym}$, $\sigma_{noise}=3.0$.

did for the Hopkins155 we create also six derived databases adding noise with different standard deviations (from 0.5 to 3 pixels) to the original database for a total of 240 synthetic sequences. Synthetic sequences with 2 motions have been previously shown in Section 4.1.1 in order to provide some examples of the trends of the principal angles, the affinity values and the entropy as functions of r. Sequences with 2, 3, 4 and 5 motions have been used in Section 4.1.2 in order to provide some evidences of the correlation between the number of motions (or the noise level) and the rank estimated by EMS. Sequences with 3 motions have been used in Section 4.2 to show the differences between the eigenvalue spectrum of $\mathbf{L}$ and $\mathbf{L}_{sym}$. Finally, in Sections 5.3 and 5.5 we use sequences with 4 and 5 motions to test the estimation of the number of clusters and ELSA algorithm in a more challenging context.

5.2. Evaluation of the Enhanced Model Selection

In Section 4.1 it is explained how EMS is able to automatically adjust the parameter k to different noise conditions and different number of motions in the sequence. In this section the results of the misclassification over the Hopkins155 database, and the Hopkins155 database with extra noise, are shown. In these first sets of experiments the knowledge about the number of motions is assumed so that it is possible to assess the model selection techniques (MS, EMS and EMS+) independently from the accuracy of the estimation of the number of motions.

To evaluate the model selection the results of ELSA with EMS and ELSA with EMS+ are compared with the results obtained with: LSA fixing the global subspace size to 5 and $4n$ (where n is the number of motions), LSA estimating the global subspace size with MS using the best k per each sequence (i.e. the k that provided the lowest misclassification rate per each sequence on the original Hopkins155 database), and LSA with MS using the overall best k (i.e. the k value, common for the whole database, that provided the lowest mean misclassification on the original Hopkins155 database). For all the algorithms the Recursive Two-Way Ncut [47] is used for the final clustering (this is why for 0 extra noise level the LSA 5 and $4n$ results are slightly different than the ones in [17] where the clustering algorithm used was k-means with multiple restarts). We perform two set of experiments: the first is performed fixing the subspace sizes to 4, while the second is performed estimating both global and local subspace sizes.

In Fig. 9 the results obtained fixing the subspace sizes to 4 are shown. With no additional noise (noise level equal to 0 in the graphs, i.e. the original Hopkins155 database) the highest misclassification is obtained when the global subspace size is fixed to 5, this happens because a dimension of 5 corresponds for most of the global subspaces to a considerable underestimation of their size. The performances of LSA with best overall k, best k per sequence, and fixing the global subspace to $4n$ are very similar to each other, however, when the noise increases the MS tends to fail (as it was tuned for the original Hopkins155) proving that MS is very sensitive to noise. ELSA, both with EMS and EMS+, performs better than any other technique, with EMS+ proving to be more solid than EMS when the number of motions increases. ELSA performs better than MS with best k per sequence even on the original database. This may seem counterintuitive as one would expect that the best k per sequence leads to the best misclassification rate. However, it has to be remembered that the best k values were computed when also the local subspace sizes were estimated and not fixed to 4. This small difference clearly changes which are the best k values that have to be used and shows, once again, how unstable the MS is. When the noise level increases EMS and EMS+ perform better than MS because they are able to adapt automatically to the different conditions. Moreover, ELSA performs better than LSA $4n$ because when the global subspace size is fixed to $4n$ the motions are considered rigid and fully independent even when this is not true. Overall, this results show that a good model selection allows to obtain better results than fixing the global subspace size. However, model selection is also very sensitive to the noise and it requires a manual tuning in order to cope successfully with different noise levels.

The following set of experiments is obtained estimating also the local subspace sizes (as LSA $4n$ and 5 do not use any instrument to estimate the global subspace size they are not included in this set of experiments). The results are shown in Fig. 10. As expected, with the original Hopkins155 database the lowest misclassification is obtained when the k is manually tuned for each sequence. As expected, EMS and EMS+ perform worse than MS with the best k per sequence, however, they do better than MS with the best overall k. As soon as extra noise is added, the misclassification of both MS strategies rises drastically whereas EMS and EMS+ are able to tune k automatically in order to reduce the effects of the noise. As in the previous set of experiments, the misclassification rate of ELSA with EMS+ is lower than that of ELSA with EMS. With 2 motions the performances are very similar. However, with 3 motions the compensation strategy plays an important role. In fact, without extra noise ELSA with EMS+ has performances very close to the one of LSA with best k per sequence. In general ELSA with EMS+ misclassification rate is around 6% lower than the misclassification of ELSA with pure EMS. If the plots of Figs. 9 and 10 are compared it is possible to notice that, given a technique, the misclassification when the local subspace sizes are estimated are almost always better than when the local subspace sizes are fixed to 4. Such a result shows that also a correct estimation of the size of the local subspaces plays a (minor) role in providing a better segmentation.

