Generalized Projection Based M-Estimator: Theory and Applications

Sushil Mittal Saket Anand Peter Meer
ECE Department, Rutgers University, Piscataway, NJ - 08904
{smittal@caip, anands@eden, meer@jove}.rutgers.edu

Abstract

We introduce a robust estimator called generalized projection based M-estimator (gpbM) which does not require the user to specify any scale parameters. For multiple inlier structures, with different noise covariances, the estimator iteratively determines one inlier structure at a time. Unlike pbM, where the scale of the inlier noise is estimated simultaneously with the model parameters, gpbM has three distinct stages – scale estimation, robust model estimation and inlier/outlier dichotomy. We evaluate our performance on challenging synthetic data, face image clustering upto ten different faces from Yale Face Database B and multi-body projective motion segmentation problem on Hopkins155 dataset. Results of state-of-the-art methods are presented for comparison.

1. Introduction

RANdom SAmple Consensus (RANSAC) is the most widely used robust algorithm for computer vision applications and it depends on the user for specifying the scale of the inlier noise [9]. There are applications where it is hard for the user to provide the scale. For example, in video sequences, the scale of the inlier noise could change from frame to frame based on how fast the camera is moving. The various enhancements of RANSAC, like MLESAC, LO-RANSAC, PROSAC and QDEGSAC etc. (see [16]), propose changes to either the cost function, the sampling method, or detecting the degeneracies in data. However, none of these address the problem of manual scale selection.

Estimating the scale of the inlier noise is an important problem for any robust regression algorithm. The robust K^{th} Ordered Scale Estimator (KOSE) and Adaptive Least K^{th} order Squares (ALKS) [14] are generalization of the MAD (Median Absolute Deviation) based method and were among the first ones to address the problem of automatic scale estimation. Similarly, an algorithm to compute both the model and scale of the noise simultaneously using Weighted Median Absolute Deviation (WMAD) method was proposed in [8]. All previous versions of the projection based M-estimator (pbM) [4, 18, 19] also used a variant of the MAD scale estimate. Due to their dependence on MAD based scale estimation, all these methods are bound to fail when inliers comprise less than half the data points or contain noise from an asymmetric distribution. This is often the case when several inlier structures are present.

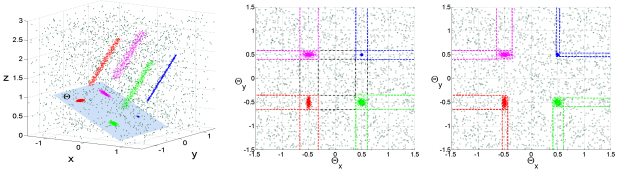


Figure 1. *Left*: Data containing multiple inlier structures and outliers. *Center*: Scale of noise estimated independently in each dimension of the two-dimensional null space, $\mathbf{\Theta}$ (*incorrect*). *Right*: Scale estimated simultaneously in both the dimensions (*correct*).

The Modified Selective Statistical Estimator (MSSE) [2] is a generalization of the Least Median of Squares method and tries to estimate the fraction of data points that belong to an inlier structure. However, it requires the user to specify an initial estimate of the minimum acceptable population of every inlier structure. The Two-Step Scale Estimator (TSSE) [26] uses mean shift to first find an inlier-outlier dichotomy and then estimate the scale, but their method assumes symmetric inlier distribution around the mode.

The main disadvantage of *all* these methods is that they estimate the scale of noise independently for each dimension of the null space. This may lead to gross inaccuracies in the scale estimate especially when data contains multiple inlier structures. Fig. 1 illustrates this problem using a multiple line fitting example in 3D. The four inlier structures lie along four parallel lines each having a different scale of noise along the two axes of the null space, $\mathbf{\Theta}$. It is impossible to estimate the scale correctly if the estimation is done along each axis of $\mathbf{\Theta}$ independently.

The projection based M-estimator (pbM) is described in [19] for estimating $m-k$ dimensional subspaces in $\mathbb{R}^m$. This method maximizes the M-score over randomly chosen subspace hypotheses. Let $\mathbf{\Theta} \in \mathbb{R}^{m\times k}$ represent the k-dimensional null space of the subspace hypothesis. The computation of the M-score depends on the $k \times k$ diagonal scale matrix $\mathbf{S}_{\mathbf{\Theta}}$ whose diagonal entries are computed for $p = 1, \ldots, k$ using

$$\mathbf{S}_{\mathbf{\Theta}}(p,p) = n^{-1/5} \operatorname*{med}_j \left| z_j^p - \operatorname*{med}_i z_i^p \right|, \quad i,j = 1,\ldots,n \quad (1)$$

where $z_i^p = \theta_p^\top \mathbf{x}_i$ is the projection of the i^{th} data point, $\mathbf{x}_i$ on to the p^{th} column of $\boldsymbol{\Theta}$. Since $\mathbf{S}_{\boldsymbol{\Theta}}$ depends on a particular $\boldsymbol{\Theta}$, it *does not* correspond to the actual scale of the inlier noise. The independence of the M-score over the $\boldsymbol{\Theta}$-dependent scale is only *partially* achieved by normalizing each M-score with the determinant of $\mathbf{S}_{\boldsymbol{\Theta}}$ [19]. Often, mean shift cannot converge to the correct mode using an incorrect scale estimate especially when the data contains asymmetric noise and multiple inlier structures. Using a $\boldsymbol{\Theta}$-dependent scale reduces the discrimination between correct and incorrect hypotheses.

We address these issues and present the generalized pbM (gpbM) algorithm for estimating multiple inlier structures from the data in the presence of outliers. An estimate of the scale and fraction of points belonging to an inlier structure is computed automatically in the beginning. The model estimation is then performed very efficiently using just the inliers returned by the scale estimate. Being completely user independent this method has obvious advantages over RANSAC-like algorithms and pbM [19].

- We propose an automatic method for estimating the scale of inlier noise in k dimensions simultaneously.
- We develop the most general form of pbM which can handle heteroscedastic data for single or multiple constraints in a unified framework.
- We introduce a new, theoretically justified method for inlier/outlier dichotomy.

