Adaptive Motion Segmentation Algorithm Based on the Principal Angles Configuration

L. Zappella[a], E. Provenzi[b], X. Lladó[a] and 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

Abstract. Many motion segmentation algorithms based on manifold clustering rely on a accurate rank estimation of the trajectory matrix and on a meaningful affinity measure between the estimated manifolds. While it is known that rank estimation is a difficult task, we also point out the problems that can be induced by an affinity measure that neglects the distribution of the principal angles. In this paper we suggest a new interpretation of the rank of the trajectory matrix and a new affinity measure. The rank estimation is performed by analysing which rank leads to a configuration where small and large angles are best separated. The affinity measure is a new function automatically parametrized so that it is able to adapt to the actual configuration of the principal angles. Our technique has one of lowest misclassification rates on the Hopkins155 database and has good performances also on synthetic sequences with up to 5 motions and variable noise level.

1 Introduction

Given a cloud of features tracked throughout a video sequence, the motion segmentation problem consists of clustering together features that follow the same movement. Such a problem is a fundamental step for many computer vision tasks like robotics, inspection, video surveillance, and many other applications. Motion segmentation has become even more important after the introduction of the structure from motion algorithms, which can mostly deal with only one motion at a time [8].

A 3D cloud of P points that belong to N independent and rigid motions can be mapped onto the video sequence through affine projection. The 2D position of each point at each frame can be stored into a *trajectory matrix* $\mathbf{W} \in \mathbb{R}^{2F \times P}$, where F is the total number of frames of the input sequence. Assuming no noise and no outliers, most of the rigid motion segmentation algorithms based on manifold clustering rely on a simple assumption: each independent motion generates a *local subspace* of size at most 4, therefore the union of the local subspaces generates a *global subspace* of size at most $4N$, size that corresponds to the rank of $\mathbf{W}$.

Two main ideas distinguish the majority of motion segmentation algorithms based on manifold clustering that can be found in the literature: the first is *how*

they estimate the manifold generated by each trajectory, the second is how they group them through the selection of *suitable common properties*.

Related Works on Motion Segmentation via Manifold Clustering. In [10] the authors use the Generalized Principal Component Analysis (GPCA) in order to fit a polynomial of degree N to the data, where N is the number of subspaces. Then, they estimate the basis of the subspaces using the derivatives of the polynomial and they build a similarity matrix based on the $\cos^2$ function of the principal angles (PAs) between the subspaces. Another way for the subspace estimation is via the singular value decomposition (SVD) of $\mathbf{W}$, like in the Local Subspace Affinity [11] framework (LSA). LSA also uses the PAs between subspaces in order to build the affinity matrix, however, LSA adopts a different similarity function. An Enhanced LSA (ELSA) is proposed in [12] where one of the improvements is a more robust model selection for the estimation of the global subspace size. Also in [6] the dimension of the global subspace is at the center of the study, they suggest lower and upper bounds together with a data-driven procedure for choosing the optimal ambient dimension. In [3], a new way for describing the subspaces called Sparse Subspace Clustering (SSC) is presented. The authors exploit the fact that each point (in the global subspace) can be described with a sparse representation (obtained by an ℓ_1 optimization) with respect to the dictionary composed by all of the points. The final similarity matrix is built using the coefficients of the sparse representation. Another idea is used in [4], where the authors propose a subspace segmentation algorithm based on a Grassmannian minimization approach. The estimation of the subspaces is performed via the Maximum Consensus Subspace (MCS) criteria. The same framework is further extended by using the Normalized Subspace Inclusion (NSI) similarity measure [5] between the PAs of the estimated subspaces. The Agglomerative Lossy Compression (ALC) algorithm [7] differs from the previous methods in that it does not require a similarity matrix. ALC is an agglomerative strategy that consists of minimizing the segmentation coding length in order to find the shortest coding length which is theoretically the optimal.

