Subspace segmentation with outliers: a Grassmannian approach to the maximum consensus subspace*

Nuno Pinho da Silva† João Paulo Costeira‡
ISR-Instituto Superior Técnico
Torre Norte - 7 Piso, Av.Rovisco Pais, 1, 1049-001 Lisboa, Portugal
(nmps,jpc)@isr.ist.utl.pt

Abstract

Segmenting arbitrary unions of linear subspaces is an important tool for computer vision tasks such as motion and image segmentation, SfM or object recognition. We segment subspaces by searching for the orthogonal complement of the subspace supported by the majority of the observations, i.e.*, the maximum consensus subspace. It is formulated as a grassmannian optimization problem: a smooth, constrained but nonconvex program is immersed into the Grassmann manifold, resulting in a low dimensional and unconstrained program solved with an efficient optimization algorithm. Nonconvexity implies that global optimality depends on the initialization. However, by finding the maximum consensus subspace, outlier rejection becomes an inherent property of the method. Besides robustness, it does not rely on prior global detection procedures (*e.g.*, rank of data matrices), which is the case of most current works. We test our algorithm in both synthetic and real data, where no outlier was ever classified as inlier.*

1. Introduction

We segment linear subspaces by finding the maximum consensus subspace ($\mathcal{MCS}$), *i.e.*, the subspace with the largest number of inliers. Its null space ($\mathcal{MCS}^\perp$) contains, by definition, the majority of zero point projections, which we use as the criterion to searching for the $\mathcal{MCS}$ (see figure 1). We call *arbitrary union* to a mixture of an unknown number of linear subspaces of unknown dimensions, with arbitrary intersections and containing outliers. When segmenting, we seek *nontrivial linear subspaces*: d-dimensional subspaces with, at least, $d+1$ features' consensus. Since our patterns are nontrivial subspaces, we define an *outlier* to be a trivial $1D$ subspace (or the unique feature

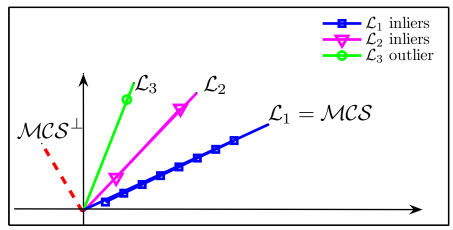

Figure 1. An arbitrary union of linear subspaces, containing 2 nontrivial 1D subspaces ($\mathcal{L}_1$ and $\mathcal{L}_2$) and 1 trivial 1D subspaces,*i.e.* outlier ($\mathcal{L}_3$). We find the $\mathcal{MCS}$ by searching for the $\mathcal{MCS}^\perp$.

supporting it). Finding the subspace with maximum support discriminates trivial solutions, rejecting outliers. Furthermore, the maximum consensus is a global criterion that does not need global data explanation, such as rank detection. We efficiently search for the $\mathcal{MCS}$ by formulating it as an optimization problem on the Grassmann manifold (or Grassmannian). The $d(n-d)$-dimensional Grassmannian, denoted by $\mathcal{G}_{n,d}$, is the space of all d-dimensional linear subspaces of a n-dimensional vector space (*e.g.*, the euclidean $\mathcal{R}^n$). The grassmannian optimization framework allows to perform guided search by employing a geodesic descend method, which provably converges to a local optimum, with superlinear rate near the solution.

We apply our generic methodology to multiple motion segmentation with outliers, an active, and theoretically challenging research topic in computer vision. Assuming affine cameras, if the objects have independent motions, the segmentation is obtained [2]. Extensions to degenerate and articulated motions have been recently addressed in [10, 11]. However, all these methods rely on $rank$ detection of some data matrix, which is highly prone to errors in the presence of outliers. Therefore, some additional procedures, such as RANSAC (*e.g.*, [7, 8]) or voting schemes (*e.g.*, [4]), must be considered. There are two main reasons why segmenting arbitrary unions is still an open issue: none of the approaches is provably correct and both are computationally inefficient,

*This work was partially supported by FCT, under ISR/IST plurianual funding (POSC program, FEDER)
†Supported by FCT, under grant SFRH / BD / 24879 / 2005
‡Partially supported by FCT, under grant PTDC/EEA-ACR/72201/2006, "MODI - 3D Models from 2D Images".

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

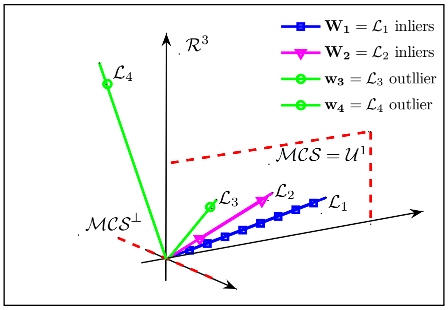

Figure 2. An arbitrary union in $\mathcal{R}^3$ where the $\mathcal{MCS}$ is, itself, an arbitrary union.