In Section 2, we formulate the robust subspace estimation problem in k dimensions. In Section 3, we describe in detail the generalized projection based M-estimator. In Section 4 we present experimental results. We evaluate our algorithm on challenging synthetic data, face image clustering for all ten subjects from Yale Face Database B and multi-body projective motion segmentation problem on *Hopkins155* dataset as well as on a real-world example that contains unstructured outliers also.

2. Robust Subspace Estimation

In computer vision there is usually a non-linear relationship between the variables $\mathbf{y}$ and the carriers $\mathbf{x}$. The estimation problem is heteroscedastic, i.e., each carrier vector has a different covariance matrix, and in general can even have different mean. Let $\mathbf{x}_{io}$, $i = 1, \ldots, n_1$, be the true values of the inlier carrier points $\mathbf{x}_i \in \mathbb{R}^m$. Given a set of k linearly independent constraints, they can be expressed by an equivalent set of orthonormal constraints. The $m \times k \quad (k < m)$ orthonormal matrix $\boldsymbol{\Theta}$ represents the k constraints satisfied by the inliers. The inliers have $m - k$ degrees of freedom and thus lie in a subspace of dimension $m - k$. Geometrically, $\boldsymbol{\Theta}$ is the basis of the k-dimensional *null space* of the data.

Given n $(> n_1)$ data points $\mathbf{x}_i$, $i = 1, \ldots, n$, the problem of robust linear subspace estimation is to estimate the parameter matrix $\boldsymbol{\Theta} \in \mathbb{R}^{m \times k}$ and the intercept $\boldsymbol{\alpha} \in \mathbb{R}^k$ from the system of equations

$$\boldsymbol{\Theta}^\top \mathbf{x}_{io} - \boldsymbol{\alpha} = \mathbf{0}_k \; . \tag{2}$$

The multiplicative ambiguity is resolved by requiring $\boldsymbol{\Theta}^\top \boldsymbol{\Theta} = \mathbf{I}_{k \times k}$. For example, in fundamental matrix estimation, $\boldsymbol{\theta} \in \mathbb{R}^8$. Each data point is a vector of variables $\mathbf{y} = [x_1 \;\; y_1 \;\; x_2 \;\; y_2]^\top$, and lies in $\mathbb{R}^4$. Here, $(x_i, y_i), i = 1, 2$ are the coordinates of the corresponding points in the two images. The carrier vector used for linear regression is $\mathbf{x} = [x_1 \;\; y_1 \;\; x_2 \;\; y_2 \;\; x_1x_2 \;\; x_1y_2 \;\; y_1x_2 \;\; y_1y_2]^\top$ which lies in $\mathbb{R}^8$. Assuming the variables $\mathbf{y}$ have covariance $\sigma^2\mathbf{I}_{4\times4}$, the first order approximation of the covariance matrix of $\mathbf{x}$ is computed from the Jacobian using error propagation [15]

$$\mathbf{J}_{\mathbf{x}|\mathbf{y}} = \begin{bmatrix} 1 & 0 & 0 & 0 & x_2 & y_2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & x_2 & y_2 \\ 0 & 0 & 1 & 0 & x_1 & 0 & y_1 & 0 \\ 0 & 0 & 0 & 1 & 0 & x_1 & 0 & y_1 \end{bmatrix} = [\, \mathbf{I}_{4\times4} \;\; \mathbf{J}(\mathbf{y}) \,] \tag{3}$$

$$\mathbf{C}_{\mathbf{x}} = \sigma^2 \mathbf{J}_{\mathbf{x}|\mathbf{y}}^\top \, \mathbf{I}_{4\times4} \, \mathbf{J}_{\mathbf{x}|\mathbf{y}} = \sigma^2 \begin{bmatrix} \mathbf{I}_{4\times4} & \mathbf{J}(\mathbf{y}) \\ \mathbf{J}(\mathbf{y})^\top & \mathbf{J}(\mathbf{y})^\top \mathbf{J}(\mathbf{y}) \end{bmatrix} . \tag{4}$$

The covariance matrices, $\mathbf{C}_{\mathbf{x}}$ are used to estimate the point dependent scale of the noise in the regression data.

The points $\mathbf{x}_i, \;\; i = n_1 + 1, \ldots, n$ are outliers and no assumptions are made about their distribution. Since there could be several inlier structures, relative to one inlier structure the outliers can either belong to another inlier structure (structured outliers) or can be completely unstructured (gross outliers). No prior knowledge about the number of inlier structures is assumed. We consider here that the value of k is known and the data in not degenerate.

3. Generalized pbM Algorithm

The gbpM algorithm uses elemental subsets from which the estimates of $\boldsymbol{\Theta}$ are generated. The elemental subset based hypothesis generation was a well established method in statistics even before RANSAC. While RANSAC uses the elemental subsets to estimate both $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$, gpbM uses them to estimate only $\boldsymbol{\Theta}$.

We define the robust heteroscedastic objective function as

$$\left[\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{\alpha}}\right] = \underset{\boldsymbol{\Theta}, \boldsymbol{\alpha}}{\arg\max} \, \frac{1}{n} \sum_{i=1}^{n} \frac{K\left(\left((\boldsymbol{\Theta}^\top \mathbf{x}_i - \boldsymbol{\alpha})^\top \mathbf{B}_i^{-1} (\boldsymbol{\Theta}^\top \mathbf{x}_i - \boldsymbol{\alpha}) \right)^{\frac{1}{2}} \right)}{\sqrt{\det \mathbf{B}_i}} \tag{5}$$

where, $\boldsymbol{\Theta}^\top \mathbf{x}_i - \boldsymbol{\alpha}$ measures the deviation of the data from the required constraint. The kernel function $K(u)$ is related to the M-estimator loss function $\rho(u)$ by $K(u) = 1 - \rho(u)$, where $\rho(u)$ is a redescending M-estimator and is non-negative, symmetric and non-decreasing with $|u|$. It

has a unique minimum of $\rho(0) = 0$ and a maximum of one for $|u| > 1$.

