Subspace Estimation Using Projection Based M-Estimators over Grassmann Manifolds

Raghav Subbarao and Peter Meer

Department of Electrical and Computer Engineering,
Rutgers University, Piscataway, NJ 08854, USA
{rsubbara, meer}@caip.rutgers.edu

Abstract. We propose a solution to the problem of robust subspace estimation using the projection based M-estimator. The new method handles more outliers than inliers, does not require a user defined scale of the noise affecting the inliers, handles noncentered data and nonorthogonal subspaces. Other robust methods like RANSAC, use an input for the scale, while methods for subspace segmentation, like GPCA, are not robust. Synthetic data and three real cases of multibody factorization show the superiority of our method, in spite of user independence.

1 Introduction

The estimation of subspaces is a problem which occurs frequently in computer vision, e.g., in the analysis of dynamic scenes [5, 8, 14]. Given data lying in a N dimensional space, linear regression estimates a $N-1$ dimensional hyperplane containing the inliers. If a regression algorithm is adapted to simultaneously estimate k *linearly independent* constraints which the inliers in the data satisfy, the intersection of the hyperplanes represented by these k constraints gives the required $N-k$ dimensional subspace.

We will generalize the robust projection based M-estimator (pbM) of [3, 13] to obtain a *user independent, robust, multiple subspace* estimation algorithm. As we discuss later, the parameter space is an algebraic structure known as the Grassmann manifold and we adapt the pbM algorithm to account for the geometry of this space [6].

If all the data points lie in the same subspace, then *Principal Component Analysis* (PCA) could be used to obtain the subspace. Standard PCA is not enough in practice because the data may contain multiple subspaces and/or outliers. Methods such as [1, 2] perform *robust PCA* to handle outliers. There are two problems with robust PCA algorithms which make them infeasible for multiple subspace estimation. Firstly, the methods of [1, 2] have breakdown points of 0.5, and secondly, the algorithms cannot handle structured outliers. These methods can only be used to estimate a single subspace and an example of this is shown in Section 4.

A number of multiple subspace estimation techniques have been developed in the vision community, e.g., subspace separation [5, 10] and generalized PCA

A. Leonardis, H. Bischof, and A. Prinz (Eds.): ECCV 2006, Part I, LNCS 3951, pp. 301–312, 2006.

(GPCA) [17, 16]. Much of the work done in this area was geared towards solving the problem of motion segmentation.

Most methods make simplifying assumptions about the data. Firstly, in [5, 10] it is assumed that the subspaces are orthogonal. Therefore, for degenerate motions where the subspaces share a common basis vector, the methods break down [20]. Secondly, the methods of [5, 10] require the data to be centered which is difficult to ensure in practice, especially in the presence of outliers. Finally, [5, 17] do not account for outliers. Outliers were partially accounted for in [16], but it is assumed that even in the presence of outliers the algorithm returns a rough estimate of the true subspaces *and* the scale of the noise corrupting the inliers is known. Both these assumptions are often not true in practice.

In this paper we propose a robust, pbM based, subspace estimation method. It does not suffer from the drawback of previous methods and can be used for multiple subspace estimation by iteratively estimating the 'dominant' subspace, treating all points *not* belonging to this subspace as outliers. After removing the points lying in the estimated subspace, the procedure can be repeated on the remaining points. We assume the dimension of the subspaces and the number of motions is known beforehand although the second assumption can be relaxed. Our method offers several advantages.

- No user input is required for the scale of noise affecting the inliers.
- Handles data sets with more outliers than inliers.
- Handles noncentered data and estimates the centroid of the inliers.
- Does not require orthogonal subspaces for the inliers.

The remainder of the paper is organized as follows. Section 2 gives an introduction to Grassmann manifolds and the conjugate gradient algorithm over Grassmann manifolds. In Section 3 we discuss robust subspace estimation with the pbM estimator. In Section 4 we validate our method on synthetic data and real data by comparing its performance with subspace separation [5, 10], GPCA [17, 16] and RANSAC [7].

2 Grassmann Manifolds

We discuss a few relevant concepts about Grassmann manifolds in this section. A more thorough introduction to Grassmann manifolds can be found in [6].

A manifold is a topological space that is locally similar (homeomorphic) to Euclidean space. The dimension of the Euclidean space to which the manifold is locally similar to, is also the dimension of the manifold. Every real manifold can be embedded in a higher dimensional Euclidean space which means that we can think of the manifold as a smooth surface lying in a higher dimensional Euclidean space, as illsutrated in Figure 1a.