All of these techniques rely on the ability of the algorithms to estimate the subspaces and then to compare them (with exception of ALC). As shown in [9, 12] the size estimation of the global subspace (when required) is a critical and very difficult step. Moreover, the similarity measures used until now are *rigid* as they always assume that the features between similar and different subspaces are well separated. However, we show in section 2.4 that such an assumption is not always verified.

Our Contribution. In this work we provide two main contributions: a new interpretation of the global subspace size estimation and a new similarity measure between subspaces. Our new subspace size estimation does not depend on any sensitive parameter, and it is able to select the dimension of the global subspace where the distribution of the PAs is the most suited for the clustering step. Moreover, our similarity measure is able to dynamically adapt to the distribution of the PAs.

The results of these two contributions are evaluated on the LSA framework. We compared our model with some state of the art techniques [11, 9, 7, 12, 5, 3] on the Hopkins155 database [9], showing that our proposal outperforms all of the LSA-based algorithms, providing one of the lowest misclassification rate in the literature. Our method will be also applied on synthetic sequences from 2 to 5 motions with a controlled noise level in order to test the robustness against noise and the behaviour with more than 3 motions. Matlab source code of our algorithm can be found at: `http://eia.udg.es/~zappella`.

2 Our proposal

In this section we present a new rank estimation for $\mathbf{W}$ based on the clusterization level of the principal angles and a new adaptive similarity measure for principal angles. We apply these two techniques to the LSA framework as it is theoretically able to deal with different types of motion: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. Before going into the detail of our proposal we introduce a convenient notation and we discuss some issues regarding the principal angles.

2.1 Notation

Given a collection of N subspaces, the PAs between two subspaces S_j and S_l, for $j, l = 1, \ldots, N$, are defined recursively as a series of angles $0 \leq \theta_1 \leq \ldots \leq \theta_i \leq \ldots \leq \theta_M \leq \pi/2$, where $M = \min\{rank(S_j), rank(S_l)\}$:

$$\begin{aligned} \cos(\theta_1) &= \max_{u \in S_j, v \in S_l} u^T v = u_1^T v_1 \\ \cos(\theta_i) &= \max_{u \in S_j, v \in S_l} u^T v = u_i^T v_i, \forall i = 2, \ldots M \end{aligned} \tag{1}$$

such that: $\|u\| = \|v\| = 1$, $u^T u_j = 0$, $v^T v_j = 0$, $\forall j = 1, \ldots, i-1$. The vectors $u_1, \ldots, u_i$ and $v_1, \ldots, v_i$ are the principal vectors (u and v being two generic principal vectors). We denote with:

$$\theta_i^r(S_j, S_l) \tag{2}$$

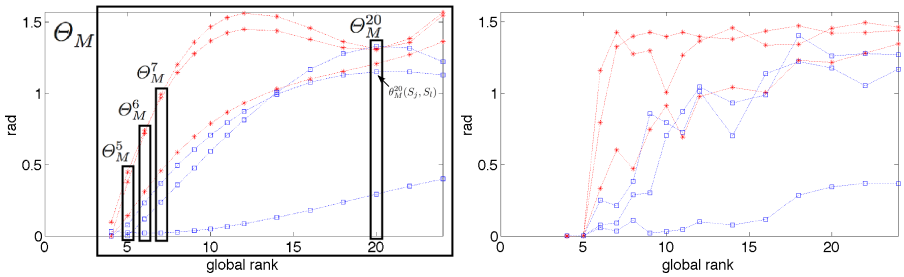
the i^{th} PA between the subspaces S_j and S_l computed when the estimated size of the global subspace is r. As j and l vary we define the set:

$$\Theta_i^r = \{\theta_i^r(S_j, S_l),\ j, l = 1, \ldots, P\} \tag{3}$$

Finally, we define:

$$\Theta_i = \bigcup_{r=1}^{r_{\max}} \Theta_i^r \tag{4}$$

where $r_{\max}$ is the upper bound of the global subspace size. For an at-a-glance overview of our notation refer to Fig. 1(a).



(a) Interpolated PAs (b) Original PAs

Fig. 1. Small random subset of the PAs of Θ_M (largest PAs) of the sequence 1R2RCT_A taken from the Hopkins155 database. PA between similar subspaces are represented with blue squares, PAs between different subspaces are represented with red asterisks.