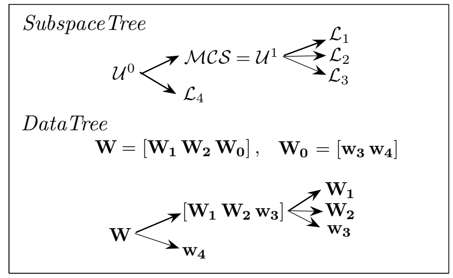

Figure 3. Segmentation tree for the union in Figure 2: clustered unions are segmented by recursion while the segments are found iteratively (superscripts refer to levels of recursion).

even in the noiseless case.

Our approach differs from the previous methods by efficiently searching the $\mathcal{MCS}^\perp$, providing a robust criterion for segmenting without relying on $rank$ detection. To be precise:

- *Contributions*: Finding the $\mathcal{MCS}$ is a nonsmooth, nonconvex problem, solved by providing an *analytic* smooth equivalent cost and immersing it into the Grassmannian, resulting in a nonconvex, unconstrained program in $\mathcal{R}^{d(n-d)}$. We use a combination of Steepest descend and Newton's algorithm, with Armijo's stepsize, to follow the grassmannian geodesics (guided search). Such method *provably* converges to a local optimizer in a finite number of steps, with superlinear rate near the solution (efficiency). The $\mathcal{MCS}$ criterion allows to classify all the inliers on some d-dimensional subspace, thus avoiding rank computation, *i.e*. classifying inliers amounts to looking for zeros in the cost vector.
- *Drawbacks*: The tradeoff is nonconvexity. Thus global optimality is not guaranteed, leaving an avenue for future work.

In the presence of noise, we must handle a local decision (detecting zeros), instead of a global one (detecting rank), making the choice of threshold easier: we may reject inliers (a local error) and still provide correct segmentation while computing the wrong rank (a global error) jeopardizes it. Summarizing, neither robustness nor segmentation rely on threshold setting, but rather on cost design.

2. Subspace segmentation with outliers

We assume that the data matrix $\mathbf{W} \in \mathcal{R}^{n\times N}$ has the following canonical (segmented) form

$$\mathbf{W} = [\mathbf{W_1} \ldots \mathbf{W_m}\, \mathbf{W_0}], \tag{1}$$

where $\mathbf{W_j} \in \mathcal{R}^{n\times N_j}$ support nontrivial linear subspaces $\mathcal{L}_j = span\{\mathbf{W_j}\}$ for all $j \neq 0$ and $\mathcal{L}_0 = span\{\mathbf{W_0}\}$ is a trivial subspace, that is, $\mathbf{W_0}$ collects N_0 points supporting N_0 trivial 1D linear subspaces,*i.e*. N_0 outliers. This means there is, at least, one nontrivial subspace in the union (feasibility condition). The methodology follows the principle of RANSAC: find the maximum consensus d-dimensional subspace. The segmentation follows iteratively until only trivial solutions exists in the union $\mathcal{U}^r$, *i.e*. outliers are the "left overs" of the segmentation process. Whenever a nontrivial solution is found, we get into it by recursion (the superscript r identifies the level of recursion), looking for nontrivial subspaces of lower dimensions, thus finding degeneracies or rejecting outliers. Inliers are classified and taken out of the observations matrix, and outliers are reintroduced in the data matrix because they may be supporting some other nontrivial subspace in the union. There are three (optional) parameters: the minimum support for a d-dimensional segment ($min\_consensus$) and its maximum (d_{max}) and minimum (d_{min}) dimension, with default values $min\_consensus = d+1$, $d_{max} = \min\{n, N\} - 1$ and $d_{min} = 1$.

As an example, consider the union in Figure 2. Let $\mathbf{W} \in \mathcal{R}^{n\times N}$, with $n = 3$, and $\mathcal{U}^0 = span\{\mathbf{W}\}$ as in Figure 3. Assuming the default values for the parameters, in the recursion level $r = 0$ we have $d^0 = d_{max} = 2$ and $min\_consensus^0 = 3$, so $\mathcal{MCS} = \bigcup_{j=1}^{3} \mathcal{L}_j$. Thus, the cost vector collecting the algebraic distances of all points to $\mathcal{MCS}$ have minimal cardinality among all cost vector maps with domain in the Grassmannian $\mathcal{G}_{3,1}$.