The variables $\mathbf{y}_i$ are assumed here to be homoscedastic (not necessary in general) and their covariance matrices, $\mathbf{C}_{\mathbf{y}_i}$ are used to compute the covariance matrices $\mathbf{C}_i$ of the carriers $\mathbf{x}_i$ by error propagation [15]. (We will use $\mathbf{C}_i$ instead of $\mathbf{C}_{\mathbf{x}_i}$ for convenience.) The $k \times k$ covariance matrices of the projections $\mathbf{z}_i = \boldsymbol{\Theta}^\top \mathbf{x}_i$ are given by $\mathbf{H}_i = \boldsymbol{\Theta}^\top \mathbf{C}_i \boldsymbol{\Theta}$. Note that each $m \times k$ matrix $\boldsymbol{\Theta}$ results in different $k \times k$ covariance matrices, $\mathbf{H}_i$.

The scale matrix $\mathbf{S}$ is a $k \times k$ diagonal matrix, where the diagonal entries correspond to the value of scale in each dimension of the null space. As opposed to pbM [19], our scale matrix is $\boldsymbol{\Theta}$-independent. Finally, the $k \times k$ bandwidth matrices $\mathbf{B}_i$ are given by $\mathbf{B}_i = \mathbf{S}^\top \mathbf{H}_i \mathbf{S}$. Please note that our formulation of the objective function in (5) is different from the general homoscedastic M-estimator formulation by an additional factor of $[\det \mathbf{B}_i]^{-\frac{1}{2}}$. By doing this we make sure that deviations of points with larger covariances have smaller weights than points with smaller covariances.

To detect and estimate an inlier structure, we solve the optimization problem (5) in three steps. In *step one*, the k-dimensional scale of the inlier noise is estimated. In *step two*, we estimate the model parameter pair $[\boldsymbol{\Theta}, \boldsymbol{\alpha}]$. While $\boldsymbol{\Theta}$ is estimated from an elemental subset, the estimate of $\boldsymbol{\alpha}$ is computed as the location of the closest mode of the kernel density function over the projections $\mathbf{z}_i$ by using mean shift in $\mathbb{R}^k$. In *step three*, we compute the inlier/outlier dichotomy using the scale estimate from step one and model parameters from step two. The inliers thus obtained are then removed for the data and the three-step process is repeated to estimate another inlier structure. The algorithm stops once the value of the kernel density at the detected mode normalized by the determinant of scale matrix goes below a small threshold. See Section 3.2 for details.

3.1. Step One: Scale Estimation

The fundamental difference between inlier and outlier points is that the inliers are always tightly packed around the regression surface while the outliers are not. We first find the approximate fraction of data points that belong to an inlier structure by capturing the difference in density of the inliers and outliers.

We generate M elemental subset-based model hypotheses, $\boldsymbol{\Theta}_j, j = 1, \ldots, M$. The value of M is specific to a particular problem and will be given in Section 4. For each $\boldsymbol{\Theta}_j$ the k-dimensional projections, $\mathbf{z}_i = {\boldsymbol{\Theta}_j}^\top \mathbf{x}_i, i = 1, \ldots, n$ are computed. Let $\mathbf{T}_{\boldsymbol{\Theta}_j}$ be the k-dimensional null space in $\mathbb{R}^m$ associated with $\boldsymbol{\Theta}_j$. We vary the value of the fraction η, uniformly between $(0, 1]$ in Q steps. For $q = 1, \ldots, Q$, let η_q be the q^{th} fraction containing n_q points. Therefore, $\eta_q = n_q/n = q/Q$. With a slight abuse of notation, let

$$vol_j^q(\mathbf{z}_i) = \sqrt{\sum_{l=1}^{n_q} \|\mathbf{z}_i - \mathbf{z}_l\|^2} \tag{6}$$

be the volume around $\mathbf{z}_i$ containing the fraction η_q of points, where $\mathbf{z}_l, l = 1, \ldots, n_q$ are the nearest neighbors of $\mathbf{z}_i$ in $\mathbf{T}_{\boldsymbol{\Theta}_j}$. For a given $\boldsymbol{\Theta}_j$ and q, our goal is to find $\mathbf{z}^q_{min} = \mathbf{z}_{\hat{i}}$ such that

$$\hat{i} = \arg\min_i \left(vol_j^q(\mathbf{z}_i) \right). \tag{7}$$

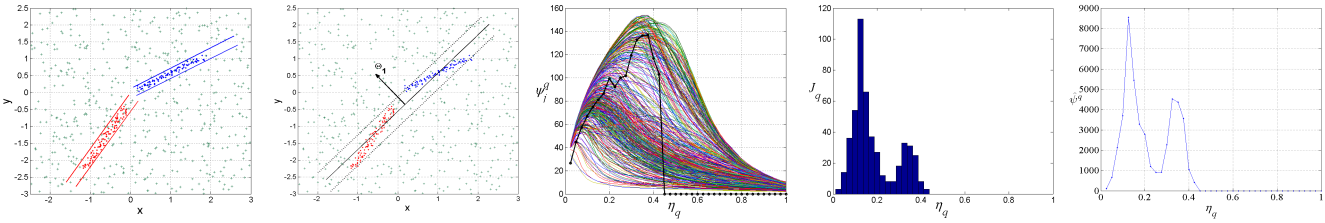
The simplest way would be to exhaustively search $\mathbf{T}_{\boldsymbol{\Theta}_j}$ for $\mathbf{z}^q_{min}$ using the nearest neighbor method [1], but unfortunately it often becomes a computational bottleneck when n and M are large. Instead, by doing linear search, we first find smallest volume regions in one dimension along the individual dimensions of $\mathbf{T}_{\boldsymbol{\Theta}_j}$. The centers of each of these k regions are the *candidate* points for $\mathbf{z}^q_{min}$. Therefore, instead of analysing all n points using nearest neighbor technique, we analyse only $k, (k \ll n)$ candidate points. Doing the k-dimensional search only for the k candidates does not guarantee finding $\mathbf{z}^q_{min}$, but our experiments showed insignificant differences in volume estimates compared to the exhaustive search. This approximation speeds up the search significantly and can also be implemented in parallel.