We are concerned with a particular class of manifolds known as Grassmann manifolds. A point on the *Grassmann manifold*, $\mathbf{G}_{N,k}$, represents a k dimensional subspace of $\mathbb{R}^N$ and is numerically represented by an orthonormal basis as a $N \times k$ matrix, i.e., $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}_{k\times k}$. Since many different basis span the same subspace,

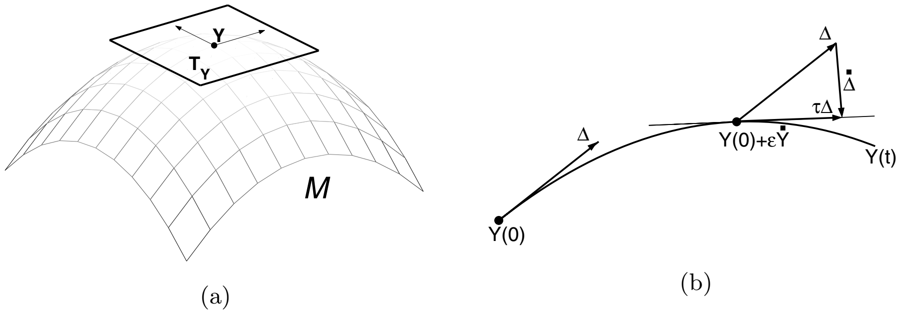


Fig. 1. Example of a manifold. (a) A two-dimensional manifold embedded in $\mathbb{R}^3$. The tangent space at the point $\mathbf{Y}$ is also shown. (b) Parallel transporting the vector Δ along the curve $\mathbf{Y}(t)$. The point moves along $\dot{\mathbf{Y}}$ and the component of $\dot{\Delta}$, which does not lie in the tangent space, is removed.

this representation of points on $\mathbf{G}_{N,k}$ is *not* unique [6]. $\mathbf{G}_{N,k}$ is a manifold of dimension $d = Nk - k(k+1)/2$ embedded in $\mathbb{R}^{Nk}$.

The *tangent space* $T_{\mathbf{Y}}$, at a point $\mathbf{Y}$, is the plane tangent to the surface of the manifold at that point. An example is shown in Figure 1a. For a d-dimensional manifold, the tangent space is a d-dimensional vector space. The tangent space is associated with an inner product g_c, such that for any two tangent vectors $\mathbf{\Delta}_1, \mathbf{\Delta}_2 \in T_{\mathbf{Y}}$ the inner product $g_c(\mathbf{\Delta}_1, \mathbf{\Delta}_2)$ lies in $\mathbb{R}$.

For a real function f defined on the manifold, the gradient at $\mathbf{Y}$ is defined to be that *unique* vector $\nabla f \in T_{\mathbf{Y}}$ which satisfies

$$tr(f_{\mathbf{Y}}^T \mathbf{\Delta}) = g_c(\nabla f, \mathbf{\Delta}) \tag{1}$$

where, $f_{\mathbf{Y}}$ is the Jacobian of f at $\mathbf{Y}$ and tr is the trace operator. For Grassmann manifolds the gradient vector is given by

$$\nabla f = f_{\mathbf{Y}} - \mathbf{Y}\mathbf{Y}^T f_{\mathbf{Y}}. \tag{2}$$

Since the tangent space of a manifold varies from point-to-point, if we move a tangent vector from one point to another point it generally does not lie on the tangent plane anymore. However, a tangent vector can be moved along paths on the manifold by taking infinitesimal steps along the curve $\mathbf{Y}(t)$, and at each step removing the component of the vector not in the tangent space. This process is known as *parallel transport.* Figure 1b shows a simple case of this idea.

A *geodesic* is defined to be the curve of shortest length between two point on the manifold. Parametric formulae can be derived for a geodesics on the Grassmann manifold, given the starting point and the tangent vector at that point [6].

Most function optimization techniques, e.g., Newton iterations and conjugate gradient, apply to functions defined over Euclidean spaces. Based on the theoretical concepts defined above, similar methods have been developed for Grassmann manifolds [6]. As we show in Section 3, the parameter space we consider is the direct product of a Grassmann manifold and a real space, $\mathbf{G}_{N,k} \times \mathbb{R}^k$. The rest of this section discusses conjugate gradient function minimization over this parameter space. The algorithm follows the same general structure as standard conjugate gradient but has some differences with regard to the movement of tangent vectors.

We now discuss a conjugate gradient algorithm for the minimization of a function $\mathbf{f}$ from the manifold $\mathbf{G}_{N,k} \times \mathbb{R}^k$ to $\mathbb{R}$. Conjugate gradient minimization requires the computation of $\mathbf{G}$ and $\mathbf{g}$, the gradients of f with respect to $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$. To obtain the gradients at a point $(\boldsymbol{\Theta}, \boldsymbol{\alpha})$, compute the Jacobians $\mathbf{J}_{\boldsymbol{\Theta}}$ and $\mathbf{J}_{\boldsymbol{\alpha}}$ of f with respect to $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$. The gradients are

$$\mathbf{G} = \mathbf{J}_{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\boldsymbol{\Theta}^T \mathbf{J}_{\boldsymbol{\Theta}} \qquad\qquad \mathbf{g} = \mathbf{J}_{\boldsymbol{\alpha}}. \tag{3}$$

Let $(\boldsymbol{\Theta}_0, \alpha_0) \in \mathbf{G}_{N,k} \times \mathbb{R}^k$ be the point at which the algorithm is initialized. Compute the gradients $\mathbf{G}_0$ and $\mathbf{g}_0$, at $(\boldsymbol{\Theta}_0, \boldsymbol{\alpha}_0)$ and the search directions are $\mathbf{H}_0 = -\mathbf{G}_0$ and $\mathbf{h}_0 = -\mathbf{g}_0$.