In summary, these results show that EMS and EMS+ are able to provide a good estimation of the rank of the trajectory matrix in an automatic fashion. They do not require any a priori knowledge and are able to deal successfully with different noise levels. As it is not necessary for EMS and EMS+ to know in advance any subspace dimension, they are able to deal with different types of

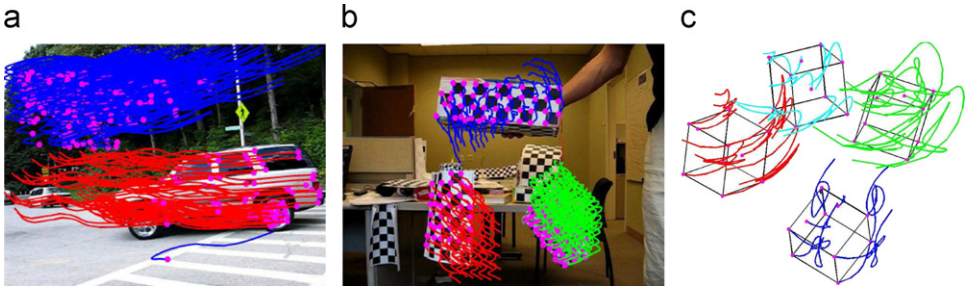

Fig. 8. Example of two input frames and the trajectories of real sequences from the Hopkins155 database and one frame from a synthetic sequence. (a) Traffic; (b) checkboards; (c) synthetic.

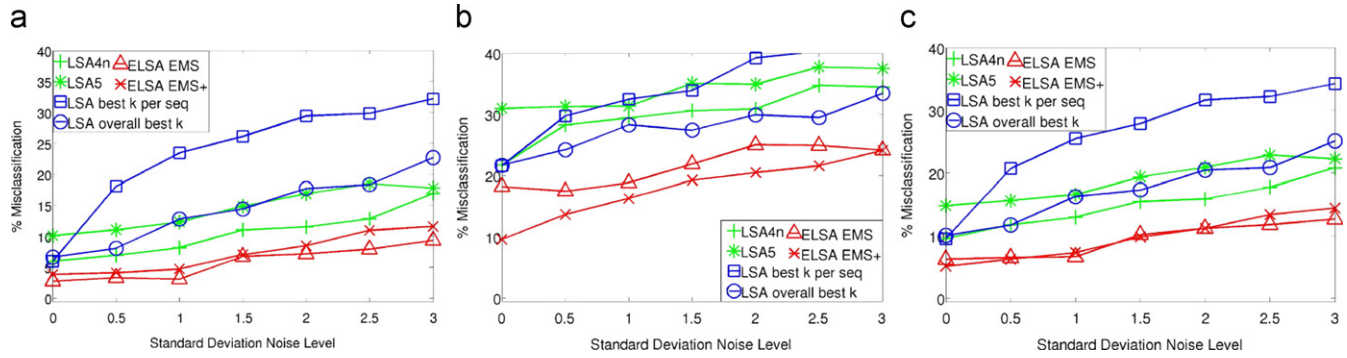

Fig. 9. Mean misclassification rate versus noise level for Hopkins155 databases (local subspace sizes are fixed to 4). (a) 2 motions; (b) 3 motions; (c) 2 and 3 motions.

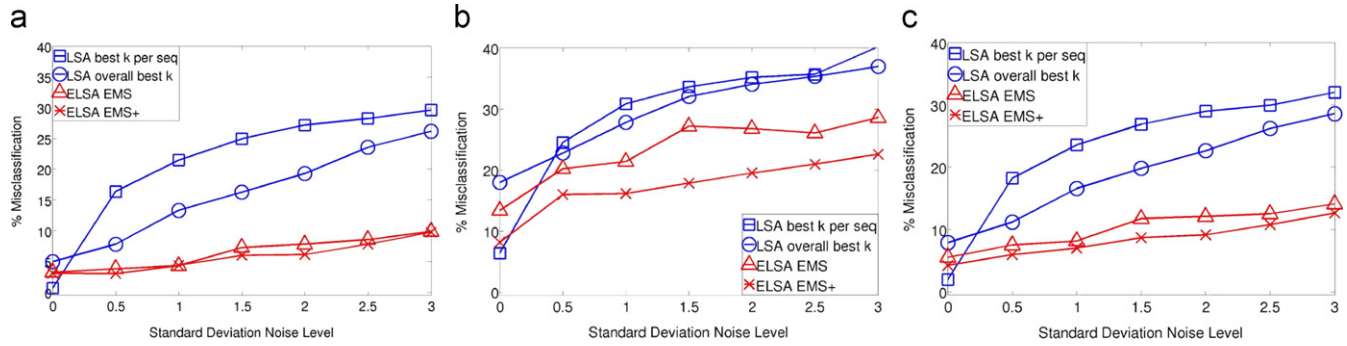

Fig. 10. Mean misclassification rate versus noise level for Hopkins155 databases (local subspace sizes are estimated). (a) 2 motions; (b) 3 motions; (c) 2 and 3 motions.

motion, which is not possible when the subspace size is fixed. Moreover, despite the fact that ELSA with EMS already provides very good results, the simple correction strategy of EMS+ allows to reach even better performances.