2.2 Issues Regarding the Behaviour of Principal Angles

PAs between two subspaces are an efficient measure of orthogonality when the exact subspace bases are known. However, when the bases are estimated there are some issues that should be taken into account, especially when the exact size of the global subspace is unknown. In [12] the behaviour of PAs, computed following the LSA algorihtm, when the estimated rank r of the global subspace changes is studied. The authors explain that the trend of PAs, going from an underestimation to an overestimation of r, is overall increasing typically starting from 0 radians and ending in $\pi/2$ radians, as in Fig. 1. In the same study it is explained that despite the overall increasing trend, the PAs may have oscillations, as in Fig. 1(b), due to the fact that when the rank is underestimated the bases are not well defined, while when the rank is overestimated the extra components introduced act like noise.

In order to reduce the influence of these oscillations we propose a *polynomial interpolation* of the PAs across the different ranks. We avoid the trivially useless interpolation of order 1. The interpolation of order 2 is decreasing after its maximum, this does not fit with the increasing behaviour of the PAs. The interpolation of order 3 is able to smoothly follow the PAs trend, as shown in Fig. 1(a). Interpolation of higher degrees would adhere too much to the data making the interpolation not effective. We conducted different tests on synthetic and real sequences that confirm the PAs behavior and the reliability of the interpolation of order 3.

2.3 Rank Selection via Principal Angles Clusterization (PAC)

One of the most recognized weaknesses of LSA is the lack of robustness of the Model Selection (MS) procedure for the estimation of the rank r:

$$r = \underset{r}{\operatorname{argmin}} \left(\frac{\lambda_{r+1}^2}{\sum_{i=1}^{r} \lambda_i^2} + kr \right) \tag{5}$$

λ_i being the i^{th} singular value of $\mathbf{W}$, and k a parameter that depends on the noise of the tracked point positions. Eq. (5), is extremely sensitive to changes of the parameter k. On the other hand, k is necessary in order to deal with sequences with different amounts of noise and number of motions. In [9] the authors decided to avoid the use of MS due to the difficult task of finding a value of k that could cope with all of the sequences of the Hopkins155 database. Therefore, they fixed the global subspace size to $4N$. Fixing the global subspace size to $4N$ implies that the motions are all rigid and fully independent. Such an assumption reduces the efficiency of LSA. In order to solve this problem in [12] the authors present an algorithm named ELSA with an Enhanced Model Selection (EMS+). EMS+ consists of computing different affinity matrices, by using different k values with the MS formula, and selecting the affinity matrix with the maximum entropy. This technique allows homogeneous affinity matrices (which correspond to over- or underestimation of the rank) to be discarded, and to use an affinity matrix with the highest content of information. ELSA with EMS+ performs better than LSA with MS. Nevertheless, as the authors explain, EMS+ tends to underestimate the rank and it fails in the ideal case when the affinity matrix is binary.

The problem of the rank estimation in real cases, with noise and dependent motions, is challenging because the eigenvalue spectrum of $\mathbf{W}$ tends to become smooth and the selection of a threshold becomes a difficult task. Therefore, we decided to renounce the computation of the rank in the traditional way and we studied the distribution of the PAs in each Θ_i. The fundamental idea on which our proposal is based is that *the rank r should be selected, for each fixed i, as the one that maximizes the clusterization level of the PAs in the set Θ_i.* By clusterization we mean that the angles between similar and different local subspaces are well separated. In the ideal case (no noise and perfectly orthogonal local subspaces) the PAs would cluster around 0 and $\pi/2$. In real cases the PAs are not perfectly clustered, however, it is possible to evaluate the clusterization level for each Θ_i^r and select the one with the highest clusterization level for each i. We propose to measure the clusterization of each Θ_i^r by using a function inspired by the Linear Discriminant Analysis, we call it Principal Angles Clusterization (PAC):