The density for the fraction η_q for a given $\boldsymbol{\Theta}_j$ is computed as $\psi_j^q = n_q/(vol_j^q(\mathbf{z}^q_{min}) + \epsilon)$. Since $vol_j^q(\mathbf{z}^q_{min})$ could be very close to zero for small fractions, a small constant ϵ is added to suppress extremely high values of densities. Computing the density for all M hypotheses and all Q fractions, we get an $M \times Q$ matrix $\boldsymbol{\Psi}$, with $\boldsymbol{\Psi}(j, q) = \psi_j^q$. For every q, let the number of rows of $\boldsymbol{\Psi}$ that have the maximum density in the q^{th} column be J_q and corresponding set of maximum density values be ψ^q_{max}. It can be verified that the sum $\sum_{q=1}^{Q} J_q = M$. The sum of peak density values for every q is then computed as

$$\hat{\psi}^q = \sum_{\psi \in \psi^q_{max}} \psi. \tag{8}$$

The summation of peak densities over all hypotheses makes the estimation more robust than any one individual peak density value. For example, in case of data containing multiple inlier structures and outliers, for some particular choices of $\boldsymbol{\Theta}_j$, the density computed for a combination of points from two or more inlier structures could be more than the densities for each individual inlier structure.

Fig. 2 illustrates the problem. The two inlier structures lie along two different lines in 2D each containing 100 points and 500 random outliers are also added. In Fig 2b, for a particular hypothesis, $\boldsymbol{\Theta}_1$, the value of ψ_j^q peaks at $\eta_q = 0.3$. This is more than the actual fraction of each inlier structure which is $\eta_q = 0.143$. Since there are very few orientations that for which ψ_j^q peaks at $\eta_q = 0.3$, the probability of selecting hypotheses like $\boldsymbol{\Theta}_1$ is much less than the



(a) (b) (c) (d) (e)

Figure 2. Inlier fraction estimation. (a) Original data with two inlier structures with the scale boundaries marked. (b) Boundary of the scale for a specific hypothesis $\boldsymbol{\Theta}_1$. (c) Density plots for all 400 randomly generated hypotheses. The solid black line shows the *average* peak density value, $\hat{\psi}^q/J_q$ for every fraction η_q. There are no peaks for $\eta_q > 0.425$. (d) Histogram of peaks. (e) Sum of peak density values, $\hat{\psi}^q$. The location of highest peak corresponds to the estimated fraction value, $\eta_{\hat{q}} = 0.125$.

probability of selecting a hypothesis where ψ_j^q peaks around $\eta_q = 0.143$. See Fig. 2d. This observation is true even when the two lines have different number of inlier points. Our conservative estimate of the fraction is computed using

$$\hat{q} = \arg\max_q \ \hat{\psi}^q. \tag{9}$$

See Fig. 2c–e. The data points are projected to the $\boldsymbol{\Theta}$ that gives the highest peak at the estimated fraction, $\eta_{\hat{q}}$. The dimensions of the smallest rectangular region in $\mathbf{T}_{\boldsymbol{\Theta}}$ enclosing $n_{\hat{q}}$ points divided by two gives the estimate of the scale in k dimensions which forms the diagonal of $\mathbf{S}$. The corresponding points are a conservative estimate of the inliers.

3.2. Step Two: Model Estimation Using Mean Shift

The set of inliers obtained in step one is used together with the estimated scale matrix to perform model estimation. Although this set may still contain a few outliers, the inlier-outlier ratio is much higher than that in the original set of data points. The model estimation is then performed by restricting the selection of elemental subsets over this inlier set only. This makes our model estimation step efficient as compared to the pbM and RANSAC algorithms.

For a given $\boldsymbol{\Theta}$, we first compute the k-dimensional projections $\mathbf{z}_i = \boldsymbol{\Theta}^\top \mathbf{x}_i, i = 1, \ldots, n$. The original non-linear robust estimation problem is reformulated into a simpler problem of estimating the kernel density in k dimensions by defining the profile of the kernel $K(u)$ as $\kappa(u^2) = K(u)$. Let the one-dimensional adaptive kernel density function based on the k-dimensional projections $\mathbf{z}_i$ is

$$f_{\boldsymbol{\Theta}}(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\kappa\left(\Delta\mathbf{z}_i^\top \mathbf{B}_i^{-1} \Delta\mathbf{z}_i\right)}{\sqrt{\det \mathbf{B}_i}}. \tag{10}$$

where $\Delta\mathbf{z}_i = \mathbf{z} - \mathbf{z}_i$. Taking the derivative of (10) we observe that the stationary points should satisfy

$$\nabla f_{\boldsymbol{\Theta}}(\mathbf{z}) = \frac{2}{n}\sum_{i=1}^{n} \frac{\mathbf{B}_i^{-1}\Delta\mathbf{z}_i\, g\left(\Delta\mathbf{z}_i^\top \mathbf{B}_i^{-1} \Delta\mathbf{z}_i\right)}{\sqrt{\det \mathbf{B}_i}} = 0 \tag{11}$$

where $g(u^2) = -\kappa'(u^2)$. The mean shift vector can be written as

$$\delta\mathbf{z} = \left[\sum_{i=1}^{n} \frac{\mathbf{B}_i^{-1} g\left(\ldots\right)}{\sqrt{\det \mathbf{B}_i}}\right]^{-1} \left[\sum_{i=1}^{n} \frac{\mathbf{B}_i^{-1}\mathbf{z}_i g\left(\ldots\right)}{\sqrt{\det \mathbf{B}_i}}\right] - \mathbf{z}. \tag{12}$$

Note that the bandwidth matrix $\mathbf{B}_i$ is different for each point, making the problem heteroscedastic. The iteration

$$\mathbf{z}^{(j+1)} = \delta\mathbf{z}^{(j)} + \mathbf{z}^{(j)} \tag{13}$$

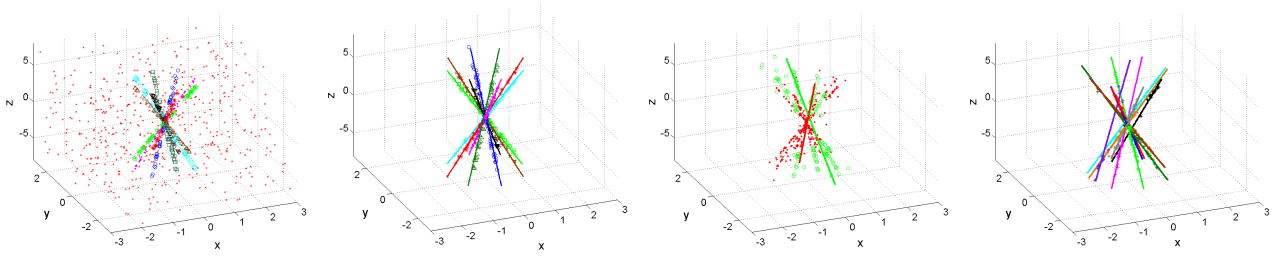
is a gradient ascent technique converging to the *closest* mode, $\boldsymbol{\alpha}$, a stationary point of the kernel density function.