Having found a nontrivial solution, we need to check if this is a union or a segment, *i.e*. see if there are lower dimensional nontrivial subspaces and/or outliers. Recognizing clustered unions is done by calling the algorithm recursively. Hence, for $r = 1$, we have $d^1 = d^0 - 1 = 1$, $min\_consensus^1 = 2$, and data lies on $\mathcal{U}^1 = span\{[\mathbf{W_1}, \mathbf{W_2}, \mathbf{w_3}]\}$ (see Figure 3). In $\mathcal{U}^1$ the $\mathcal{MCS} = \mathcal{L}_1$ (as in Figure 1, where now the $\mathcal{MCS}^\perp$ is a plane because the ambient space is $\mathcal{R}^3$). Remark that, if $N_1 = N_2$, both $\mathcal{L}_1$ and $\mathcal{L}_2$ have maximum support and one of them is segmented depending on the initialization (*c.f*. Section 3.3.1). Recurring once more ($r = 2$), we have $d^2 = d^1 - 1 = 0$ and the algorithm returns to the previous level $r = 1$. This means that $\mathcal{L}_1$ is a segment of $dim(\mathcal{L}_1) = d^1$, so the N_1 points supporting it are classified and removed from the data matrix, resulting in $\mathcal{U}^1 = span\{[\mathbf{W_2}, \mathbf{w_3}]\}$. This is the generic mechanism to classify segments of arbitrary dimen-

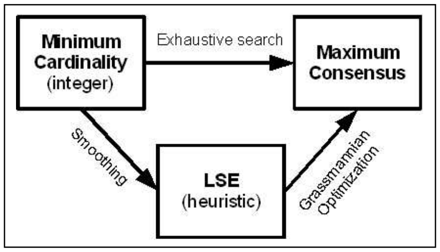

Figure 4. Grassmannian approach to the $\mathcal{MCS}$: formulating the problem as a smooth, low dimensional, unconstrained optimization program allows to employ geodesic descend algorithms, guaranteeing analytic convergence to a local optimizer with superlinear rate near the solution.

sion: in any level r, if a nontrivial d^r-dimensional subspace is a segment, all lower dimensional subspaces in it are trivial subspaces. In other words, the data matrix supporting a segment breaks the feasibility condition. Therefore, the iterative path will lead to $d^r = 0$, providing a unique condition for classifying: level r is infeasible, *i.e.* seen from level $r-1$, $d^r = 0$ and no features were classified in deeper levels of recursion.

Returning to our example and proceeding as before, we classify and remove the N_2 observations supporting $\mathcal{L}_2$: $dim(\mathcal{L}_2) = d^1$, leaving $\mathcal{U}^1 = span\{[\mathbf{w_3}]\}$. This is a trivial solution, and so we return to level $r = 0$. Since level $r = 1$ was feasible, segmentation proceeds by updating the union $\mathcal{U}^0 = span\{[\mathbf{w_3}\,\mathbf{w_4}]\}$ without classifying data. Recalling that $d^0 = 2, \mathcal{U}^0$ is now a trivial solution, because it contains only outliers (trivial 1D subspaces). Thus, iterating leads to $d^0 = 0$, ending the segmentation.

Note that using the $\mathcal{MCS}$ criterion ensures that all inliers on some d-segment are well classified. Hence, by varying the dimension of subspaces, iteratively if no admissible solution is found, or recursively otherwise, the current approach provides the intrinsic dimension of the segment supported by the classified data. In short, our method segments arbitrary unions without computing $rank$, as shown by the example.

3. Finding the $\mathcal{MCS}$: a Grassmannian approach

The Grassmannian approach allows to finding the $\mathcal{MCS}$ efficiently, a crucial distinction between our methodology and random sampling or voting schemes (Figure 4). We provide a guided search method by following the grassmannian geodesics along the maximum consensus direction[1].

3.1. Minimum cardinality formulation

Define the cost vector map as the algebraic distance of all points to a d-dimensional linear subspace $\mathcal{L}$, *i.e.* $\mathbf{v}$: $\mathcal{G}_{n,n-d} \to \mathcal{R}_+^N$ such that

$$\mathbf{v}(\mathcal{L}^\perp) = diag(\mathbf{W^T Q Q^T W}), \quad (2)$$

where $diag(\cdot)$ extracts the main diagonal of the argument and $\mathbf{Q}$ is an orthonormal basis (a representative) of $\mathcal{L}^\perp$. By the uniqueness of the orthogonal projector $\mathbf{QQ^T}$, this map is such that

$$\boxed{\mathbf{v_i}(\mathcal{L}^\perp) = 0 \Leftrightarrow \mathbf{w_i} \in \mathcal{L}.} \quad (3)$$

Therefore, the maximum number of zeros in $\mathbf{v}$ is a maximum consensus criterion. Hence, finding the $\mathcal{MCS}$ is equivalent to minimizing the cardinality of $\mathbf{v}$ over the Grassmannian $\mathcal{G}_{n,n-d}$, *i.e.*

$$\mathcal{MCS}^\perp = \arg\min\{card(\mathbf{v}(\mathcal{L}^\perp)) :\ \mathcal{L}^\perp \in \mathcal{G}_{n,n-d}\}. \quad (4)$$

Minimizing the cardinality of a vector has a variety of applications and, in general, is a NP-Hard problem. In some special cases, it may be solved by the $\ell_1 - norm$ [6]. Unfortunately, in our case, the $\ell_1 - norm$ minimizer is the $span$ of the $n-d$ least significant left singular vectors of the data matrix, taking all union into account without segmenting.