$$\mathrm{PAC}(\Theta_i^r) = \frac{(\mu_a - \mu_{\mathrm{PAC}})^2 + (\mu_b - \mu_{\mathrm{PAC}})^2}{\sigma_a^{\gamma(\sigma_a)} + \sigma_b^{\gamma(\sigma_b)}} \tag{6}$$

where μ_{PAC} is the center of Θ_i^r computed as the mean of the $\mathcal{P}$ largest and smallest angles, μ_a, σ_a and μ_b, σ_b are the arithmetic means and the standard deviations of the PAs that are above and below μ_{PAC}, respectively. Our tests have shown that $\mathcal{P} = 25\%$ of the Θ_i^r gives a μ_{PAC} that is robust with respect to the presence of outliers (due to oscillations of the PAs). Note that μ_{PAC} is not computed as the mean of all the PAs to avoid biases due to the unbalanced number of representatives of one or the other class. In our experiments r goes from 2 to $8N$.

An important component of the formula is the functional exponent $\gamma(\sigma)$. If we used $\gamma(\sigma) \equiv 2$, as in the LDA formulation, the maximum of the PAC

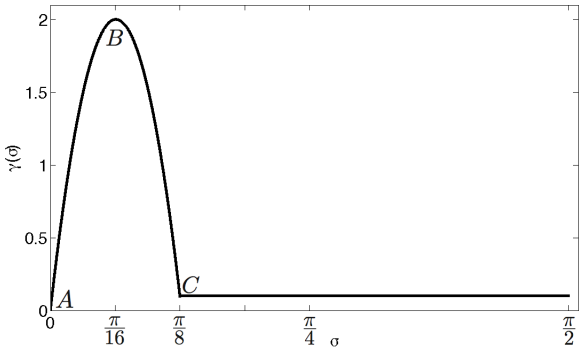


Fig. 2. $\gamma(\sigma)$ function used in the PAC formula.

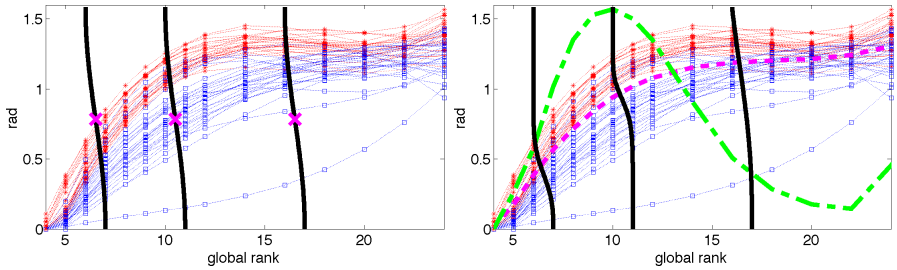
function would always be in the extremes of its domain. In fact, it was explained in section 2.2 that when $r \simeq 2$ the PAs tend to cluster around 0, hence the tiny values of the σ's that appear in the denominator of Eq. (6) would boost the PAC value, despite the fact that the μ's are very close to each other. At the other extreme, when $r \simeq r_{\max}$, the μ's increase and become well separated even though the two classes partially overlap. However, as the σ's remain smaller than 1, the global effect would be a magnification of the numerator, boosting again the PAC value. Hence, it is necessary to use a variable exponent that takes small values at the extremes while approaching to 2 for middle values.

A simple function that complies with these requirements is the following:

$$\gamma(\sigma) = \begin{cases} a_1\sigma^2 + a_2\sigma & \text{if } \sigma \leq \pi/8 \\ 0.1 & \text{if } \sigma > \pi/8 \end{cases} \tag{7}$$

The numerical coefficients a_1 and a_2 are not chosen after a tuning procedure but are determined through the following reasoning. Assuming an average case with PAs uniformly distributed, $\mu_{\text{PAC}} = \pi/4$, $\mu_a = 3\pi/8$ while $\mu_b = \pi/8$. Therefore, the upper bound of $\sigma_a, \sigma_b < \pi/8$. The numerical coefficients $a_1 = -50.63$ and $a_2 = 20.13$ define a function that fulfills the previous request making the parabola passing through the points $A \equiv (0, 0)$, $B \equiv (\pi/16, 2)$ and $C \equiv (\pi/8, 0.1)$, as shown in Fig. 2. When σ's $> \pi/8$ the angles are excessively spread and the two classes are likely to overlap. For this reason we maintain $\gamma(\sigma) \equiv 0.1$.