The following iterations are done till convergence. Iteration $j+1$ now proceeds by minimizing f along the geodesic defined by the search directions $\mathbf{H}_j$ on the Grassmann manifold and $\mathbf{h}_j$ in the Euclidean component of the parameter space. This is known as *line minimization.* The parametric form of the geodesic is

$$\boldsymbol{\Theta}_j(t) = \boldsymbol{\Theta}_j \mathbf{V}\mathbf{diag}(\cos \lambda t)\mathbf{V}^T + \mathbf{U}\mathbf{diag}(\sin \lambda t)\mathbf{V}^T \tag{4}$$

$$\boldsymbol{\alpha}_j(t) = \boldsymbol{\alpha}_j + t\mathbf{h}_j. \tag{5}$$

where, t is the parameter, $\boldsymbol{\Theta}_j$ is the estimate from iteration j and $\mathbf{U}\mathbf{diag}(\lambda)\mathbf{V}^T$ is the compact SVD of $\mathbf{H}_j$ consisting of the k largest singular values and corresponding singular vectors. The *sin* and *cos* act element-by-element.

Denoting the value of the parameter t where the minimum is achieved by t_{min}, set $\boldsymbol{\Theta}_{j+1} = \boldsymbol{\Theta}_j(t_{min})$ and $\boldsymbol{\alpha}_{j+1} = \boldsymbol{\alpha}_j(t_{min})$. The gradient vectors are parallel transported to this point by

$$\mathbf{H}_j^{\tau} = [-\boldsymbol{\Theta}_j \mathbf{V}\mathbf{diag}(\sin \lambda t_{min}) + \mathbf{U}\mathbf{diag}(\cos \lambda t_{min})]\, \mathbf{diag}(\lambda)\mathbf{V}^T \tag{6}$$

$$\mathbf{G}_j^{\tau} = \mathbf{G}_j - [\boldsymbol{\Theta}_j \mathbf{V}\mathbf{diag}(\sin \lambda t_{min}) + \mathbf{U}(\mathbf{I} - \mathbf{diag}(\cos \lambda t_{min}))]\, \mathbf{U}^T \mathbf{G}_j \tag{7}$$

where, τ is the parallel transportation operator. No explicit parallel transport is required for the Euclidean component of the parameter space since parallel transport for Euclidean spaces is trivially achieved by moving the whole vector as it is. The new gradients $\mathbf{G}_{j+1}$ and $\mathbf{g}_{j+1}$ are computed at $(\boldsymbol{\Theta}_{j+1}, \boldsymbol{\alpha}_{j+1})$. The new search directions are chosen orthogonal to *all* previous search directions as,

$$\mathbf{H}_{k+1} = -\mathbf{G}_{k+1} + \gamma_k \mathbf{H}_k^{\tau} \qquad\qquad \mathbf{h}_{k+1} = -\mathbf{g}_{k+1} + \gamma_k \mathbf{h}_k \tag{8}$$

$$\gamma_k = \frac{tr\Big((\mathbf{G}_{k+1} - \mathbf{G}_k^{\tau})^T \mathbf{G}_{k+1}\Big) + (\mathbf{g}_{k+1} - \mathbf{g}_k)^T \mathbf{g}_{k+1}}{tr\Big(\mathbf{G}_k^T \mathbf{G}_k\Big) + \mathbf{g}_k^T \mathbf{g}_k} \tag{9}$$

where, *tr* is the trace operator. The algorithm is summarized in Figure 2.

```
– Initialize at $(\boldsymbol{\Theta}_0, \boldsymbol{\alpha}_0) \in \mathbf{G}_{N,k} \times \mathbb{R}^k$,
    • Compute the gradients $\mathbf{G}_0$ and $\mathbf{g}_0$ at $(\boldsymbol{\Theta}_0, \boldsymbol{\alpha}_0)$ using (3).
    • Set $\mathbf{H}_0 = -\mathbf{G}_0$ and $\mathbf{h}_0 = -\mathbf{g}_0$.
– For $j = 0, 1, \ldots$
    • Minimize $f(\boldsymbol{\Theta}_j(t), \boldsymbol{\alpha}_j(t))$ over $t$ where $\boldsymbol{\Theta}_j(t)$ and $\boldsymbol{\alpha}_j(t)$ are as in (4) and (5).
    • Set $\boldsymbol{\Theta}_{j+1} = \boldsymbol{\Theta}_j(t_{min})$ and $\boldsymbol{\alpha}_{j+1} = \boldsymbol{\alpha}_j(t_{min})$.
    • Compute the gradients $\mathbf{G}_{j+1}$ and $\mathbf{g}_{j+1}$ at $(\boldsymbol{\Theta}_{j+1}, \boldsymbol{\alpha}_{j+1})$ according to (3).
    • Parallel transport the vectors $\mathbf{H}_j$ and $\mathbf{G}_j$ to $(\boldsymbol{\Theta}_{j+1}, \boldsymbol{\alpha}_{j+1})$ using (6) and (7).
    • Set the new search directions according to (8) and (9).
```

Fig. 2. Conjugate gradient algorithm for minimization of $f(\boldsymbol{\Theta}, \boldsymbol{\alpha})$ on $G_{N,k} \times \mathbb{R}^k$

3 Robust Subspace Estimation

Robust methods, such as RANSAC and its variations, handle data corrupted with outliers by making assumptions about the scale of the noise corrupting the inliers. The pbM estimator [3, 13] is independent of a user supplied scale parameter and exploit the intrinsic relation between the optimization criteria and the data space.