3.2. Smooth problem formulation

To avoid exhaustive search, we built a smooth heuristic for counting zeros: the LSE map. We start by showing that there is a real number λ^* shaping the sum of exponential functions to the constraint set in such way that, if $\mathbf{v}^*$ minimizes the cardinality then it maximizes the sum of the exponentials.

Proposition 1. *Consider the map* $f : \mathcal{R}_{++} \times \mathcal{R}_+^N \to [0, N]$,

$$f(\lambda, \mathbf{v}) = \sum_{i=1}^{N} e^{-\lambda \cdot \mathbf{v_i}} = \sum_{i=1}^{N} f_i(\lambda, \mathbf{v_i}), \quad (5)$$

and assume that $\mathbf{v}^* = \arg\min\{card(\mathbf{v}) :\ \mathbf{v} \in \mathcal{S}\}$ *with* $\mathcal{S} \subseteq \mathcal{R}_+^N$ *compact. Let* $\mathcal{I}(\mathbf{v}) = \{i :\ \mathbf{v_i} > 0\}$. *Then,* $\exists\ \lambda^*, v_0 > 0$ *verifying*

$$\lambda^* \geq \frac{\ln(N+1)}{v_0},\ v_0 = \min\{\mathbf{v_i} :\ \mathbf{i} \in \mathcal{I}(\mathbf{v})\} \quad (6)$$

such that $\mathbf{v}^* = \arg\max\{f(\lambda^*, \mathbf{v}) :\ \mathbf{v} \in \mathcal{S}\}$.

Proof. Let $card(\mathbf{v}^*) = l \leq N$. Then λ^* must verify

$$f_i(\lambda^*, \mathbf{v_i}) \leq \frac{1}{l+1}, \forall\, i \in \mathcal{I}(\mathbf{v}), \quad (7)$$

for any $\mathbf{v} \in \mathcal{S}$. Since $\frac{1}{l+1} \geq \frac{1}{N+1}$, $\forall\, l \leq N$, choose λ^* such that $f_i(\lambda^*, \mathbf{v_i}) \leq \frac{1}{N+1}$, $\forall i \in \mathcal{I}(\mathbf{v})$. Solving for λ^* results in $\lambda^* \geq \frac{\ln(N+1)}{v_0}$. Setting v_0 as in (6) provides a finite number verifying $f(\lambda^*, \mathbf{v}^*) \geq f(\lambda^*, \mathbf{v}), \forall\, \mathbf{v} \in \mathcal{S}$. □

Note that Mangasarian proved the existence of λ^* in [5]. Our proof differs in form and provides an expression for it. However, except for special cases, finding v_0 in (6) is not an easy task. Generically, we suggest $v_0 = \min\{\mathbf{v^+}_\mathbf{i} :\ \mathbf{i} \in \mathcal{I}(\mathbf{v}^+)\}$, where $\mathbf{v}^+$ is the $\ell_1 - norm$ minimizer over $\mathcal{S}$.

Let $\mathcal{S}$ be the image of the Grassmannian through the continuous cost vector map (2). Since the $\mathcal{G}_{n,n-d}$ is compact, $\mathcal{S}$ is compact. Setting v_0 using $\mathbf{v}^+ = \mathbf{v}(\mathcal{L}^{\perp +})$, where $\mathcal{L}^{\perp +}$ is the $\ell_1 - norm$ minimizer, and computing λ^* from (6), apply Proposition 1 to built $f(\lambda^*, \mathbf{v}(\mathcal{L}^\perp))$. Remark

[1] Informally, geodesics are generalizations of straight lines for curved spaces, in the sense that the straight lines are the euclidean geodesics.

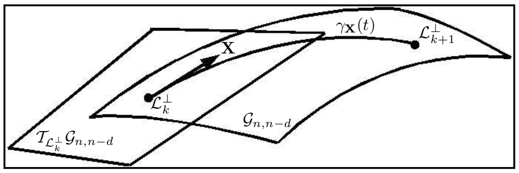

Figure 5. Geodesic descend method: $\mathcal{L}_{k+1}^{\perp} = \gamma_{\mathbf{X}}(\alpha)$ is found by following the geodesic $\gamma_{\mathbf{X}}(t)$ emanating from the current point $\mathcal{L}_k^{\perp}$ along the tangent vector $\mathbf{X} \in \mathcal{T}_{\mathcal{L}_k^{\perp}}\mathcal{G}_{n,n-d}$, where $t = \alpha$ is the time (Armijo's stepsize) when the subspace $\gamma_{\mathbf{X}}(\alpha)$ improves the cost function.