5.3. Estimation of the number of motions

The aim of ELSA is to be completely automatic without requiring any a priori knowledge. Therefore, the next step is to test the estimation of the number of motions. As explained in Section 4.2, we test different thresholding techniques in order to perform the estimation exploiting the eigenvalues spectrum of either the Laplacian matrix $\mathbf{L}$ or the Symmetric Normalized Laplacian $\mathbf{L}_{sym}$. In both cases the Laplacian matrices are built after EMS has been used in order to estimate the dimension of the global space. Concerning the setting of the thresholding algorithms, we tried different values and we present here the set that obtains the best results on a random subset of the Hopkins155 database (70 sequences). For the estimation using FCM, the best results are obtained by counting the eigenvalues with a probability of belonging to class 2 equal to or greater than 0.9. For OTSU the best results are obtained using only the last 20 eigenvalues. For LDA the best results are obtained with $Q_1 = 0.8$.

A first qualitative study suggests that $\mathbf{L}_{sym}$ is more robust against noise, as previously shown in Fig. 7. We run also a quantitative test estimating the number of motions on the Hopkins155 database using both $\mathbf{L}$ and $\mathbf{L}_{sym}$ with the thresholding techniques explained in the previous section. Table 2 shows the mean and variance of the error of the estimated number of motions for the tested thresholding techniques (the absolute value of the error is considered). As expected, regardless of technique, mean and variance are always considerably smaller when using $\mathbf{L}_{sym}$ than when using $\mathbf{L}$. As the Symmetric Normalized Laplacian spectrum seems to be more robust against noise, we choose $\mathbf{L}_{sym}$ in order to estimate the number of motions.

From the results of Table 2, OTSU seems to be the weakest measure. This is confirmed also when the same test is performed on the Hopkins155 with extra noise, as shown in Fig. 11. The numbers on each boxplot correspond to the percentage of sequences where the error in the number of cluster estimation was (from bottom to top) 0, ± 1, ± 2 or greater than 2 (in absolute value). From these boxplots it is possible to see that FCM and LDA both have a very high percentage of perfect estimation and their first and second quartiles collapse on the median.

FCM and LDA have similar performances. A deeper analysis reveals that FCM is particularly good with 2 motions: on the original database it has a percentage of perfect estimation equal to 84.2% against 75.0% of LDA. However, when the number of motions increases, FCM appears to be less robust than LDA: with 3 motions the percentage of perfect estimation of FCM is equal to 54.3% against 57.1% of LDA. The fact that the Hopkins155 database has more sequences with 2 motions tends to favour FCM. A similar conclusion can be drawn when the noise level increases, in fact the difference between the perfect estimation of FCM and LDA drops from 6.4%, in the original Hopkins155 (considering all the sequences), to only 1.3%, in the database with 3 pixels of noise level, despite that the sequences with 2 motions are still overrepresented in the Hopkins155 database.

In order to verify the reliability if these clues we perform an experiment with synthetic sequences with 4 and 5 motions and different noise levels and we compare the estimations about the number of motions obtained using FCM and LDA. The results of this experiment, shown in Fig. 11(b), confirm the conclusions drawn from the results on the Hopkins155 database and show that LDA outperforms FCM in terms of percentage of perfect estimation (40.7% against 27.9%). If one wants to focus only on the Hopkins155 database, which has a majority of sequences with 2

motions, on average FCM would have slightly better performances than LDA. However, given the results just presented, if ELSA has to be applied on a database with unknown number of motions it would be better to use LDA. As our aim is not to find the best tuning for the Hopkins155 but is to have a generally good motion segmentation algorithm, we chose to use LDA as the thresholding technique of the $\mathbf{L}_{sym}$ eigenvalue spectrum.

5.4. Segmentation without priors

Finally, the misclassification rate when segmenting without any a priori knowledge is presented. To the best of our knowledge, besides ELSA, the only technique which is able to provide satisfactory results without a priori knowledge is the ALC [18]. ALC can be forced to select the segmentation with the smallest coding length, in this way the number of motions is not required. In order to compare the two techniques we impose an upper bound in the number of motions estimated equal to 5. It should be noted that computing the misclassification rate when the number of estimated motions does not match with the real number could be done in different ways. In fact the error due to overestimation of the number of motions could be somehow corrected in a post-processing step with a merging strategy. On the other hand, an underestimation usually means that points from at least two different motions are considered as the same one, so it can be seen as a more crucial error. However, also in this case one could argue that for each cluster a deeper analysis could be performed in order to correct the segmentation with a splitting approach rather than a merging one. In these experiments the misclassification when the number of motions is wrongly estimated is as follows. We take into account all the possible one-to-one associations between the estimated groups of trajectories and the ground truth groups of trajectories (clearly, some groups will not be associated). We select the association that minimises the error computed by summing the error of the associated groups and the number of features that belong to non-associated groups.