3.1 Projection Based M-Estimators

The subspace estimation problem can be stated as follows. Let $\mathbf{y}_{io}$ be the true value of the given data points $\mathbf{y}_i$. Given $\mathbf{y}_i$, $i = 1, \ldots, n$, the problem of subspace estimation is to estimate $\boldsymbol{\Theta} \in \mathbb{R}^{N \times k}$, $\boldsymbol{\alpha} \in \mathbb{R}^k$

$$\begin{aligned} & \boldsymbol{\Theta}^T \mathbf{y}_{io} - \boldsymbol{\alpha} = \mathbf{0}_k \\ \mathbf{y}_i = \mathbf{y}_{io} + \delta\mathbf{y}_i \qquad & \qquad \delta\mathbf{y}_i \sim GI(0, \sigma^2 \mathbf{I}_{N \times N}) \end{aligned} \tag{10}$$

where, σ the *unknown* scale of the noise. Handling non-identity covariances for heteroscedastic data, is a trivial extension of this problem e.g. [11]. The multiplicative ambiguity is resolved by requiring $\boldsymbol{\Theta}^T \boldsymbol{\Theta} = \mathbf{I}_{k \times k}$.

Given a set of k linearly independent constraints, they can be expressed by an equivalent set of *orthonormal* constraints. The $N \times k$ orthonormal matrix $\boldsymbol{\Theta}$ represents the k constraints satisfied by the inliers. The inliers have $N - k$ degrees of freedom and lie in a subspace of dimension $N - k$. Geometrically, $\boldsymbol{\Theta}$ is the basis of the k dimensional null space of the data and is a point on the Grassmann manifold $G_{N,k}$. Usually $\boldsymbol{\alpha}$ is taken to be zero since any subspace must contain the origin. However, for a robust formulation where the data is not centered, $\boldsymbol{\alpha}$ represents an estimate of the centroid of the inliers. Since we are trying to estimate both $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$, the complete search space for the parameters is $G_{N,k} \times \mathbb{R}^k$. The projection of $\boldsymbol{\alpha}$ onto the column space of $\boldsymbol{\Theta}$ is given by $\boldsymbol{\Theta\alpha}$ and this product should be independent of the basis used to represent the subspace.

The robust M-estimator formulation of the subspace estimation problem is

$$\left[\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\Theta}}\right] = \arg\min_{\boldsymbol{\alpha},\boldsymbol{\theta}} \frac{1}{n\left|\mathbf{S}_{\boldsymbol{\Theta}}\right|^{1/2}} \sum_{i=1}^{n} \rho\left(\mathbf{x}_i^T \mathbf{S}_{\boldsymbol{\Theta}}^{-1} \mathbf{x}_i\right) \tag{11}$$

where, $\mathbf{x}_i = \boldsymbol{\Theta}^T \mathbf{y}_i - \boldsymbol{\alpha}$, $\mathbf{S}_{\boldsymbol{\Theta}}$ is a scale matrix and $\left|\mathbf{S}_{\boldsymbol{\Theta}}\right|$ is its determinant. Note, that M-scores are usually not normalized by the determinant of the scale matrix. In our case, the scale matrix varies with the subspace $\boldsymbol{\Theta}$ and this normalization is required [13]. The function $\rho(u)$ considered here is a *loss function* in u, i.e., it is nondecreasing with $|u|$, has a unique minimum at $\rho(0) = 0$ and a maximum of one as $|u| \to 1$. The M-estimator problem can be rewritten in terms of the function $\kappa(u) = 1 - \rho(u)$ which is referred to as the *M-kernel function*

$$\left[\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\Theta}}\right] = \arg\max_{\boldsymbol{\alpha},\boldsymbol{\Theta}} \frac{1}{n\left|\mathbf{S}_{\boldsymbol{\Theta}}\right|^{1/2}} \sum_{i=1}^{n} \kappa\left(\mathbf{x}_i^T \mathbf{S}_{\boldsymbol{\Theta}}^{-1} \mathbf{x}_i\right) . \tag{12}$$

We use the redescending M-estimator with the biweight loss function [3].