that, in practice, λ^* may be quite large, resulting in numerical ill-conditioning. To prevent flatness of the gradient at points away from local optima, we define the map $LSE : \mathcal{G}_{n,n-d} \rightarrow \mathcal{R}$

$$LSE(\mathcal{L}^{\perp}) = \ln \sum_{i}^{N} e^{-\lambda^* \mathbf{v_i}(\mathcal{L}^{\perp})}, \tag{8}$$

solving the problem

$$\boxed{\mathcal{MCS}^{\perp} = \arg\max\{LSE(\mathcal{L}^{\perp}) : \mathcal{L}^{\perp} \in \mathcal{G}_{n,n-d}\},} \tag{9}$$

to find a (possibly global) optimum for Problem (4). The logarithm induces a fractional form on the gradient favoring stationary points with zeros in $\mathbf{v}(\mathbf{Q})$, as desired. Note that the Problem (9) is equivalent to maximizing $f(\lambda^*, \mathbf{v}(\mathcal{L}^{\perp}))$ over $\mathcal{G}_{n,n-d}$. In summary, instead of minimizing the cardinality of the cost vector, we solve Problem (9) to find the $\mathcal{MCS}$ with efficient optimization methods.

3.3. Grassmannian optimization

Instead of modeling the Grassmannian by some smooth parameterization, suppose there is a way to transport Problem (9) into the constraint set. In this *intrinsic* scenario, we would achieve accuracy w.r.t. the constraints, dimensionality reduction and simplification of processes (constrained vs. unconstrained problems). As an example, consider the $\mathcal{G}_{10,8}$ and note that $dim(\mathcal{G}_{10,8}) = 16$: an intrinsic optimization method solves an unconstrained problem over $\mathcal{R}^{16}$, finding an exact representative of some subspace in $\mathcal{G}_{10,8}$; while an extrinsic method works over constrained $\mathcal{R}^{80}$ (or $\mathcal{R}^{55}$, if projectors were the variables), guaranteeing the constraints up to a tolerance. In short, the intrinsic approach simplifies the problem and increases the accuracy of the solution (in a numerical sense).

We follow the work of Edelman *et al.* [3] to obtain the intrinsic formulation for problem (9), and solve it by an unconstrained optimization method, switching from the *Steepest descent* to the *Newton's Method* in a neighborhood of a local maximum, converging superlinearly. The stepsize is chosen by Armijo's rule, applied to Grassmannian's geodesics (Figure 5). It is proven that such unconstrained optimization method always converges to a local optimizer, with better global properties than the pure Newton's algorithm, although global optimality is not guaranteed [1].

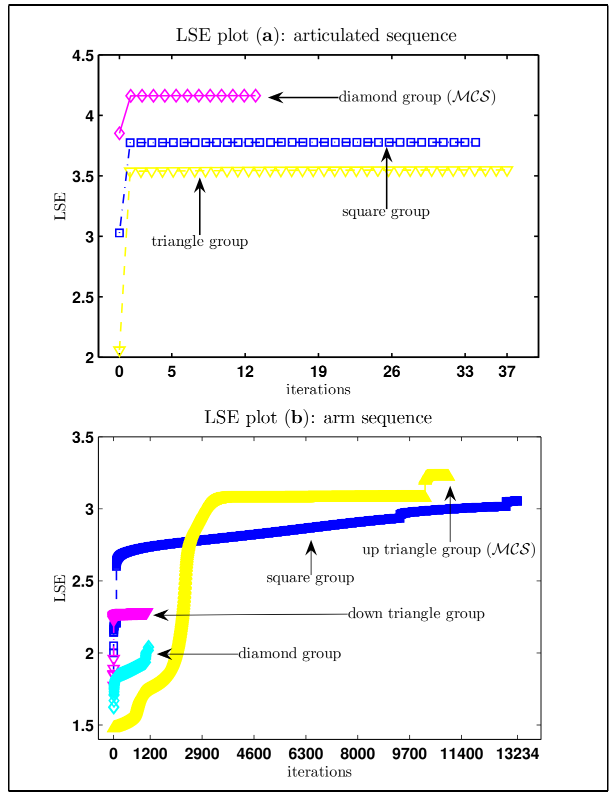

Figure 6. LSE convergence plots for real data (Figure 8): (**a**) with a good initialization, the grassmannian approach converges superlinearly;(**b**) a poor initialization makes the optimization algorithm switching between Steepest descend and the Newton's method, severely increasing the number of iterations (expected since minimizing the cardinality is NP-Hard), though convergence is guaranteed in a finite number of steps.