Table 2
Mean and variance of the absolute value error of the number of motions estimation on the Hopkins155 database.

|Error|	FMC	OTSU	LDA
$\mathbf{L}$			
μ	0.6	0.8	0.7
σ	0.6	0.9	0.9
$\mathbf{L}_{sym}$			
μ	0.23	0.47	0.37
σ	0.19	0.48	0.42

The best results of ELSA are obtained using EMS+. However, the dimension (i.e. the final rank used) of the global space and the spectrum of the eigenvalues of $\mathbf{L}_{sym}$ have a mutual influence on each other. Thus, after the first estimation of the rank performed with EMS an iterative procedure is used. In an alternating fashion the number of motions and the rank is estimated. This alternation is iterated until one of the following conditions is verified: the estimated number of motions does not change, a loop is detected, or a maximum number of iterations is reached (for the experiments the maximum number of iterations is set to 5, however, this condition is never verified in our tests). Summarising, the results presented in this section for ELSA are obtained using: EMS+ for the model selection, and the eigenvalue spectrum of $\mathbf{L}_{sym}$ thresholded dynamically by LDA for the estimation of the number of motions.

Table 3 shows mean, variance and median of the misclassification rate of ALC and ELSA on the original Hopkins155 database divided per number of motions and per type of sequence. The overall mean misclassification rate is similar for the two algorithms: 12.86% for ALC and 10.75% for ELSA. However, more remarkable is the difference between the median values: the median of ALC is 10.23% whereas the median of ELSA is only 1.31%. This suggests that ELSA performs very well in most of the sequences and it fails only in a few of them. This is confirmed also by the results shown in Fig. 12 where the histogram of the misclassification is presented. From the histogram it is possible to appreciate that ELSA has more than 75 sequences with a misclassification rate between 0% and 1%, around 30 sequences more than ALC. ELSA has more sequences than ALC also in the ranges 1–2% and 2–3%. The reason why ELSA is generally very good, but when it fails the misclassification is very high, could be explained by Fig. 3(i)–(p). From those plots it is possible to notice that if the rank is not well estimated (especially if it is underestimated) the affinity values quickly collapse to 1 or 0. When this happens the clustering process becomes very noisy, hence the segmentation result is greatly affected. ELSA also performs better than ALC in terms of estimation of the number of motions. The mean errors (in absolute values) of the two algorithms are 1.16 for ALC and 0.36 for ELSA.

In Fig. 13 the trends of the mean and median misclassification rate of ALC and ELSA when using the Hopkins155 database with extra noise are presented. The trends of both ALC and ELSA are very stable showing that the algorithms are robust against noise. Mean and median performances of ELSA are always better than those of ALC with 2 motions. With 3 motions this is also true for the original version of the database. When there are 3 motions and high noise levels ALC is more robust than ELSA. This is probably due to the fact that the EMS+ correction is

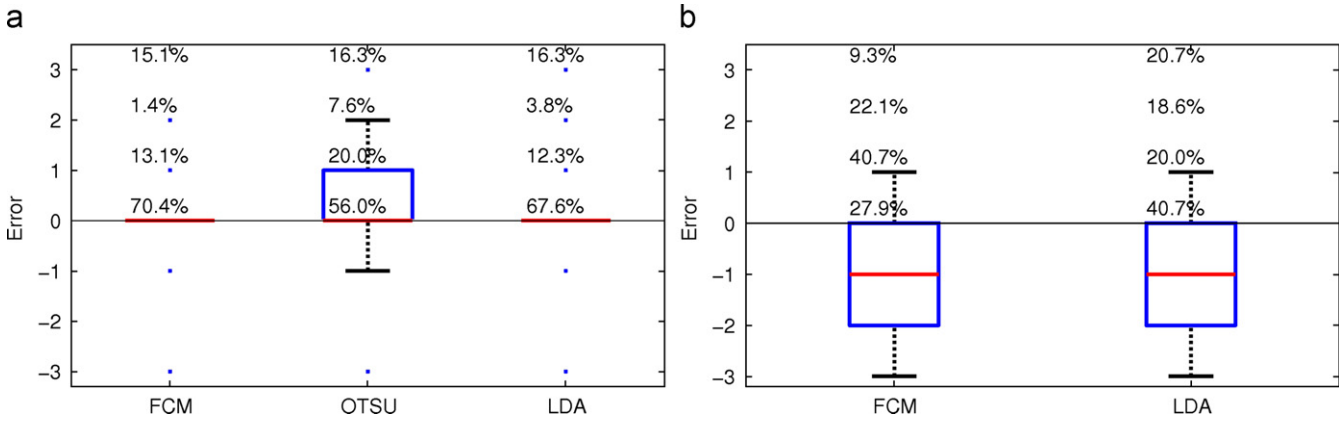

Fig. 11. Boxplots of the error of the number of motions estimation. (a) Hopkins155 with extra noise; (b) 4 and 5 synthetic motions with noise.

parameterised with the number of motions but not with the noise level (it was shown in Fig. 5 that also the noise plays a role in the underestimation of the rank).

As far as the computation time is concerned, the average time required by ALC to complete a segmentation is almost 20 times longer (around 31 seconds for ELSA and 573 seconds for ALC). In order to compute all the 1085 sequences ALC took 172.69 hours against 9.33 hours of ELSA on the same computer (Matlab implementation on Intel Core2 Duo CPU at 2.66 GHz, with 4 GB RAM).

Table 3
Mean, variance and median of the misclassification on the Hopkins155 database with no a priori knowledge; # column shows the number of sequences.