Consider a set of points $\mathbf{x}_i \in \mathbb{R}^k$, $i = 1, \ldots, n$ which have been generated by some *unknown* probability distribution direction, $f(\mathbf{x})$. *Kernel density estimation*, also known as the Parzen window method in pattern recognition literature, returns an estimate of this unknown distribution as[1]

$$\hat{f}_{\boldsymbol{\Theta}}(\mathbf{x}) = \frac{1}{n\left|\mathbf{H}\right|^{1/2}} \sum_{i=1}^{n} k\left((\mathbf{x}_i - \mathbf{x})^T \mathbf{H}^{-1} (\mathbf{x}_i - \mathbf{x})\right) . \tag{13}$$

where, $\mathbf{H}$ is a bandwidth matrix, $k(u)$ is the *profile function* which decreases with increasing $|u|$.

The optimal choice for the bandwidth used is dependent on the true distribution. For one-dimensional kernel density estimation the following approximate bandwidth selection formula was derived in [18, Sec.3.2.2]

$$h = n^{-1/5} \operatorname*{med}_j \left| x_j - \operatorname*{med}_i x_i \right| \tag{14}$$

and we later discuss how we adapt this for data dependent bandwidth matrices.

There exist obvious similarities between (12) and (13). In (13), if we take the M-kernel function $\kappa(u)$ as the kernel $k(u)$, the projections $\boldsymbol{\Theta}^T \mathbf{y}_i$ as the data points $\mathbf{x}_i$, replace $\mathbf{x}$ with $\boldsymbol{\alpha}$ and the bandwidth matrix $\mathbf{H}$ with the scale matrix $\mathbf{S}_{\boldsymbol{\Theta}}$, we get (12). The M-estimator problem can be rewritten as

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \left[\max_{\mathbf{x}} \hat{f}_{\boldsymbol{\Theta}}(\mathbf{x})\right] \tag{15}$$

[1] For $\hat{f}(x)$ to be a true density function and satisfy $\int_{\mathbb{R}} \hat{f}(x)dx = 1$ we should use $c\kappa(x)$ where c is chosen such that $c\int_{\mathbb{R}} \kappa(x)dx = 1$. However, this global scaling does not affect any of the further analysis and is ignored.

where, $\hat{f}_{\boldsymbol{\Theta}}(x)$ refers to the estimate defined in (13). The formulation of (15) maximizes the value of the kernel density estimate at the mode. The inner maximization in (15) returns the intercept as the mode of $\hat{f}_{\boldsymbol{\Theta}}(x)$, i.e., $\boldsymbol{\alpha} = \max_x \hat{f}_{\boldsymbol{\Theta}}(\mathbf{x})$. The pbM algorithm is based on this similarity between kernel density estimation and M-estimators.

3.2 The pbM Algorithm

The first part of each pbM iteration consists of *probabilistic sampling.* An elemental subset which uniquely defines a k-dimensional subspace of $\mathbb{R}^N$ is chosen to get an estimate of $\boldsymbol{\Theta}$.

Given $\boldsymbol{\Theta}$, the data points are projected into $\mathbb{R}^k$ and mean shift [4] is used find the mode of the projections in $\mathbb{R}^k$. The bandwidth matrix is taken to be diagonal, with the values for each direction independently chosen by (14). This method depends on the basis used and a rotation of the basis gives a bandwidth matrix which depends on the rotation in a complex manner. The pbM estimator exhibits a weak dependence on the exact form of the bandwidth, and this method is sufficient. Of the modes returned, the mode with highest density is retained as the intercept $\boldsymbol{\alpha}$ and the density at $\boldsymbol{\alpha}$ is assigned as the score of $(\boldsymbol{\Theta}, \boldsymbol{\alpha})$.

This score is now maximized in a neighborhood of $\boldsymbol{\Theta}$. In spite of the non-differentiable nature of (15), derivative based methods can be used for this optimization by ignoring the dependence of $\boldsymbol{\alpha}$ and $\mathbf{S}_{\boldsymbol{\Theta}}$ on $\boldsymbol{\Theta}$. To ensure $\boldsymbol{\Theta}^T\boldsymbol{\Theta} = I_{k\times k}$ continues to hold, conjugate gradient is adapted to the Grassmann manifold [6]. We include $\boldsymbol{\alpha}$ in the search space and the complete parameter space is actually $G_{N,k} \times \mathbb{R}^k$. The algorithm is given in Figure 2. At the convergence of the minimization, the mode is refined again using mean shift initialized at the current estimate of $\hat{\boldsymbol{\alpha}}$.

The procedure is repeated for each elemental subset and the $(\boldsymbol{\Theta}, \boldsymbol{\alpha})$ with the highest score is taken as $(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}})$. The inlier-outlier dichotomy estimation is user independent. Denote the i-th column of $\hat{\boldsymbol{\Theta}}$ by $\hat{\boldsymbol{\theta}}_i$ and consider the one-dimensional kernel density estimate of the projections along $\hat{\boldsymbol{\theta}}_i$. The mode of this distribution is given by $\hat{\alpha}_i$, the i-th value of $\hat{\boldsymbol{\alpha}}$. The first *strong minima* of this density on either side of the mode are used to define the limits of the inliers. Points with projections lying in this range for all the k basis vectors are declared to be inliers. Multiple subspaces are estimated by repeatedly running the above algorithm and removing the inliers at each stage from the data set.