Since the LSE is a maximum consensus heuristic, following the geodesics along the maximum consensus direction amounts to plugging the intrinsic entities (*c.f.* Appendix) into the unconstrained optimization algorithm, thus searching for the $\mathcal{MCS}$ efficiently (Figure 6).

3.3.1 Initialization

To obtain the initial estimate, we use a k-nearest neighbors algorithm, with $k = d + 1$ and similarity function

$$\max |\tilde{\mathbf{w}}_{\mathbf{i}}{}^{\mathbf{T}} \tilde{\mathbf{w}}_{\mathbf{k}}|, \;\; \tilde{\mathbf{w}}_{\mathbf{i}} = \frac{\mathbf{w_i}}{\|\mathbf{w_i}\|_2} \tag{10}$$

evaluated over all observations $\mathbf{i}, \mathbf{k} = 1, \ldots, N$. The initial point is the orthogonal complement of some d-dimensional subspace spanned by the nearest neighbors. This is neither equivalent to proximity on $\mathcal{G}_{n,n-d}$ nor robust. Intuitively, points lying on different subspaces are likely to be rejected from the nearest neighbors, though it's easy to provide counter-examples. Neighbors supporting the same subspace is a sufficient, but *not* necessary, condition for optimality, in the sense that the solution is the $\mathcal{MCS}^{\perp}$ on which the nearest neighbors lie, assuring correct segmentation and rejecting outliers. In other words, unlike wrong rank detection, which surely ruins segmentation, converging to a local optimum may only alter the order by which segments are identified, producing a different, yet correct, subspace tree.

4. Motion segmentation application

We apply our algorithm to segmenting linear subspaces spanned by the tracked trajectories of points belonging to moving objects and observed by affine cameras. Let N be the number of features and F the number of frames. The observation matrix $\mathbf{W} \in \mathcal{R}^{2F\times N}$ is such that

$$\mathbf{w_i} = [u_i^1 \, \dots \, u_i^F \, , \, v_i^1 \, \dots \, v_i^F]^T, \, \mathbf{i} = 1, \dots, N \quad (11)$$

$$\left[u_i^f \; v_i^f\right]^T = \mathbf{A}_{2\times4}^f \mathbf{x_i}, \; \mathbf{x_i} \in \mathcal{P}^3, \; f = 1, \dots, F \quad (12)$$

where $\mathbf{A}_{2\times4}^f$ is the affine camera matrix at frame f. Under affine projections, the canonical form (1) provides the segmentation of an arbitrary mixture of trajectories from articulated, and/or independently rigid, moving shapes, with possible degeneracies [11]. The algorithm is evaluated on both synthetic and real data.

4.1. Identification in the presence of noise

When the observations are corrupted by noise, we cannot rely on null entries of the $\mathbf{v}$ map (2). Therefore, we relax the optimal condition (3) to

$$\mathbf{v_i}(\mathcal{L}^\perp) < \xi \Leftrightarrow \mathbf{w_i} \in \mathcal{L} \quad (13)$$

with $\xi \gtrsim 0$.

With $\mathcal{L}^\perp$ given by the Grassmannian approach, classifying inliers amounts to searching null entries in the cost vector. A wide threshold jeopardizes the segmentation, while if the threshold is too conservative, inliers may be misclassified as outliers and/or over-segmentation may occur. However, misclassifying inliers as outliers still provides a correct segmentation, as long as there are, at least, $d+1$ observations verifying (13) for each d-dimensional segment. Over-segmentation may be solved by (*i*) a merge criterion, *e.g.* any metric on the Grassmannian [3], at the cost of another threshold, or (*ii*) by increasing the support of the d-dimensional segments, *i.e.* $min\_consensus > d+1$. Nevertheless, it has little influence on scene reconstruction since, for example, SfM is still correctly obtained.

This local approach presents an advantage over methods relying on $rank$ detection, where a small mistake is critical in segmentation, as well as for scene reconstruction, since it jeopardizes SfM.

4.2. Synthetic results

We performed 10000 trials on synthetic data, modelling articulated objects increasingly contaminated with outliers. The number of frames was set to $F = 10$ and the number of inliers uniformly chosen between 25 and 50 for each part of the object. The object has a randomly chosen link (joint or hinge) for each trial and the outliers were modeled by 1D trivial subspaces, drawn from the standard normal distribution.

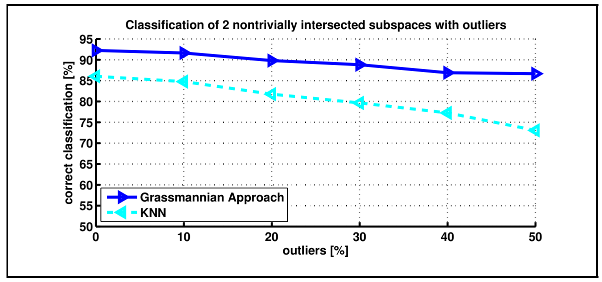

Figure 7. Average of correct classifications for 10000 trials on synthetic data.