Hopkins155	2 motions			3 motions			2 and 3 motions		
	#	ALC (%)	ELSA (%)	#	ALC (%)	ELSA (%)	#	ALC (%)	ELSA(%)
Check.									
Mean	75	14.15	**8.90**	25	12.51	**10.71**	100	13.74	**9.35**
Var		1.86	2.48		1.04	1.53		1.64	2.23
Median		12.02	**0.53**		11.33	**5.96**		11.84	**0.57**
Traff.									
Mean	34	**12.03**	12.82	8	**17.46**	19.84	42	**13.07**	14.16
Var		2.06	3.91		2.03	4.29		2.05	3.96
Median		4.43	**2.75**		15.10	**12.00**		7.10	**3.40**
Artic.									
Mean	11	**5.27**	10.36	2	**6.72**	11.17	13	**5.49**	10.48
Var		1.67	2.41		0.73	2.50		1.46	2.22
Median		**0.88**	3.03		**6.72**	11.17		**0.88**	3.03
All									
Mean	120	12.73	**10.15**	35	13.31	**12.82**	155	12.86	**10.75**
Var		1.93	2.86		1.25	2.19		1.77	2.71
Median		9.10	**0.94**		11.79	**9.37**		10.23	**1.31**

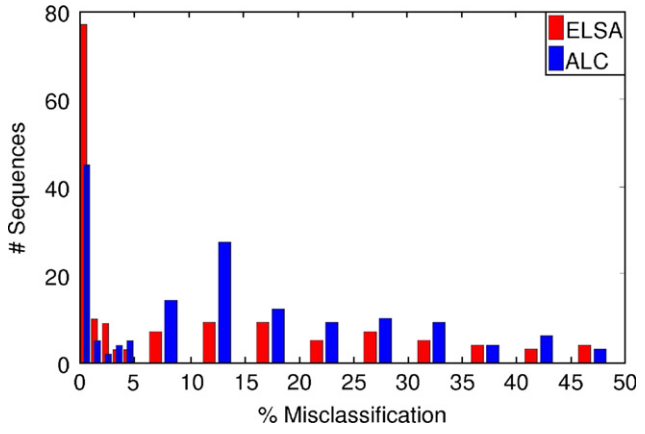

Fig. 12. Histogram of the misclassification of ALC and ELSA on the original Hopkins155 databases with no a priori knowledge; misclassification from 0% to 5% are sub-sampled with bins of 1%, misclassification greater than 5% are sub-sampled with bins of 5%.

5.5. Extension to 4 and 5 motions

As the Hopkins155 DB contains maximum 3 motions, some synthetic experiments were also done in order to have an idea of the behaviour of ELSA and ALC when the number of motions increases. Fig. 14 shows the misclassification rate for 4 and 5 motions depending on the noise level. It is interesting to notice that the performances with no noise between ALC and ELSA are almost the same. When the noise is between 0.5 and 2 pixels of standard deviation, ELSA has a lower misclassification rate. For both 4 and 5 motions with 2.5 and 3 pixels of noise the performance of ELSA degrades faster than that of ALC. The triangles in Fig. 14 show the variance of the misclassification, note that in these last two cases ELSA variance is quite large showing that ELSA still performs well in most of the sequences but fails in a few of them.

The behaviour of the two algorithms in these synthetic experiments is coherent with that on the Hopkins155 database. In fact, with a small amount of noise ELSA is better than ALC; when the noise increases ALC is more robust while ELSA tends to have a good behaviour but fails badly in few circumstances (as previously shown in the histogram of Fig. 12).

6. Conclusions and perspectives

The main contribution of this paper is the new motion segmentation algorithm called Enhanced Local Subspace Affinity (ELSA) which is able to strengthen the weak points of LSA. Especially, we have proposed a new Enhanced Model Selection (EMS/EMS+) technique which does not require any a priori knowledge. The comparison between ELSA and LSA with classical Model Selection (MS) highlighted the ability of EMS/EMS+ to adapt to different noise levels and number of motions. We have also introduced an estimation of the number of motions based on the eigenvalue spectrum of the Symmetric Normalized Laplacian matrix. The number of motions is automatically estimated by finding a threshold which is dynamically computed using an LDA inspired approach.

ELSA has been compared with one of the best performing techniques for motion segmentation without a priori knowledge: the Agglomerative Lossy Compression (ALC). On the Hopkins155 database ELSA performs better than ALC both in terms of mean and median misclassification. With extra noise ELSA is better than ALC in sequences with 2 motions while ALC is better with 3 motions. On synthetic sequences with 4 and 5 motions the results

