Generalized Projection-Based M-Estimator

Sushil Mittal, *Member, IEEE*, Saket Anand, *Student Member, IEEE*, and Peter Meer, *Fellow, IEEE*

Abstract—We propose a novel robust estimation algorithm—the *generalized projection-based M-estimator* (gpbM), which does not require the user to specify any scale parameters. The algorithm is general and can handle heteroscedastic data with multiple linear constraints for single and multicarrier problems. The gpbM has three distinct stages—scale estimation, robust model estimation, and inlier/outlier dichotomy. In contrast, in its predecessor pbM, each model hypotheses was associated with a different scale estimate. For data containing multiple inlier structures with generally different noise covariances, the estimator iteratively determines one structure at a time. The model estimation can be further optimized by using Grassmann manifold theory. We present several homoscedastic and heteroscedastic synthetic and real-world computer vision problems with single and multiple carriers.

Index Terms—Generalized projection-based M-estimator, robust estimation, heteroscedasticity, RANSAC

✦

1 INTRODUCTION

REGRESSION algorithms estimate a parameter vector given a dataset and a functional relation between the data and the parameters. Data points satisfying the relation are known as *inliers*, while the ones not satisfying it are called *outliers*. Outliers interfere with the regression and lead to incorrect results unless they are appropriately accounted for. Robust algorithms estimate model parameters in the presence of outliers without significant loss of accuracy.

In vision applications, outliers almost always occur and any system which aims to solve even simple visual tasks must address this problem. The most widely used robust algorithm in computer vision is *Random Sample Consensus* (RANSAC) [12]. The popularity of the original RANSAC algorithm is due to its ease of implementation. However, RANSAC suffers from a major drawback, namely, sensitivity to *scale*. Scale is the measure of additive noise which corrupts the inliers. RANSAC requires a user-specified estimate of the scale and the performance of RANSAC is sensitive to it. Using a low value of the scale leads to rejecting valid inlier data, while a large value lets the outliers affect the parameter estimate.

Computer vision problems have evolved into estimating multiple inlier structures in the presence of a large number of outliers. For an inlier structure, since the scale can be different in each dimension, it is very hard (sometimes impossible) for the user to find a good estimate. For example, in video sequences, the scale of the inlier noise could change from one frame to another, based on how fast the camera is moving.

- S. Mittal is with the Department of Statistics, Columbia University, New York, NY 10027. E-mail: mittal@stat.columbia.edu.
- S. Anand and P. Meer are with the CAIP Center, Department of Electrical and Computer Engineering, Rutgers University, CoRE Building 96 Frelinghuysen Road, Piscataway, NJ 08854.
  E-mail: anands@eden.rutgers.edu, meer@jove.rutgers.edu.

RANSAC has been applied to many computer vision problems. See [17] for references. The various enhancements of RANSAC, like MLESAC [33], LO-RANSAC [7], PROSAC [6], and QDEGSAC [13], proposed changes to either the cost function, the sampling method, or detecting the degeneracies in data. A comprehensive review of all these algorithms, along with a new method for real-time implementation of RANSAC, is given in [24]. Cov-RANSAC [25] proposed a modification to RANSAC by explicitly incorporating the uncertainty of the estimation procedure, but it still required a threshold on the trace of the covariance matrix.

Estimating the scale of the inlier noise is an important problem for any robust regression algorithm. Among the first papers addressing this issue were the robust Kth Ordered Scale Estimator (KOSE), Adaptive Least Kth order Squares (ALKS) [20], and Weighted Median Absolute Deviation (WMAD) [10] methods. All the above and different versions of the projection-based M-estimator (pbM) [28], [29] (explained below) used variants of the Median Absolute Deviation (MAD) scale estimate. The MAD-based scale estimation could fail if the inliers contain noise from an asymmetric distribution or comprise less than 50 percent of the data points. This is often the case when several inlier structures are present.

The Modified Selective Statistical Estimator (MSSE) [1], the Two-Step Scale Estimator (TSSE) [39] and the kernel consensus-based robust estimator [38] also return a scale estimate. The MSSE used generalization of the Least Median of Squares and the user had to specify an initial estimate of the minimum acceptable population of every inlier structure. The TSSE is valid only for data with symmetric inlier distribution around the mode and used mean shift to find an inlier-outlier dichotomy followed by estimating the scale.