Summarizing, we select for the next step the set of angles in the Θ_i^r with the highest PAC value for each i. Note that the selected rank may be different for each Θ_i. This is a new interpretation of the size of the global subspace: we are not estimating the rank of $\mathbf{W}$, but we are identifying the "most expressive" dimension for each set Θ_i in terms of clusterization level. An example of the PAC function applied to a Θ_i can be seen in Fig. 3(b).



(a) $\cos^2$ functions (black) (b) CbA functions (black)

Fig. 3. Example of a random subset of the PAs of Θ_M (largest PAs, 3 rigid independent motions, hence maximum rank 12). PAs between similar subspaces are represented with blue squares, PAs between different subspaces are represented with red asterisks. The affinity functions computed at the rank $r = 6, 10, 16$, appear in black. In Fig. 3(a) the inflection point of the function is denoted with a magenta cross. In Fig. 3(b) the magenta dotted line is μ_{PAC} (which for every r it is also the inflection point of the CbA function), the green line-dot-line is the value of the PAC function.

2.4 Sum of Clusterization-based Affinity (SCbA)

Another fundamental step of manifold clustering based algorithms is to compare subspaces through an affinity measure (as a measure of (dis)similarity). In the literature it is possible to find many affinity measures with different characteristics. A discussion of different affinity measures can be found in [5].

All affinity measures applied to PAs share a common assumption: the angles between similar subspaces are always close to zero, and the angles between different subspaces are always close to $\pi/2$. None of them takes into account that the recursive definition of the PAs tends to force the angle between two subspaces to increase as we move from Θ_i^r to Θ_{i+1}^r. Moreover, none of them takes into account that the angles in a given Θ_i tend to increase when r increases, as explained in section 2.2.

In the example of Fig. 3(a) we have randomly plotted some PAs of Θ_M of the sequence 1R2RCR (Hopkins155 database). In black it is possible to see the $\cos^2$ function. The $\cos^2$ function always has the same shape and the inflection point (magenta cross) is always in the same position, regardless of the rank to which it is applied. As a consequence of this *rigidity*, if the estimated rank is $r = 6$ all of the PAs have an affinity value that falls before the inflection point. Opposite cases are when $r = 10$ and $r = 16$, in which most of the PAs have an affinity value after the inflection point. Therefore, the $\cos^2$ function, as well as any other rigid function, is very sensitive to the rank estimation.

The affinity measure that we propose is able to adapt itself to the distribution of the PAs in Θ_i^r, so that it minimizes the negative effects of a wrong rank estimation and it emphasizes the difference between similar and different subspaces. We define the not normalized Clustering-based Affinity ($\overline{\text{CbA}}$) between

two generic subspaces S_j, S_l, for $j, l = 1, \ldots, P$, for a given Θ_i^r as the function $\overline{\text{CbA}} : \Theta_i^r \to \mathbb{R}^+$,

$$\overline{\text{CbA}}(\theta_i^r(S_j, S_l)) = \exp\left(-\frac{\beta - 1}{\beta}\left(\frac{\theta_i^r(S_j, S_l)}{\alpha}\right)^\beta\right) \tag{8}$$

where θ_i^r is the i^{th} principal angle computed at the rank r. α and β are the two positive parameters ($\alpha > 0$, $\beta \geq 2$) that allow the function to change in relation to the distribution of the PAs. We can now define the normalized Clustering-based Affinity (CbA) as follows:

$$\text{CbA}(\theta_i^r(S_j, S_l)) = \frac{\overline{\text{CbA}}(\theta_i^r(S_j, S_l)) - \min(\overline{\text{CbA}})}{\max(\overline{\text{CbA}}) - \min(\overline{\text{CbA}})} \tag{9}$$