4 Experimental Results

We compare the performance of our algorithm against various other estimators: robust PCA [1, 2], subspace separation [10], GPCA [17, 16] and RANSAC [7]. Most previous methods either try to handle multiple subspaces with no outliers e.g., GPCA, or estimate only one subspace in the presence of outliers e.g., robust PCA. RANSAC is the only previous method which can be used for estimating multiple subspaces even in the presence of outliers, but requires a user defined noise level. The superiority of pbM to RANSAC has also been experimentally verified before [3].

4.1 Synthetic Data

The synthetic data consisted of 100 points lying along two randomly chosen intersecting lines in 3D with 40 points on one line, 30 points on the other and 30 outliers. Zero mean Gaussian noise of increasing variance was added to the data and 1000 trials were run for each noise level. In each trial we considered four different estimation techniques, robust PCA, GPCA, RANSAC and the pbM estimator. The line with 40 points was estimated. Since robust PCA and GPCA do not account for noncentered data, the inliers are centered. Both RANSAC and pbM use 500 elemental subsets for estimation. Since the true scale of the noise corrupting the inliers is known, RANSAC was tuned to the optimal scale estimate as suggested in [15]. No user defined scale estimate is required for pbM.

The error between the true subspace $\boldsymbol{\Theta}$ and estimated subspace $\hat{\boldsymbol{\Theta}}$ is the geodesic length along the Grassmann manifold given by

$$e_{\boldsymbol{\Theta}} = d_{gm}(\hat{\boldsymbol{\Theta}}, \boldsymbol{\Theta}) = \|\boldsymbol{\omega}\|_2 \tag{16}$$

where, $\boldsymbol{\omega}$ is the vector of angles between the basis of $\hat{\boldsymbol{\Theta}}$ and $\boldsymbol{\Theta}$. These angles can be found by taking the SVD of $\hat{\boldsymbol{\Theta}}^T \boldsymbol{\Theta} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$. The values along the diagonal of $\boldsymbol{\Sigma}$ are the cosines of the angles in $\boldsymbol{\omega}$. The elements of $\boldsymbol{\omega}$ can be found by taking the inverse cosine of each diagonal elemnt of $\boldsymbol{\Sigma}$.

	Mean				Standard Deviation			
σ	RPCA	GPCA	RANSAC	pbM	RPCA	GPCA	RANSAC	pbM
0.25	0.432	0.498	0.012	0.003	0.160	0.293	0.001	0.049
0.50	0.445	0.494	0.015	0.006	0.151	0.300	0.003	0.034
0.75	0.431	0.488	0.017	0.008	0.157	0.295	0.004	0.019
1.00	0.440	0.492	0.020	0.011	0.165	0.309	0.006	0.024
1.25	0.434	0.490	0.020	0.013	0.156	0.299	0.006	0.022
1.50	0.451	0.479	0.020	0.016	0.158	0.319	0.008	0.018
1.75	0.442	0.492	0.020	0.017	0.158	0.335	0.009	0.019
2.00	0.429	0.483	0.021	0.019	0.161	0.343	0.011	0.016

Fig. 3. For the synthetic data the line with 40 points is estimated. Robust PCA and GPCA break down due to the outliers. RANSAC performs almost as good as pbM but requires a user defined scale input which has been tuned to the optimal value.

The mean and standard deviation of the error $e_{\boldsymbol{\Theta}}$ are shown in Figure 3. Robust PCA finds the direction which maximizes the variance of the projections and always estimates a line lying in between the two lines on the same plane, leading to a large mean error and relatively moderate standard deviation. GPCA breaks down because of the outliers. Even when applied only to the inliers, GPCA deteriorates with increasing noise levels. RANSAC is the only algorithm which is comparable to pbM.

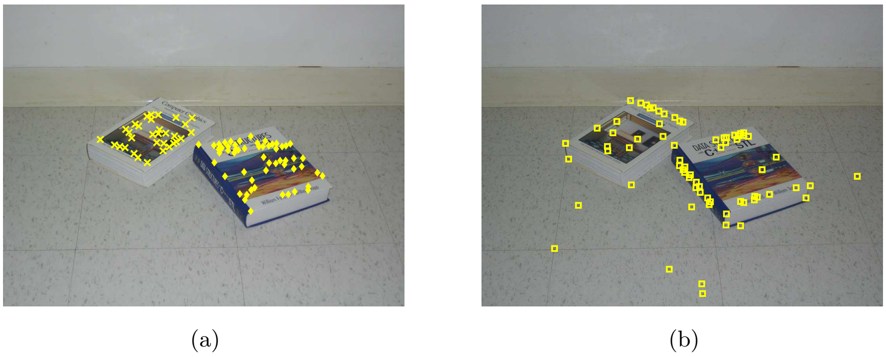
4.2 Real Data: Multibody Factorization

For real data we consider the factorization problem [14], since it is well studied and the degeneracies are well understood [19, 20]. Factorization is based on the fact that if n rigidly moving points are tracked over f *affine* images, then $2f$ image coordinates are obtained which can be used to define feature vectors in $\mathbb{R}^{2f}$. These vectors lie in a four-dimensional subspace of $\mathbb{R}^{2f}$ [14]. If the data is centered then the dimension of the subspace is only three.