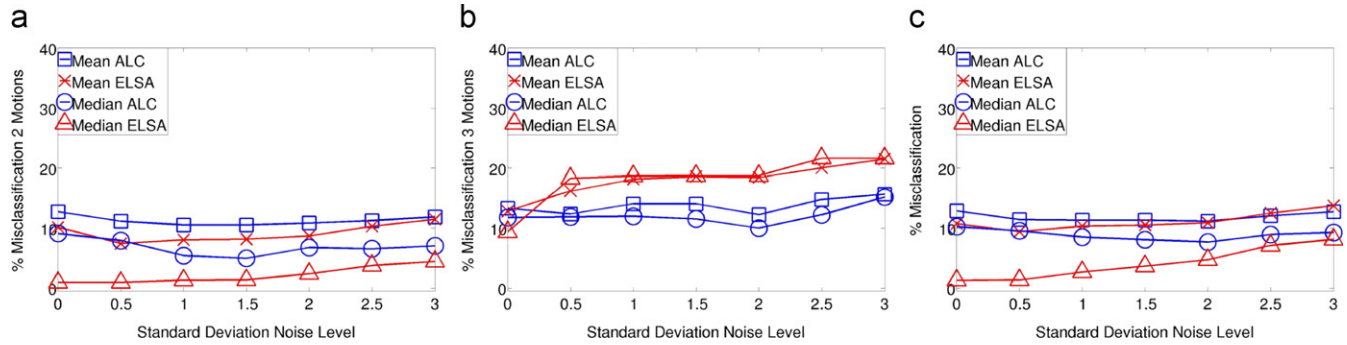

Fig. 13. Mean and median misclassification rate versus noise level for Hopkins155 databases with no a priori knowledge. (a) 2 motions; (b) 3 motions; (c) 2 and 3 motions.

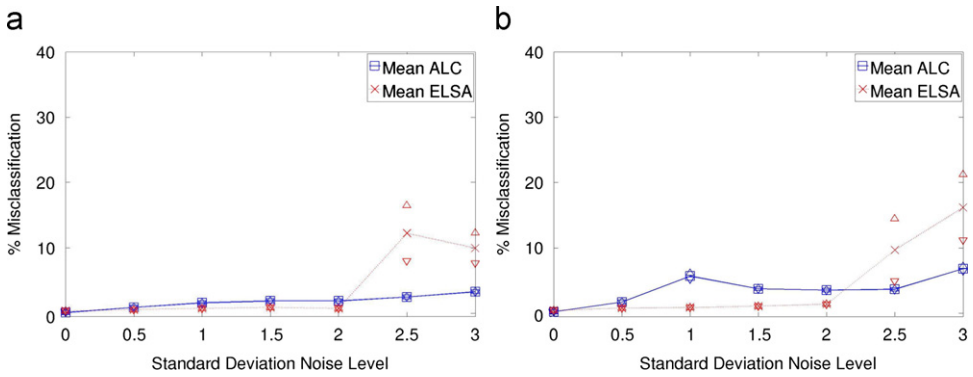

Fig. 14. Mean misclassification rate and variance versus noise level for synthetic experiments with no a priori knowledge. (a) 4 motions; (b) 5 motions.

where coherent with the ones obtained on the Hopkins155 database: ELSA performs better than ALC with low noise levels but ALC is more robust when the noise increases. The main drawback of ALC is the computation time, which is almost 20 times longer than that required for ELSA.

To conclude, we discuss now some future directions of research. As for future work connected with ELSA, we pointed out that we found a correlation between the number of motions, the noise and the position of the maximum entropy. It seems that the higher the number of motions and the noise level, the more the entropy maximum tends to occur with a lower r. This has been already shown with synthetic sequences and confirmed by the fact that EMS+ performs better than EMS. However, a deeper study could reveal more useful ways of exploiting such a correlation.

Nowadays the misclassification rates, assuming the number of motions is known, are already good. Despite the fact that the misclassification rates could be further improved we believe that future work should focus on the ability to estimate the number of motions in a more efficient way. ALC and ELSA already provide satisfactory results without having this information, however, there is room for improvement especially when the motions becomes dependent. In general, feature-based techniques, such as manifold clustering, have the advantage over dense-based approaches of reducing dramatically the computation. However, feature-based techniques have to rely on the ability of the tracker to find salient points and track them successfully throughout the video sequence. Nowadays, such an assumption is not too constraining but it is important to develop algorithms able to deal with only a few points (ideally from 5 to 7) per motion instead of requiring lots of them. Moreover, another feature that we believe is very important in order to have a complete motion segmentation system, useful in real time applications, is the ability to work incrementally. An ideal incremental algorithm should be able to refine the segmentation at every new frame (or every small group of frames) without recomputing the whole solution from the beginning.

Acknowledgements

This work has been supported by the Spanish Ministry of Science projects DPI2007-66796-C03-02 and DPI2008-06548-C03-03/DPI. L. Zappella is supported by the Catalan government scholarship 2009FI_B1 00068. E. Provenzi acknowledges the Ramón y Cajal fellowship by Ministerio de Ciencia y Tecnología de España. The authors would like to thank the reviewers for their valuable and crucial comments.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.patcog.2010.08.015.