The main disadvantage of *all* these methods is that they estimate the scale of noise independently in 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
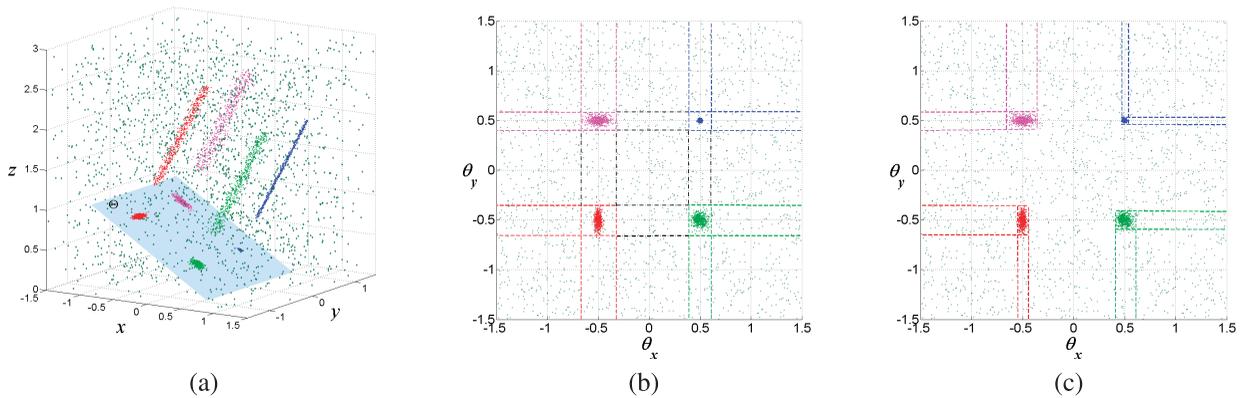
(a) (b) (c)

Fig. 1. Scale estimation in multiple dimensions. (a) Data containing multiple inlier structures and outliers. (b) Scale of noise estimated independently in each dimension of the 2D null space, Θ (*incorrect*). (c) Scale estimated simultaneously in both the dimensions (*correct*).

noise along the two axes of the null space, Θ. Here, the null space is the blue plane shown in Fig. 1a. It is impossible to estimate the scale correctly if the estimation is done along each axis of Θ independently.

Recently, a few fully automatic methods have also been proposed. The Ordered Residual Kernel (ORK)-based method [3], [4] was introduced where the scale estimate is not computed explicitly, but the method suffers from poor computational efficiency and sensitivity to unstructured (gross) outliers. The Stable Random Sample Consensus (StaRSaC), introduced in [5], uses RANSAC to estimate a set of model parameters by varying the threshold in a prespecified range and then selects the model parameters corresponding to the smallest variance. The rather complicated REsidual CONsensus (RECON) approach of [23] first finds a given number of mutually consistent models using elemental subsets and then refines the model estimate using PROSAC [6] type sampling on the inliers. Both StaRSaC and RECON work only on data containing one inlier structure.

The projection-based M-estimator was described in [28] and [29]. It estimates $m - k$ dimensional subspaces in $\mathbb{R}^m$ by maximizing the M-score over randomly chosen subspace hypotheses. In general, the elements of a carrier vector $\mathbf{x}$ are nonlinear monomials of the elements of vector of variables $\mathbf{y}$. See Section 2 for details. Given the carrier vectors $\mathbf{x}$, the M-score relates to the M-estimation (see (13)) and solves a linearized system with unknowns $\Theta \in \mathbb{R}^{m \times k}$, representing the k-dimensional null space, and $\boldsymbol{\alpha} \in \mathbb{R}^k$, the intercept. Due to this nonlinearity, the estimation problem becomes heteroscedastic, i.e., each carrier vector can have a different noise covariance matrix. The noise in variables $\mathbf{y}$ is usually homoscedastic, i.e., the covariances of the noisy measurements are equal, but heteroscedastic variables can also be easily handled. The covariances of the carriers are computed from those of the variables through error propagation [21] (see Section 2 for details).