We compare pbM to subspace separation [10], GPCA [17, 16] and RANSAC [7]. Our sequences have large displacements between frames leading to more outliers. They also consist of few frames leading to more degeneracies, for e.g., with three motions over four frames it is impossible to have independent subspaces since only 8 independent vectors can exist in the space, while at least 9 linearly independent vectors are required for each motion subspace to have an independent basis.

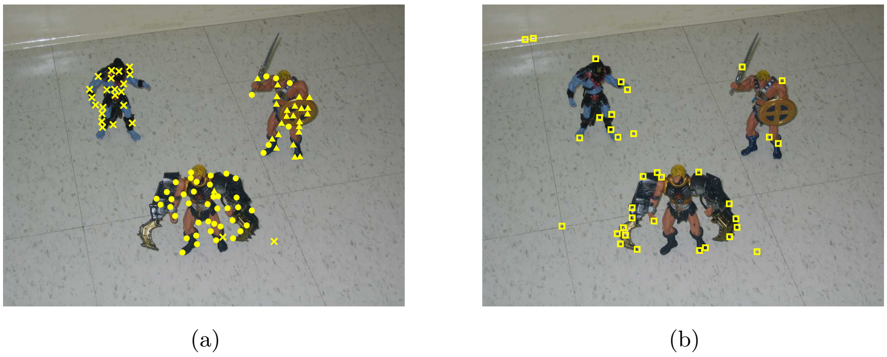
In subspace separation [10], a similarity measure is defined for pairs of feature vectors and these are arranged in a $n \times n$ symmetric *shape interaction matrix*. The clustering is done by making this matrix block diagonal. In our implementation we use the similarity measure of [20] which is more appropriate for dependent subspaces. For block diagonalization we use the algorithm of [12]. Since outliers do not lie in any subspace they may have high interactions with the inliers and the result is not robust.

An analytic solution to the multiple subspace estimation problem, GPCA, was presented in [17, 16]. This method is fast and can handle dependencies among the subspaces, but it is not robust. RANSAC [7] requires a user defined estimate for the scale of the noise corrupting the inliers. The ground truth was found


(a) (b)

	Inliers	GPCA	SS	RANSAC	pbM
Motion 1	52	94/35	157/52	61/52	56/51
Motion 2	30	64/14		35/29	32/29

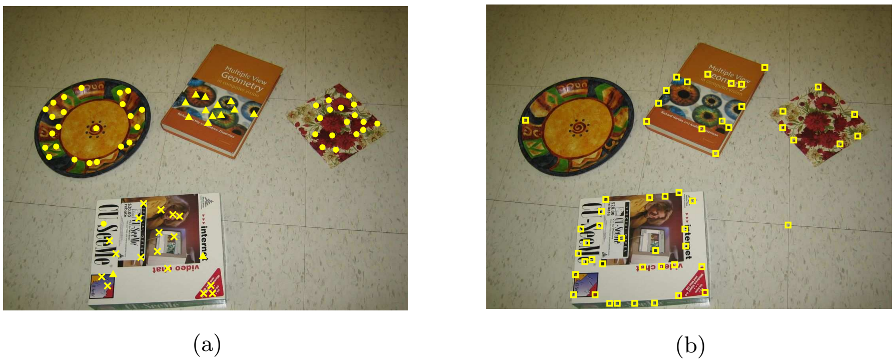
Fig. 4. *First Experiment.* (a) Segmented inliers returned by pbM for both motions, plotted on one of the frames. (b) Outliers returned by pbM. The table shows the results of the different estimators for the complete sequence.



(a) (b)

	Inliers	GPCA	SS	RANSAC	pbM
Motion 1	40	72/40	127/40	73/40	46/39
Motion 2	30	42/30			24/23
Motion 3	21	14/0		24/21	23/21

Fig. 5. *Second Experiment.* (a) Inliers returned by pbM for the three motions in the sequence. (b) Outliers returned by pbM. The table shows the results of the estimators.



(a) (b)

	Inliers	GPCA	SS	RANSAC	pbM
Motion 1	45	86/22	124/45	62/45	41/40
Motion 2	17	31/0			17/17
Motion 3	13	8/0		18/13	14/12

Fig. 6. *Third Experiment.* (a) Inliers returned by pbM for the three motions in the sequence. (b) Outliers returned by pbM. The table shows the results of the estimators.

through manual inspection. Given the ground truth, we compute the scale of the inlier noise $\hat{\sigma}$, and the RANSAC scale input is optimally set to $1.96\hat{\sigma}$ [15].

We used the point matching algorithm of [9] to track points. For the real data sets, both RANSAC and pbM used 1000 elemental subsets for estimating the first subspace, and 500 elemental subsets for estimating each further subspace.

An algorithm's performance is measured by its ability to cluster points correctly. This is measured by the ratio of the points declared as inliers to the number, among them, which are truly inliers. The closer this is to one the better.