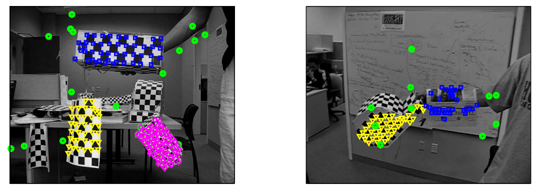

Figure 8. First frame from the articulated $(\mathbf{a})$ and the arm sequence $(\mathbf{b})$ (Hopkins155 database), with 10% of outliers superimposed (green $\circ$).

Figure 7 depicts the results. As expected, the Grassmannian approach increases the accuracy for kNN initialization, most noticeably when the outlier's contamination is high. Also, no outlier was classified as an inlier. Notice the decrease of 5.58% on the accuracy for an increase on the outliers' population corresponding to 50% of the total number of observations. In fact, since the LSE is a maximum consensus heuristic, the accuracy of the method does not depend so much on the outlier contamination as it does on the initialization. This explains the small decrease on performance with a high percentage of outliers and is why, on average, we cannot achieve 100% accuracy for 0% of outliers.

4.3. Real data results

We tested our approach on the articulated $(\mathbf{a})$ and arm $(\mathbf{b})$ sequences from the Hopkins155 database [9], adding 10% of random trajectories (Figure 8). The first sequence has three linked bodies, forming one articulated object, with 150 inlier trajectories over 31 frames. The arm sequence has 77 inlier trajectories over 30 frames. We set $min\_consensus = 5$ and look for $4D$ dimensional subspaces, *i.e.* $d_{max} = d_{min} = 4$, taking a conservative approach by setting $\xi = 10^{-5}$ (*c.f.* Section 4.1). Figure 9 shows the results.

In no experiment was ever an outlier misclassified as inlier. In sequence $(\mathbf{a})$ we found three subspaces, corresponding to the bodies of the articulated object and one inlier was classified as outlier, as a result of conservative threshold. The results for sequence $(\mathbf{b})$ show four $4D$ subspaces (a case of over-segmentation), two inliers were considered outliers and one inlier from the $\triangle$-group was wrongly clus-

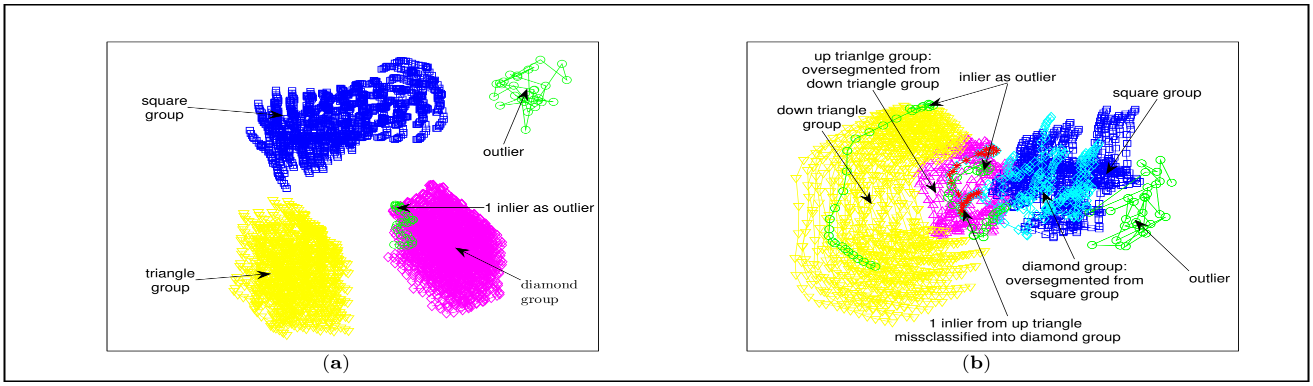

Figure 9. Segmentation results for the sequences in Figure 8.

tered in the $\Diamond$-group (red $*$-trajectory). Our algorithm classified 33 points in the ∇-group, 25 in the $\Box$-group, 9 in the $\triangle$-group and 8 in the $\Diamond$-group. Data points in both the $\triangle$-group and the $\Diamond$-group were left out of their reference clusters (∇-group and $\Box$-group, respectively) by the threshold. In this case, over-segmentation and misclassification would be solved without any threshold adjustment by simply imposing a higher consensus (*e.g.*, $min\_consensus = 20$).

5. Conclusion

We presented a methodology for segmenting arbitrary unions of linear subspaces. Our approach differs from the conventional methods by efficiently searching for the maximum consensus subspace inside the Grassmann manifold, providing a criterion for segmenting, without relying on $rank$ detection, voting or random search. The drawback is that global optimality depends on the initialization. However, by finding the maximum consensus subspace, outlier rejection becomes an inherent property of the method.