References

[1] A. Cavallaro, O. Steiger, T. Ebrahimi, Tracking video objects in cluttered background, IEEE Trans. Circuits Syst. Video Technol. 15 (4) (2005) 575–584.
[2] A. Colombari, A. Fusiello, V. Murino, Segmentation and tracking of multiple video objects, Pattern Recogn. 40 (4) (2007) 1307–1317.
[3] D. Cremers, S. Soatto, Motion competition: a variational approach to piecewise parametric motion segmentation, Int. J. Comput. Vision. 62 (3) (2005) 249–265.
[4] H. Shen, L. Zhang, B. Huang, P. Li, A map approach for joint motion estimation, segmentation, and super resolution, IEEE Trans. Image Process. 16 (2) (2007) 479–490.
[5] N. Vaswani, A. Tannenbaum, A. Yezzi, Tracking deforming objects using particle filtering for geometric active contours, IEEE Trans. Pattern Anal. Machine Intell. 29 (8) (2007) 1470–1475.
[6] R. Stolkin, A. Greig, M. Hodgetts, J. Gilby, An em/e-mrf algorithm for adaptive model based tracking in extremely poor visibility, Image Vision Comput. 26 (4) (2008) 480–495.
[7] L. Wiskott, Segmentation from motion: combining Gabor- and Mallat-wavelets to overcome aperture and correspondence problem, Lecture Notes in Computer Science, vol. 1296, 1997, pp. 329–336.
[8] M. Kong, J.-P. Leduc, B. Ghosh, V. Wickerhauser, Spatio-temporal continuous wavelet transforms for motion-based segmentation in real image sequences, IEEE Image Proc. 2 (1998) 662–666.
[9] L.M. Jos, A. Zuloaga, C. Cuadrado, J. Lzaro, U. Bidarte, Hardware implementation of optical flow constraint equation using fpgas, Comput. Vis. Image Und. 98 (3) (2005) 462–490.
[10] A. Bugeau, P. Pérez, Detection and segmentation of moving objects in complex scenes, Comput. Vis. Image Und. 113 (2009) 459–476.
[11] B. Ommer, T. Mader, J.M. Buhmann, Seeing the objects behind the dots: recognition in videos from a moving camera, Int. J. Comput. Vision 83 (2009) 57–71.
[12] M.P. Kumar, P.H. Torr, A. Zisserman, Learning layered motion segmentations of video, Int. J. Comput. Vision 76 (3) (2008) 301–319.
[13] C. Min, G. Medioni, Inferring segmented dense motion layers using 5d tensor voting, IEEE Trans. Pattern Anal. Machine Intell. 30 (9) (2008) 1589–1602.
[14] J. Yan, M. Pollefeys, A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate, Lecture Notes in Computer Science, vol. 3954, 2006, pp. 94–106.
[15] J. Yan, M. Pollefeys, A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video, IEEE Trans. Pattern Anal. Machine Intell. 30 (5) (2008) 865–877.
[16] L. Zappella, X. Lladó, J. Salvi, Rank estimation of trajectory matrix in motion segmentation, Electron. Lett. 45 (11) (2009) 540–541.
[17] R. Tron, R. Vidal, A benchmark for the comparison of 3-d motion segmentation algorithms, in: Proceedings of CVPR IEEE, 2007, pp. 1–8.
[18] S.R. Rao, R. Tron, R. Vidal, Y. Ma, Motion segmentation via robust subspace separation in the presence of outlying, in: Proceedings of CVPR IEEE, 2008, pp. 1–8.
[19] L. Zappella, X. Lladó, J. Salvi, New trends in motion segmentation, In-TECH (Ed.), Intechweb.org, 2009, pp. 31–46.
[20] T. Li, V. Kallem, D. Singaraju, R. Vidal, Projective factorization of multiple rigid-body motions, in: Proceedings of CVPR IEEE, 2007, pp. 1–6.