This step is repeated for N randomly generated hypotheses of $\boldsymbol{\Theta}$ generated from the set of inliers returned by step one. The value of N is specific to the problem and will be given in Section 4. The estimated intercept $\widehat{\boldsymbol{\alpha}}$ corresponds to the location of the highest mode found, while the corresponding matrix $\widehat{\boldsymbol{\Theta}}$ is the estimate of $\boldsymbol{\Theta}$.

Stopping Criterion: To decide whether the estimated model belongs to an actual inlier structure, we compute a measure of the strength of the current inlier structure as $\xi = f_{\widehat{\boldsymbol{\Theta}}}(\widehat{\boldsymbol{\alpha}}) / \|\mathbf{S}\|^2$. The algorithm stops if the strength drops by a factor of 20 compared to the maximum of the strengths of previously computed inlier structures, indicating that the remaining points comprise only unstructured outliers.

3.3. Step Three: Inlier/Outlier Dichotomy

Given the model estimate $[\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{\alpha}}]$, let $\hat{\mathbf{z}}_i = \widehat{\boldsymbol{\Theta}}^\top \mathbf{x}_i$. Starting mean-shift iterations from every point $\hat{\mathbf{z}}_i,\ i = 1, \ldots, n$, the points for which the procedure converges at $\widehat{\boldsymbol{\alpha}}$ (with a small tolerance) are considered as inliers. The same bandwidth matrices $\mathbf{B}_i$ are used for the mean shift kernel. This method of dichotomizing data points into inliers and outliers is coherent with the maximum likelihood rule according to which points with residuals outside the *basin of attraction* of the mode are more likely to be outliers. However, points lying close to the boundary of the basin of attraction should be carefully dichotomized. Even a small error in the estimation of model parameters could lead to misclassifications around the boundary. One way of solving the problem is to use additional information that can



Original data gpbM Kernel fitting J-Linkage

Figure 3. Conic lines example. There are eight lines with 50 points per line and 500 random unstructured outliers. The inlier noise scale is 0.02. Only gpbM is almost always able to recover all eight lines.

be reliably extracted after model estimation. For example, in multiple motion segmentation, fundamental matrices for each motion can be robustly estimated and used to classify the points lying close to the boundary. This will be discussed in Section 4.3.

4. Experimental Results

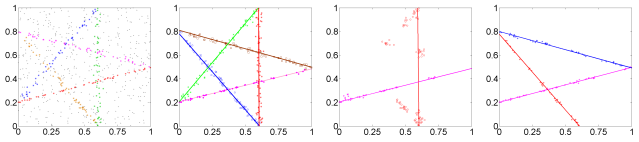
We present three groups of experiments. First we show the performance of our algorithm on two synthetic line fitting examples. Then we present two real-world applications: face image clustering and projective motion segmentation. While the first two problems are homoscedastic due to linear relationship between the carriers and variables, the problem of motion segmentation is heteroscedastic. The kernel density is estimated using the Epanechnikov kernel in the first two problems and Gaussian kernel in the third. The value of $Q = 40$ was used in all the experiments.

4.1. Synthetic Examples

Conic Lines. A line in 3D should satisfy two linear constraints. We generated 50 inliers each along eight different lines in 3D for z coordinate between $(-6, 6)$. The lines lie on surface of a double cone with its vertex at $(0, 0, 0)$ and axis aligned with z-axis. The angle between consecutive pairs of lines is about 7.3^o. The three coordinates of the inliers were independently corrupted with zero mean Gaussian noise with standard deviation of 0.02. In addition 500 outliers were added uniformly in x, y and z between $(-3, -3, -8)$, $(3, 3, 8)$. This is a very challenging problem because each inlier structure comprises only a fraction 0.056 of the total number of points. Neither the number of inlier structures nor the scale of noise in each structure is known when applying the gpbM algorithm. For each inlier structure, in the scale estimation $M = 1000$ and in the model estimation $N = 200$ were used. The results were compared with RANSAC [9], the kernel fitting (KF) method [6], and the J-Linkage method [21]. Both RANSAC and J-Linkage had to be provided the value of *true scale* of inlier noise. Additionally, RANSAC was also given the actual number of inlier structures and J-Linkage the minimum number of inliers present in an inlier structure.

Fig. 3 shows a comparison of the results obtained. RANSAC was able to find all the eight structures only if right parameters were given and is not shown in the figure. Over 100 runs, KF and J-Linkage detected an average of 1.73 and 10.36 lines respectively. In 99 out of 100 runs gpbM was able to detect all eight lines and only in one case it detected seven lines instead of eight. For gpbM, the error in the estimation of $\boldsymbol{\Theta}$ as a difference in the angle between the estimated and true lines in 3D, averaged for all eight lines over 100 runs was $0.214°$. The corresponding error in the estimation of $\boldsymbol{\alpha}$ as the norm of 2D deviation from $(0, 0, 0)$ was 0.02.

Star Lines. The data consisted of five inlier structures containing 50 points each arranged in 2D as a star (Fig.4). Each line was corrupted with zero mean Gaussian noise but with different standard deviation: 0.005, 0.01, 0.015, 0.02, 0.025. Additionally, 500 random outliers were also added uniformly between $(0, 0)$ and $(1, 1)$. Our results with $M = 500$ and $N = 200$ are compared with KF [6] and J-Linkage [21]. J-Linkage was given the true scale of inlier noise and minimum number of inliers present in each structure. Over 100 runs, KF and J-Linkage detected 3.74 and 2.85 lines on an average. In *all* 100 runs gpbM was able to detect all five lines. For gpbM, the difference in angle between the true and fitted lines averaged for all five lines over 100 runs was $0.312°$. The corresponding error in the intercept was 0.016.