The arrangement of the parameters of Eq. (8) has been chosen so that CbA has a negative first derivative over all its domain, while its second derivative is negative for $\theta < \alpha$, positive for $\theta > \alpha$ and equal to zero for $\theta = \alpha$. We propose to set $\alpha = \mu_{\text{PAC}}$ so that the inflexion point occurs at the estimated center of the distribution. In this way the function is always stretched or compressed in order to fit the distribution of the PAs. The β parameter is used in order to emphasize the differences between similar and different subspaces in an automatic fashion. In fact, β controls the slope of the function: the higher the β the steeper the slope. We would like an affinity function with a steep slope when the PAs are well clustered and a more gentle slope when the clusterization is not clear. A natural candidate for β is $\beta = \text{PAC}(\Theta_i^r) \cdot \mathcal{F}$, as the PAC function gives a measure of how well clustered the two groups are and how far away the two centroids are. $\mathcal{F}$ is a constant, a boosting factor, that we use in order to give more or less importance to β. In all of our experiments we have used $\mathcal{F} = 5$ which has empirically shown to be a suitable factor.

In Fig. 3(b) we plot three CbA functions applied to different ranks r within the set Θ_M. In this picture it is possible to appreciate that, thanks to the parameter α, the inflection point changes so that it always corresponds to the μ_{PAC} value, hence minimizing the effect of possible errors in the choice of the rank r. Moreover, thanks to the parameter β the slope of CbA changes depending on how well the small angles are separated from the large angles.

The final affinity between two subspaces is defined as the normalized weighted Sum of CbA (SCbA):

$$\text{SCbA}(S_j, S_l) = \frac{\sum_{i=1}^{M} \text{CbA}(\theta_i^r(S_j, S_l))\text{PAC}(\Theta_i^r)}{\sum_{i=1}^{M} \text{PAC}(\Theta_i^r)} \tag{10}$$

M being the minimum size between subspaces S_j and S_l. In this work we have not investigated the estimation of the local subspace size which was fixed to 4. Note that by weighting the CbA values by the PAC function we give more importance to Θ_i^r where the angles between similar and different subspaces are well separated.

SCbA respects the axioms of an affinity function proposed in [5]:

– **symmetry**: from Eq. (10) we see that $\mathrm{SCbA}(S_j, S_l) = \mathrm{SCbA}(S_l, S_j)$;
– **orthogonality consistency**: given that

$$S_j \perp S_l \iff \theta_i^r(S_j, S_l) = \pi/2 \tag{11}$$

$\forall i = 1, \ldots, M$, from Eq. (9) and (10) it follows that:

$$\mathrm{SCbA}(S_j, S_l) = 0 \tag{12}$$

– **inclusion consistency**: given that

$$S_j \subseteq S_l \iff \theta_i^r(S_j, S_l) = 0 \tag{13}$$

$\forall i = 1, \ldots, M$, from Eq. (10) it follows that:

$$\mathrm{SCbA}(S_j, S_l) = 1 \tag{14}$$

2.5 Summary of our proposal

In this section we summarize our proposal: LSA+PAC+SCbA.

1. Build a trajectory matrix $\mathbf{W}$;
2. for $r = 2$ to $r_{\max}$ (in our tests $r_{\max} = 8N$)
   (a) 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;
   (b) exploiting the fact that in the new space (global subspace) most points and their closest neighbours lie in the same subspace, compute by SVD the local subspaces generated by each trajectory and its nearest neighbours (NNs);
   (c) compute PAs between all of the subspaces;
3. smooth the PAs;
4. apply PAC to find the best r for each Θ_i $(i = 1 \ldots M)$;
5. apply SCbA to build the affinity matrix $\mathbf{A}$;
6. cluster $\mathbf{A}$ by K-means in order to have the final motion segmentation.

More details can be found in the source code available at: `http://eia.udg.es/~zappella`.