[21] M.A. Fischler, R.C. Bolles, Ransac random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (1981) 381–395.
[22] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, D. Kriegman, Clustering appearances of objects under varying illumination conditions, Proceedings of CVPR IEEE, vol. 1, 2003, pp. 11–18.
[23] N.P. da Silva, J.P. Costeira, Subspace segmentation with outliers: a Grassmannian approach to the maximum consensus subspace, in: Proceedings of CVPR IEEE, 2008, pp. 1–6.
[24] K. Kanatani, C. Matsunaga, Estimating the number of independent motions for multibody motion segmentation, Proceedings of the Fifth ACCV, vol. 1, 2002, pp. 7–12.
[25] K. Kanatani, Statistical optimization and geometric visual inference, in: Lecture Notes in Computer Science, 1997, pp. 306–322.
[26] Y. Sugaya, K. Kanatani, Geometric structure of degeneracy for multibody motion segmentation, in: Lecture Notes in Computer Science, 2004, pp. 13–25.
[27] A. Gruber, Y. Weiss, Multibody factorization with uncertainty and missing data using the em algorithm, Proceedings of CVPR IEEE, vol. 1, 2004, pp. 707–714.
[28] A. Gruber, Y. Weiss, Factorization with uncertainty and missing data: exploiting temporal coherence, NIPS, vol. 16, MIT Press, Cambridge, MA, 2004.
[29] A. Gruber, Y. Weiss, Incorporating non-motion cues into 3d motion segmentation, in: Lecture Notes in Computer Science, 2006, pp. 84–97.
[30] Y. Ma, H. Derksen, W. Hong, J. Wright, Segmentation of multivariate mixed data via lossy data coding and compression, IEEE Trans. Pattern Anal. Machine Intell. 29 (9) (2007) 1546–1562.
[31] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization method, Int. J. Comput. Vision 9 (2) (1992) 137–154.
[32] J.P. Costeira, T. Kanade, A multibody factorization method for independently moving objects, Int. J. Comput. Vision 29 (3) (1998) 159–179.
[33] N. Ichimura, F. Tomita, Motion segmentation based on feature selection from shape matrix, Syst. Comput. Jpn. 31 (4) (2000) 32–42.
[34] L. Zelnik-Manor, M. Irani, Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations, Proceedings of CVPR IEEE, vol. 2, 2003, pp. 287–293.
[35] H. Zhou, T.S. Huang, Recovering articulated motion with a hierarchical factorization method, Gesture Workshop, 2003, pp. 140–151.
[36] R. Vidal, R. Hartley, Motion segmentation with missing data using powerfactorization and gpca, Proceedings of CVPR IEEE, vol. 2, 2004, pp. 310–316.
[37] R. Vidal, R. Tron, R. Hartley, Multiframe motion segmentation with missing data using powerfactorization and gpca, Int. J. Comput. Vision 79 (2008) 85–105.
[38] C. Julia, A. Sappa, F. Lumbreras, J. Serrat, A. Lopez, Rank estimation in 3d multibody motion segmentation, Electron. Lett. 44 (4) (2008) 279–280.
[39] G. Chen, G. Lerman, Spectral curvature clustering (scc), Int. J. Comput. Vision 81 (2009) 317–330.
[40] A. Goh, R. Vidal, Segmenting motions of different types by unsupervised manifold clustering, in: Proceedings of CVPR IEEE, 2007, pp. 1–6.
[41] L.K. Saul, S.T. Roweis, Think globally, fit locally: unsupervised learning of low dimensional manifolds, J. Mach. Learn. Res. 4 (2003) 119–155.
[42] A. Goh, R. Vidal, Clustering and dimensionality reduction on riemannian manifolds, in: Proceedings of CVPR IEEE, 2008, pp. 1–7.
[43] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Proceedings of CVPR IEEE, 2009, pp. 2790–2797.
[44] X. Llado, A.D. Bue, L. Agapito, Non-rigid metric reconstruction from perspective cameras, Image Vision Comput. 28 (9) (2010) 1339–1353.
[45] S. Koterba, S. Baker, I. Matthews, C. Hu, J. Xiao, J.F. Cohn, T. Kanade, Multiview aam fitting and camera calibration, in: ICCV, 2005, pp. 511–518.
[46] K. Kanatani, Motion segmentation by subspace separation and model selection, IEEE Internat. Conf. Comput. Vision 2 (2001) 586–591.
[47] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Machine Intell. (2000) 888–905.
[48] J. Cheeger, A lower bound for the smallest eigenvalue of the laplacian, Problems in Analysis (1970) 195–199.
[49] A.V. Knyazev, M.E. Argentati, Principal angles between subspaces in an a-based scalar product: algorithms and perturbation estimates, SIAM J. Sci. Comput. 23 (6) (2002) 2008–2040.
[50] A. Björck, G.H. Golub, Numerical methods for computing angles between linear subspaces, Math. Comput. 27 (123) (1973) 579–594.
[51] P.-A. Absil, A. Edelman, P. Koev, On the largest principal angle between random subspaces, Linear Algebra Appl. 414 (1) (2006) 288–294.
[52] L. Ambrosio, N. Gigli, G. Savaré, Gradient flows in metric spaces and in the space of probability measures, in: Lectures in Mathematics, Birkhauser, 2005.
[54] F. Chung, Spectral Graph Theory, CBMS Regional Conference Series in Mathematics, vol. 92, American Mathematical Society, 1997.
[55] U. von Luxburg, A tutorial on spectral clustering, Statist. Comput. 17 (4) (2007) 395–416.
[56] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, Norwell, MA, USA, 1981.
[57] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.

L. Zappella received his MSc degree in Computer Science at the University of Milan at the end of 2003. He received the European MSc degree in Computer Vision and Robotics (VIBOT) in 2008. Currently, he is completing his PhD at the University of Girona. His research interests are in d image processing and computer vision, with specific focus on motion segmentation and structure from motion.

X. Lladó received the B.S. degree in Computer Science in 1999, and the PhD in Computer Engineering in 2004. Currently, he is a lecturer at the University of Girona. His research interests are in the field of image processing and computer vision, focusing on colour and texture analysis, shape from shading, and structure from motion.

Edoardo Provenzi received the degree in Physics at Università di Milano, Italy, in 2000 and the PhD in Mathematics and applications at Università di Genova, Italy, in 2003. He is currently a post-doc Ramón y Cajal researcher at Universitat Pompeu Fabra in Barcelona, Spain, in the field of computer vision, with particular interest in mathematical models of color image processing.

J. Salvi graduated in Computer Science at the Technical University of Catalonia in 1993, received the MSc in Computer Science in July 1996 and the PhD in Industrial Engineering in 1998 both at the University of Girona, Spain. He is currently an associate professor and a senior researcher at the Computer Vision and Robotics Group, University of Girona.