Original data gpbM Kernel fitting J-Linkage

Figure 4. Star lines. There are 50 points per line and 500 random unstructured outliers. The gpbM always found all five lines.

4.2. Face Image Clustering

Clustering face images under varying illumination is an important and difficult problem [12]. We follow this work to compute the symmetric global affinity matrix with non-negative entries. We apply the gpbM algorithm to fit linear

subspaces in a low-dimensional representation of this affinity matrix and test it on data containing 500 images – 50 frontal face images of each of the *ten subjects* of Yale Face Database B. The number of subspaces (number of subjects) is *not known* a priori.

Each image $\mathbf{x}_i$ is vectorized and represented as a linear combination of the remaining images $\mathbf{x}_j, j = 1, \ldots, n$; $(n = 500)$ with weights w_{ij}

$$\mathbf{x}_i = \sum_{j, j \neq i} w_{ij} \mathbf{x}_j, \;\; i, j = 1, \ldots, n. \tag{14}$$

These weights are computed by solving a constrained least squares estimation problem, subject to $w_{ij} > 0$ and $w_{ii} = 0$. The weights are then stacked in a matrix $\mathbf{W}$ such that $\mathbf{W}(i,j) = w_{ij}$. This matrix is usually quite sparse due to less similarity between faces of different subjects, which is true for this database. The symmetric affinity matrix is formed as $\mathbf{A} = \left(\mathbf{W} + \mathbf{W}^\top\right)/2$ where $\mathbf{A}$ is 500×500. The matrix $\mathbf{A}$ is then normalized by computing $\mathbf{P} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}(i,i) = \sum_j \mathbf{A}(i,j) = \sum_j \mathbf{A}(j,i)$. The eigenvectors corresponding to the r largest eigenvalues of $\mathbf{P}$ form a $n \times r$ matrix $\mathbf{Q}$. Images of same subject taken under varying lighting conditions generally lie in a $d < r$ dimensional subspace [12].

The clustering algorithms proposed in [25] and [3] used frontal images of three out of the total ten subjects for evaluation of their techniques. However, in all their experiments, they used the uncropped face images where the presence of substantial background makes the problem relatively easier. For example, Fig. 5 shows the face data of the three subjects projected in three dimensions with and without the background. The subspaces are well separated in the first image due to different backgrounds of the three subjects.

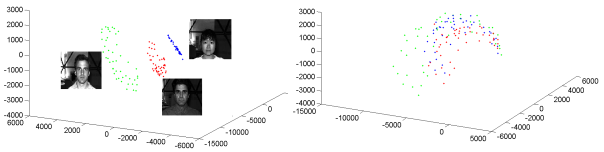


Figure 5. Frontal faces of three subjects of Yale Database B projected to 3D. *Left*: Face images with background as used in [25] and [3]. *Right*: Same images projected to 3D after cropping faces.

We evaluated our algorithm on all the 500 frontal face images, manually cropped to remove the background and downsampled to 64×64 size for faster computation. Fig. 6 shows three examples for each of the ten subjects. The $\mathbf{Q}$ matrix was formed by using first $r = 20$ significant eigenvectors. With $M = 5000$ and $N = 500$, we used the gpbM algorithm to fit *two-dimensional* subspaces, each corresponding to one subject in this 20-dimensional space. For 100 runs, the average, median and maximum errors over 500 images were 3.42%, 3.4% and 5.6%. Similar performance was achieved on data containing images of three to nine subjects (50 per subject), with the value of r varying between 10 to 18. Results are omitted due to lack of space and will be presented at the conference. In all experiments, the performance was slightly worse for $d = 3$. For $d = 2$ and $r = 20$, we tested the method of [3] on data containing the same 500 cropped images and it gave an error of 61.4%.

Figure 6. Example frontal face images of Yale Database B used in our experiments.

4.3. Projective Motion Segmentation

The task of a motion segmentation algorithm is to segment *multiple* rigid body motions using the point trajectories across multiple frames. Several approaches have been proposed which can be categorized into factorization based [22, 23], clustering based [7, 13], robust estimation based [9, 17, 5, 19], algebraic [25] and statistical methods [20, 11]. A brief review of most of these techniques can be found in [7]. Except [5] and [19], all other methods assume that the number of motions is known a priori. We focus on detecting multiple motions on the *Hopkins155* dataset without knowing the number of motions. We show comparisons on this dataset with state-of-the-art robust subspace estimation methods.

Projective motion factorization corresponds to estimating the motion subspace of an object in a dynamic scene perceived through perspective cameras. Consider only one inlier structure. If n_1 rigidly moving points lying on a single motion are tracked over F frames, then the $2F$ image coordinates obtained can be used to define feature vectors in $\mathbb{R}^{2F}$. In general, these n_1 vectors lie in a four-dimensional subspace of $\mathbb{R}^{2F}$ [22]. If the data is centered then the dimension of the subspace is only three. In homogeneous coordinates, the i^{th} image point in the j^{th} frame, $\mathbf{q}_i^j$ and its corresponding 3D world point $\mathbf{Q}_i$ are related as

$$\lambda_i^j \mathbf{q}_i^j = \mathbf{P}^j \mathbf{Q}_i, \;\; i = 1, \ldots, n_1; \; j = 1, \ldots, F \tag{15}$$

where λ_i^j is the projective depth of $\mathbf{q}_i^j$ and $\mathbf{P}^j$ is the 3×4 camera matrix for j^{th} frame. Equations (15) can be combined into a single factorization equation in matrix form as

$$\mathbf{T} = \begin{bmatrix} \lambda_1^1 \mathbf{q}_1^1 & \lambda_2^1 \mathbf{q}_2^1 & \ldots & \lambda_{n_1}^1 \mathbf{q}_{n_1}^1 \\ \lambda_1^2 \mathbf{q}_1^2 & \lambda_2^2 \mathbf{q}_2^2 & \ldots & \lambda_{n_1}^2 \mathbf{q}_{n_1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^F \mathbf{q}_1^F & \lambda_2^F \mathbf{q}_2^F & \ldots & \lambda_{n_1}^F \mathbf{q}_{n_1}^F \end{bmatrix} = \mathcal{M}\mathcal{S}$$

where $\mathcal{M}$ is the $3F \times 4$ motion matrix and $\mathcal{S}$ is the $4 \times n_1$ structure matrix. The unknown projective depths λ_i^j are estimated using the iterative method of [23].