We present our results on three progressively more complicated data sets. The *first sequence* consists of two moving bodies tracked over five frames. The motions subspaces are independent. Of the 158 features tracked, the two motions contained 52 and 30 points and 76 outliers. The results are shown in Figure 4. GPCA and subspace separation break down due to the outliers. GPCA randomly classifies the points into subspaces while subspace separation classifies all but one points into a single motion. Only on clean data, *with no outliers*, do GPCA and subspace separation give good results, but this never occurs in practice. The performance of RANSAC, when tuned to its optimal scale, is the same as pbM. A few of the mismatched points lie in the subspaces and are declared inliers.

The *second sequence* has three moving toys over four frames, with two of the motions having dependent subspaces. Of the 128 features tracked, the three motions contain 40, 30 and 21 inliers while 37 points were outliers. The results are shown in Figure 5. GPCA and subspace separation break down due to the outliers. In fact, subspace separation also breaks down on the clean data set due to degeneracies. RANSAC is unable to separate between the two degenerate motions since it cannot differentiate between outliers and noisy inliers, and classifies the inliers of both motions as a single motion. Only pbM is able to detect and segment all three motions.

The *third sequence* has three independent motions over four frames. The results are shown in Figure 6. The *plate* and *napkin* have the same motion, while the *book* and the *box* move independently. There are a large number of mismatches, and the motions subspaces are dependent. Among the 125 feature vectors the three motions contain 45, 17 and 13 inliers and there are 50 outliers. As before, GPCA and subspace separation break down. RANSAC cannot distinguish between two of the motions and combines both sets of inliers into one motion. Only pbM segments all motions correctly.

5 Conclusions

We proposed a robust subspace estimation algorithm based on the pbM estimator. The pbM algorithm required theoretical and computational modifications to estimate subspaces. For multiple structure estimation, currently, we recursively estimate the dominant subspace. We are working on methods which can simultaneously estimate the number of motions and segment them in a single step.

References

1. L. P. Ammann, "Robust singular value decompositions: A new approach to projection pursuit," *J. of Amer. Stat. Assoc.*, vol. 88, no. 422, pp. 505–514, 1993.
2. N. A. Campbell, "Robust procedures in multivariate analysis i: Robust covariance estimation," *Applied Statistics*, vol. 29, no. 3, pp. 231–237, 1980.

3. H. Chen and P. Meer, “Robust regression with projection based M-estimators,” in *9th Intl. Conf. on Computer Vision*, (Nice, France), Oct 2003, pp. 878–885.
4. D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 603–619, May 2002.
5. J. Costeira and T. Kanade, “A multi-body factorization method for motion analysis,” in *Proc. 5th Intl. Conf. on Computer Vision,* Cambridge, MA, 1995, pp. 1071–1076.
6. A. Edelman, T. A. Arias, and S. T. Smith, “The geometry of algorithms with orthogonality constraints,” *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.
7. M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Comm. Assoc. Comp. Mach*, vol. 24, no. 6, pp. 381–395, 1981.
8. C. W. Gear, “Multibody grouping from motion images,” *International J. of Computer Vision*, vol. 29, no. 2, pp. 133–150, 1998.
9. B. Georgescu and P. Meer, “Point matching under large image deformations and illumination changes,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, pp. 674–689, 2004.
10. K. Kanatani, “Motion segmentation by subspace separation and model selection,” in *8th Intl. Conf. on Computer Vision*, volume II, (Vancouver, Canada), July 2001, pp. 301–306.
11. B. Matei and P. Meer, “A general method for errors-in-variables problems in computer vision,” in *2000 IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, (Hilton Head Island, SC), June 2000, pp. 18–25.
12. J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 888–905, 2000.
13. R. Subbarao and P. Meer, “Heteroscedastic projection based M-estimators,” in *Workshop on Empirical Evaluation Methods in Computer Vision,* San Diego, CA in conjunction with *IEEE CVPR*, 2005.
14. C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: A factorization method,” *Intl. J. of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.
15. P. H. S. Torr and D. W. Murray, “The development and comparison of robust methods for estimating the fundamental matrix,” *Intl. J. of Computer Vision*, vol. 24, no. 3, pp. 271–300, 1997.
16. R. Vidal, Y. Ma, and J. Piazzi, “A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Washington, DC, vol. I, 2004, pp. 510–517.
17. R. Vidal, Y. Ma, and S. Sastry, “Generalized principal component analysis (GPCA),” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Madison, WI, vol. I, 2003, pp. 621–628.
18. M. P. Wand and M. C. Jones, *Kernel Smoothing.* Chapman & Hall, 1995.
19. Y.Sugaya and K. Kanatani, “Geometric structure of degeneracy for multi-body motion segmentation,” in D.Comaniciu et al., editor, *The 2nd Workshop on Statistical Methods in Video Processing*, no. 3247 in LNCS, pp. 1–2, Springer-Verlag, Berlin, Dec 2004.
20. L. Zelnik-Manor and M. Irani, “Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Madison, WI, number 2, 2003, pp. 297–293.