Experimental results on both synthetic and real data show a decrease of 5.58% on the accuracy for a 50.0% increase on the number of outliers and the capability to deal with challenging real sequences. Also, no outliers were ever classified as inliers. Ongoing work is being conducted towards exploring alternative initializations and the recursive structure of the algorithm for building automatic kinematic chains in the presence of outliers.

Appendix

Let $\mathcal{T}_{\mathcal{L}^\perp}\mathcal{G}_{n,n-d}$ denote the tangent space to the $\mathcal{G}_{n,n-d}$ at $\mathcal{L}^\perp$. Recall that the tangent space to a p-dimensional manifold is a p-dimensional vector space with origin at tangency point. A *geodesic* emanating from $\mathcal{L}^\perp \in \mathcal{G}_{n,n-d}$ along the tangent direction $\mathbf{X}$ is computed by

$$\gamma_{\mathbf{X}}(t) = span\{[\mathbf{QV}\ \mathbf{U}] \begin{bmatrix} \cos \mathbf{\Sigma}t \\ \sin \mathbf{\Sigma}t \end{bmatrix} \mathbf{V^T}\}, \quad (14)$$

where $\mathbf{Q}$ is a representative of $\mathcal{L}^\perp$, $\mathbf{X} = \mathbf{U\Sigma V^T}$ and $t \in (0, 1]$ is the stepsize. The *intrinsic gradient* at $\mathcal{L}^\perp$, is such that

$$\mathbf{G_{int}}(\mathcal{L}^\perp) = (\mathbf{I_n} - \mathbf{QQ^T})\mathbf{G_{ext}}(\mathbf{Q}), \quad (15)$$

where $\mathbf{G_{ext}}(\mathbf{Q}) \in \mathcal{R}^{n\times(n-d)}$ is the extrinsic (usual) gradient of the LSE map at point $\mathbf{Q} \in \mathcal{R}^{n\times(n-d)}$, in matrix form. The *intrinsic Newton's direction* is the tangent vector $\mathbf{X}$ computed as $\mathbf{X} = \mathbf{Q}_\perp\mathbf{A}$, with $\mathbf{Q^T}\mathbf{Q}_\perp = \mathbf{0}$ and $\mathbf{A} \in \mathcal{R}^{d\times(n-d)}$ verifying

$$\mathbf{H_{int}}vec(\mathbf{A}) = -vec(\mathbf{Q}_\perp^\mathbf{T}\mathbf{G_{int}}), \quad (16)$$

and $\mathbf{H_{int}} = \mathbf{I_{n-d}} \otimes \mathbf{Q}_\perp^\mathbf{T}\mathbf{H_{ext}}(\mathbf{Q})\mathbf{Q}_\perp - sym(\mathbf{Q^T G_{ext}}) \otimes \mathbf{I_d}$, where $vec(\cdot)$ stacks the argument's columns into a vector, $\otimes$ is the Kronecker product, $sym(\cdot)$ is the symmetric part of the argument and $\mathbf{H_{ext}}(\mathbf{Q}) \in \mathcal{R}^{n\times n}$ is the extrinsic hessian of the LSE map.

References

[1] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2^{nd} edition, 1999. 4

[2] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3):159–179, 1998. 1

[3] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. MAtrix. Anal. Appl.*, 20(2):303–353, 1998. 4, 5

[4] Z. Fan, J. Zhou, and Y. Wu. Multibody grouping by inference of multiple subspaces from high-dimensional data using oriented-frames. *in IEEE Trans. on PAMI*, 28(1):91–105, 2006. 1

[5] O. L. Mangasarian. Minimum-support solutions of polyhedral concave programs, 1997. Mathematical Programming Technical Report 97–05. 3

[6] M. Mesbahi and G. P. Papavassilopoulos. On the rank minimization problem over a positive semidefinite linear matrix inequality. *IEEE Trans. on Aut. Control*, 42(2):239–243, 1997. 3

[7] Y. Sugaya and K. Kanatani. Outliers removal for motion tracking by subspace separation. *IEICE Trans. on Inf. and Syst.*, E86D(6):1321–1340, 2003. 1

[8] P. Torr. Geometric motion segmentation and model selection. *Phil. Trans. Royal Society of London*, 356(1740):1321–1340, 1998. 1

[9] R. Tron and R. Vidal. A benckmark for the comparison of 3–d motion segmentation algorithms. *in Proc. CVPR'07*. 5

[10] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *IEEE Trans. on PAMI*, 27(12):1945–1959, 2005. 1

[11] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, non-rigid, degenerate and non-degenerate. *in Proc. ECCV'06*. 1, 5