The method starts by initializing all the depths to one. The rank-four approximation $\widetilde{\mathbf{T}}$ of $\mathbf{T}$ is computed using

SVD. The least-squares estimates of the depths are then obtained from $\widetilde{\mathbf{t}}_i^j$, the entries of $\widetilde{\mathbf{T}}$ corresponding to $\lambda_i^j \mathbf{q}_i^j$ as

$$\lambda_i^j = \left(\widetilde{\mathbf{t}}_i^j\right)^\top \mathbf{q}_i^j / ||\mathbf{q}_i^j||^2 \,. \tag{16}$$

where $\mathbf{q}_i^j$ is the original image point and λ_i^j, $\widetilde{\mathbf{t}}_i^j$ change in each iteration. The iterations for estimating $\widetilde{\mathbf{T}}$ and λ_i^j end when $\widetilde{\mathbf{T}}$ is within a small tolerance of $\mathbf{T}$.

A point $\mathbf{q}_i^j = [x_i^j,\ y_i^j,\ 1]^\top$, together with its depth λ_i^j, gives the vector of variables $\mathbf{y}_i = \left[(x_i^1,\ y_i^1,\ \lambda_i^1), \ldots (x_i^F,\ y_i^F,\ \lambda_i^F)\right]^\top$ and the corresponding carrier vector is given by $\mathbf{x}_i = \left[(\lambda_i^1 x_i^1\ \ \lambda_i^1 y_i^1\ \ \lambda_i^1) \ldots (\lambda_i^F x_i^F,\ \lambda_i^F y_i^F,\ \lambda_i^F)\right]^\top$. In this case both $\mathbf{y}_i$ and $\mathbf{x}_i$ lie in $\mathbb{R}^{3F}$. As opposed to the affine case [19], in projective motion estimation the carrier vector is heteroscedastic due to the multiplication of the image points with their depths. Assuming the noise corrupting the depth and the image coordinates to be identical and known upto a common scale σ^2, the first order approximation of the $3F \times 3F$ covariance matrix $\mathbf{C}_i$ of $\mathbf{x}_i$, computed using error propagation is $\mathbf{C}_i = \sigma^2 \mathbf{J}_{\mathbf{x}_i|\mathbf{y}_i}^\top \mathbf{J}_{\mathbf{x}_i|\mathbf{y}_i}$, where

$$\mathbf{J}_{\mathbf{x}_i|\mathbf{y}_i} = \begin{bmatrix} \boldsymbol{\mathcal{J}}_i^1 & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{J}}_i^2 & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \boldsymbol{\mathcal{J}}_i^F \end{bmatrix} \ ; \ \boldsymbol{\mathcal{J}}_i^j = \begin{bmatrix} \lambda_i^j & 0 & 0 \\ 0 & \lambda_i^j & 0 \\ x_i^j & y_i^j & 1 \end{bmatrix} .$$

For data containing multiple, *non-degenerate* motions, we estimate the motion subspaces in two steps. In the first step, assuming all the unknown depths to be equal, an affine motion estimation is performed. This step returns the initial estimates of all the motions along with their respective inliers. Due to the affine assumption, the inliers obtained for each motion are not necessarily completely correct. In the second step, for each motion, we construct a $\mathbf{T}$ matrix of the inlier points and apply projective correction to recover the depths (16). Each point is then multiplied with its depth and the modified data is subjected to multiple projective motion estimation using gpbM algorithm. Due to centering of the elemental subset data for hypotheses generation, the dimensionality of the null space is $(2F-3)$ for affine estimation and $(3F-3)$ for projective.

In both the steps, we use fundamental matrices to dichotomize the points lying between the boundary of the basin of attraction and the scale margin on either side of the mode. The inliers of each motion are used to robustly estimate the $(F-1)!$ fundamental matrices between all pairs of frames using gpbM. The carrier vector and its covariance matrix for fundamental matrix estimation were given in Section 2. A boundary point $\mathbf{x}_i$ is assigned to the motion for which the sum of residuals of the epipolar constraint, computed over *all* pairs of frames is minimum. For this simple *classification* problem, the eight point algorithm for estimating fundamental matrices is sufficient.

We present two groups of experiments. The *Hopkins155* dataset has 155 sequences without unstructured outliers. The *parking lot* sequence with three moving cars, has unstructured outliers too. For consistency, all the examples are processed with the values $M = 500$ and $N = 500$, which are sometimes too large. In all the experiments, we use only every 6th or 7th frame in the sequence, so the number of frames $F = 5$.

***Hopkins155* Dataset.** This dataset is available online at http://www.vision.jhu.edu/data/hopkins155 and consists of 120 two-motion and 35 three-motion sequences which are divided into three categories – traffic, articulated and checkerboard. The gpbM algorithm determines the number of motions and the points belonging to each motion automatically, without *any* user input. We compare the performance of our algorithm with five other methods – Generalized PCA [25], RANSAC, Local Subspace Affinity (LSA) [27], pbM [19] and the Ordered Residual Kernel (ORK) method [5]. The classification error is computed similar to [5] and [25]. Except pbM and ORK, all other methods rely on the user to specify the actual number of motions present in the data. Additionally, RANSAC also requires an estimate of the scale of inlier noise. To our knowledge, ORK [5] has reported the best results on the Hopkins155 dataset without any user intervention.

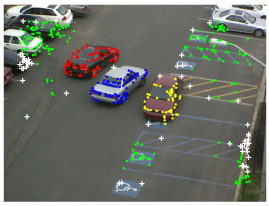
Tables 1 and 2 compare the results obtained by various methods on two and three-motion sequences. The results of the REF (reference/control) method, generated for benchmarking, were obtained using the ground truth information. Refer [24] for details. The results for GPCA, LSA and RANSAC were obtained from [24]. The code for pbM was obtained from http://coewww.rutgers.edu/riul/research/code.html. The results of ORK [5] were not reported for individual categories. Since gpbM is based on random sampling, the results reported here are averaged over 100 runs for each sequence. The results of pbM are averaged over 20 runs.

Table 1. Percentage classification errors for 2-motion sequences. Note, only pbM, ORK and gpbM are completely user independent.