3 Experiments

We tested our proposal on the 155 real sequences of the Hopkins155 database and we compared our performances with: LSA + MS (with $k = 10^{-7.5}$, best k value as explained in [12]), ELSA EMS+ (results extracted using available code [12]), LSA $4N$ (results taken from [5]), MSC+NSI (results taken from [5]), ALC (results taken from [5]), and SSC (results taken from [3]).

Table 1 shows the average misclassification rates and the standard deviations of each method. The misclassification rates are presented for each type of video

Table 1. State of the art comparison. Misclassification rates on the Hopkins155 database. In brackets the number of sequences for each type of video. NA stands for value not available.

2 Motions	Checkboards(78)		Articulated(11)		Traffic(31)		All types(120)	
Method	% Avg	% Std	% Avg	% Std	% Avg	% Std	% Avg	% Std
LSA + MS	5.15	9.61	3.65	4.29	4.95	8.66	4.96	8.96
LSA 4N	2.57	6.79	4.10	6.47	5.43	11.17	3.45	8.14
ELSA EMS+	2.20	7.19	2.32	3.87	5.58	10.89	3.08	8.17
ALC	1.49	4.58	10.70	15.00	1.75	1.83	2.40	6.35
MCS + NSI	3.75	7.89	8.05	8.51	1.69	7.00	3.61	7.84
SSC	1.12	NA	0.62	NA	0.02	NA	0.82	NA
Our Proposal	**1.00**	5.64	**1.75**	3.13	**0.57**	1.06	**0.96**	4.67
3 Motions	Checkboards(26)		Articulated(2)		Traffic(7)		All types(35)	
Method	% Avg	% Std	% Avg	% Std	% Avg	% Std	% Avg	% Std
LSA + MS	19.09	13.02	9.57	13.54	16.06	5.72	17.94	11.91
LSA 4N	5.70	10.89	7.25	9.30	25.30	19.05	9.71	14.71
ELSA EMS+	8.76	15.18	6.38	9.03	6.354	12.36	8.15	14.14
ALC	5.00	9.14	21.08	28.87	8.86	13.16	6.69	11.48
MCS+NSI	2.29	5.73	6.38	9.03	1.67	1.51	2.87	5.28
SSC	2.97	NA	1.42	NA	0.58	NA	2.45	NA
Our Proposal	**2.41**	8.05	**3.72**	5.26	**1.11**	1.87	**2.22**	7.03

sequence (checkboards, articulated and traffic). Firstly, it is possible to see that our proposal outperforms every LSA-based technique proving that our method improves the weaknesses of LSA. Also when the other techniques are taken into account, our proposal has, together with SSC, the lowest misclassification rates both with 2 and 3 motions (the average misclassification rate of our proposal on the whole Hopkins155 database is of 1.25%). However, we would like to remark that for our algorithm the only two free parameters, ($\mathcal{P}$ and $\mathcal{F}$) were fixed for the whole database whereas it is not clear from [3] whether the results of SSC where obtained with a fixed set of parameters or each sequence required a different set.

In Fig. 4 the histogram of the misclassification rates of our proposal is presented. The majority of the sequences, 134, has a misclassification rate smaller than 1%, and the total number of sequences with a misclassification rate below 5% is 145. The median misclassification of every group is always 0% with the exception of the articulated with 3 motions group where the median is equal to the mean (due to the presence in this group of only 2 sequences).

As far as the computational time is concerned, the bottle neck or our method is the interpolation process of all the PAs. In fact, on the whole Hopkins155 database our proposal required 147600 seconds, of which 143775 were spent for the interpolation (Matlab implementation on Quad-Core AMD @ 2.4GHz, with 16 GB RAM).

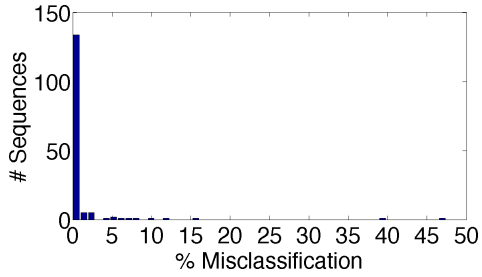


Fig. 4. Histogram of the misclassification rate of our proposal.