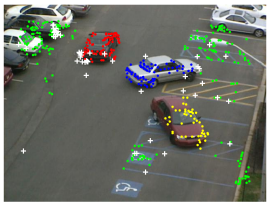
Method	REF	GPCA	LSA	RANSAC	pbM	ORK	gpbM
Traffic: 31 sequences							
Mean	0.30	1.41	5.43	2.55	18.52	–	5.23
Articulated: 11 sequences							
Mean	1.71	2.88	4.10	7.25	15.18	–	6.41
Checkerboard: 78 sequences							
Mean	2.76	6.09	2.57	6.52	32.43	–	8.48
All: 120 sequences							
Mean	2.03	4.59	3.45	5.56	28.25	7.83	**7.60**

Additionally, we obtained a median error of 5.6% for 2-motion and 6.2% for 3-motion sequences. Results can be further improved by handling degeneracies in the data.

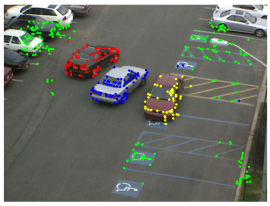
Dataset with Unstructured Outliers. The sequence contains four motions (background and three moving cars). The points across various frames were matched using [10].



(a)



(b)



(c)

	M1	M2	M3	M4	Outliers
M1	212	1	0	0	0
M2	0	78	0	0	0
M3	0	0	75	0	0
M4	0	0	0	46	0
Outliers	3	0	4	3	52

(d)

Figure 7. Sequence with four motions and unstructured outliers. (a) and (b) First and last frames with various motions marked. White points marked + show unstructured outliers. (c) Motion factorization results using gpbM (only inliers are shown). (d) Confusion matrix. M1, M2, M3 and M4 correspond to background, black car, silver car and maroon car respectively.

Table 2. Percentage classification errors for 3-motion sequences. Note, only pbM, ORK and gpbM are completely user independent.

Method	REF	GPCA	LSA	RANSAC	pbM	ORK	gpbM
Traffic: 7 sequences							
Mean	1.30	19.83	25.07	12.83	22.00	–	3.10
Articulated: 2 sequences							
Mean	2.66	16.85	7.25	21.38	18.32	–	4.28
Checkerboard: 26 sequences							
Mean	6.28	31.95	5.80	10.38	26.08	–	11.10
All: 35 sequences							
Mean	5.08	28.66	9.73	22.94	25.26	12.62	**9.64**

In total there were 474 points – 213 on the background, 78 on first car (black), 75 on second car (silver), 46 on third car (maroon) and 62 unstructured outliers. Fig. 7 shows the motion segmentation results using gpbM along with the corresponding confusion matrix.

5. Conclusions

We presented a robust estimation method called the generalized projection based M-estimator (gpbM) which can estimate multiple heteroscedastic inlier structures *without* any user intervention. We showed its performance on challenging synthetic and real-world applications, but *Hopkins155* dataset with synthetic outliers was not considered.

6. Acknowledgments

We would like to thank Raghav Subbarao for providing some initial ideas about this work.

References

[1] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45:891–923, 1998.

[2] A. Bab-Hadiashar and D. Suter. Robust segmentation of visual data using ranked unbiased scale estimate. *Robotica*, 17:649–660, 1999.

[3] G. Chen and G. Lerman. Spectral curvature clustering (SCC). *IJCV*, 81:317–330, 2009.

[4] H. Chen and P. Meer. Robust regression with projection based M-estimators. In *ICCV03*, volume II, pages 878–885, Oct 2003.

[5] T. J. Chin, H. Wang, and D. Suter. The ordered residual kernel for robust motion subspace clustering. In *Advances in NIPS09*, pages 333–341. 2009.

[6] T. J. Chin, H. Wang, and D. Suter. Robust fitting of multiple structures: The statistical learning approach. In *ICCV09*, pages 413 – 420, 2009.

[7] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR09*, pages 2790–2797, 2009.

[8] L. Fan and T. Pylvänäinen. Robust scale estimation from ensemble inlier sets for random sample consensus methods. In *ECCV '08*, pages 182–195, 2008.

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24:381–395, 1981.

[10] B. Georgescu and P. Meer. Point matching under large image deformations and illumination changes. *PAMI*, 26:674–689, 2004.

[11] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the EM algorithm. In *CVPR04*, volume I, pages 707–714, 2004.

[12] J. Ho, M. H. Yang, J. Lim, K. C. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *CVPR03*, pages 11–18, 2003.

[13] F. Lauer and C. Schnorr. Spectral clustering of linear subspaces for motion segmentation. In *ICCV09*, pages 678–685, 2009.

[14] K. M. Lee, P. Meer, and R. H. Park. Robust adaptive segmentation of range images. *PAMI*, 20:200–205, 1998.

[15] B. Matei and P. Meer. Estimation of nonlinear errors-in-variables models for computer vision applications. *PAMI*, 28:1537–1552, 2006.

[16] R. Raguram, J. M. Frahm, and M. Pollefeys. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In *ECCV08*, pages 500–513, 2008.

[17] S. R. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. *CVPR08*, pages 1–8, 2008.

[18] R. Subbarao and P. Meer. Beyond RANSAC: User independent robust regression. In *Workshop on 25 Years of RANSAC,* New York, NY, June 2006.

[19] R. Subbarao and P. Meer. Subspace estimation using projection based M-estimators over Grassmann manifolds. In *ECCV06*, volume I, pages 301–312, 2006.

[20] Y. Sugaya and K. Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. In *SMVP04*, pages 13–25. 2004.

[21] R. Toldo and A. Fusiello. Robust multiple structures estimation with J-linkage. In *ECCV08*, pages 537–547, 2008.

[22] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9:137–154, 1992.

[23] B. Triggs. Factorization methods for projective structure and motion. In *CVPR96*, volume I, pages 845–851, 1996.

[24] R. Tron and R.Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *CVPR07*, pages 1–8, 2007.

[25] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *PAMI*, 27:1–15, 2005.

[26] H. Wang and D. Suter. Robust fitting by adaptive-scale residual consensus. In *ECCV04*, volume III, pages 107–118, 2004.

[27] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV06*, pages 94–106, 2006.