In order to verify how our proposal performs on a different database, we tested it on synthetic sequences with 2, 3, 4 and 5 rigid and independent motions (10 different sequences for each number of motions) and an increasing noise level. Specifically, each sequence is composed of 50 frames, with rigidly rotating and translating cubes. Each cube has 56 tracked features. Then we created 2 additional databases adding noise with standard deviations of 0.5 and 1 pixel to the tracked feature positions. In total we used 150 synthetic sequences. The misclassification rates are shown in table 2. All the misclassification rates are smaller than 1%. For a given number of motions the misclassification remains rather stable even when the noise level increases. Moreover, the behaviour of our proposal even with 4 and 5 motions (more than the motions in the Hopkins155 database) is very satisfactory.

Table 2. Misclassification rates on synthetic sequences with 2, 3, 4 and 5 motions and increasing noise level. In brackets the number of sequences for each type of video.

Motions	2(10)		3(10)		4(10)		5(10)	
Our Proposal	% Avg	% Std	% Avg	% Std	% Avg	% Std	% Avg	% Std
$\sigma_{\text{noise}} = 0$	0	0.0	0.24	0.31	0.36	0.35	0.68	0.39
$\sigma_{\text{noise}} = 0.5$	0.09	0.28	0.12	0.25	0.31	0.22	0.75	0.43
$\sigma_{\text{noise}} = 1$	0.27	0.60	0.24	0.31	0.31	0.22	0.75	0.36

4 Conclusions and Perspectives

We presented two improvements for motion segmentation based on manifold clustering. The first improvement is a new way of selecting the global subspace size based on the analysis of the principal angles clusterization, such that the selected size is the one where the principal angles between similar and different subspaces are best separated. The second improvement is a new affinity measure that is automatically able to adapt itself in order to fit the distribution of the

principal angles. The major achievement of this measure is that it can deal with every distribution of principal angles minimizing the effect of an erroneous rank estimation of **W** while maximising the distance between similar and different local subspaces. The results of our experiments show that, even without changing the value of the only two free parameters that we have, the misclassification rates of our proposal are among the lowest in the literature.

Future works should aim to reduce the computational time of the algorithm by adopting other ways for reducing the principal angles oscillations. Moreover, better segmentations could be achieved by extending our algorithm to the estimation of the local subspaces size.

Acknowledgement. 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.

References

1. Absil, P.A., Edelman, A., Koev, P.: On the largest principal angle between random subspaces. Linear Algebra and its Applications **414** (2006) 288–294
2. Björck, A., Golub, G.H.: Numerical methods for computing angles between linear subspaces. Mathematics of Computation **27** (1973) 579–594
3. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Proc. CVPR IEEE. (2009) 2790–2797
4. Pinho da Silva, N., Costeira, J.P.: Subspace segmentation with outliers: a grassmannian approach to the maximum consensus subspace. In: Proc. CVPR IEEE. (2008) 1–6
5. Pinho da Silva, N., Costeira, J.c.P.: The normalized subspace inclusion: Robust clustering of motion subspaces. In: IEEE I. Conf. Comp. Vis. (2009) 1444–1450
6. Lauer, F., Schnrr, C.: Spectral clustering of linear subspaces for motion segmentation. In: IEEE I. Conf. Comp. Vis. (2009) 678–685
7. Rao, S.R., Tron, R., Vidal, R., Ma, Y.: Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: Proc. CVPR IEEE. (2008) 1–8
8. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. J. Comput. Vision. **9** (1992) 137–154
9. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: Proc. CVPR IEEE (2007) 1–8
10. Vidal, R., Hartley, R.: Motion segmentation with missing data using powerfactorization and gpca. In: Proc. CVPR IEEE **2** (2004) 310–316
11. Yan, J., Pollefeys, M.: A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. IEEE Trans. Pattern Anal. Machine Intell. **30** (2008) 865–877
12. Zappella, L., Lladó, X., Provenzi, E., Salvi, J.: Enhanced Local Subspace Affinity for Feature-Based Motion Segmentation. In: Pattern Recognition (2010 in-press)