For pbM, the computation of the M-score depends on the $k \times k$ diagonal scale matrix $\mathbf{S}_\Theta$ whose diagonal entries were computed for $p = 1, \ldots, k$ using

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

where $z_i^p = \boldsymbol{\theta}_p^\top \mathbf{x}_i$ is the projection of the ith carrier vector, $\mathbf{x}_i$ on to the pth column of Θ. One can see that pbM uses a variant of MAD. Since $\mathbf{S}_\Theta$ depends on a particular Θ, it *does*

not necessarily correspond to the actual scale of the inlier noise. The independence of the M-score over the Θ-dependent scale is only *partially* achieved by normalizing each M-score with the determinant of $\mathbf{S}_\Theta$ [29]. Often, mean shift cannot converge to the correct mode using an incorrect scale estimate, especially when data contains asymmetric noise and multiple inlier structures. Using a Θ-dependent scale reduces the discrimination between correct and incorrect hypotheses.

In addition to lacking a stable method for automatic scale estimation, the pbM algorithm [29], [28] cannot handle *multicarrier* problems, i.e., when more than one carrier vector arises from a single measurement. For example, in planar homography estimation and camera calibration, a single point correspondence yields two carrier vectors—one each corresponding to the x and y coordinates of the images.

Let us see the following two-carrier synthetic example. In Fig. 2a, the two surfaces in 3D correspond to the paraboloid $4x^2 + 4y^2 - z = 0$ and the plane $4x + 4y - z = 0$. It can be verified that their intersection is always an ellipse (shown in red). The vector of variables is given by $[x\ y\ z]^\top$ and the two carrier vectors are $[x^2\ y^2\ z]^\top$ and $[x\ y\ z]^\top$. Since we have a multicarrier problem, the vector $\boldsymbol{\theta} = \frac{1}{\sqrt{33}}[4\ \ 4\ \ -1]^\top$ is the common null space for both surfaces and the corresponding intercept $\alpha = 0$. Given a point on the ellipse such that the two carrier vectors are linearly independent, the equation of the ellipse in the plane can be obtained from the estimated $\boldsymbol{\theta}$.

Fig. 2b shows 100 noisy inlier points along the ellipse obtained by independently corrupting x, y, z coordinates with zero-mean Gaussian noise and standard deviation
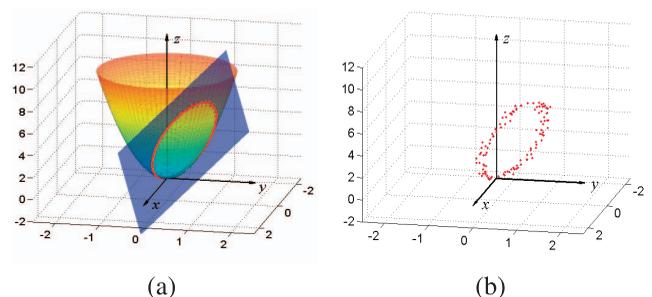

(a) (b)

Fig. 2. A synthetic two-carrier problem. (a) Intersection of a paraboloid and a plane in 3D gives an ellipse. (b) The noisy inlier points along the ellipse.

$\sigma = 0.0707 = \sqrt{0.005}$. The scale of the inlier noise of the two carriers, using the true $\boldsymbol{\theta}$, can be computed either homoscedastically or heteroscedastically. Since here the data contains only inliers, the estimates of the scales are computed as the standard deviations of their residuals after projecting them to the null space, $\boldsymbol{\theta}$. In the homoscedastic case, the scale estimates are directly computed as $\widehat{s}_{1hom} = 0.1084$ and $\widehat{s}_{2hom} = 0.0770$. In the heteroscedastic case, the scale of the residuals of the paraboloid is computed after first normalizing them with their point-dependent covariances. See Section 2 for computational details. The scale estimates in this case are $\widehat{s}_{1het} = 0.0772$ and $\widehat{s}_{2het} = 0.0770$, which are comparable for the two carriers.

We will present the generalized projection-based M-estimator (gpbM) algorithm for estimating multiple, linearized inlier structures in the presence of unstructured outliers. In general, the carriers are heteroscedastic. The algorithm has three different steps. 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 by using just the inliers returned by the scale estimate. Finally, the inlier/outlier dichotomy is performed using the estimated scale and model parameters. Being completely user independent, this method has obvious advantages over RANSAC-like algorithms and pbM [29].

The contributions of our work are summarized below.

- 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, arising from both single and multicarrier problems.
- We formulate the model refinement step as an optimization problem over Grassmann manifolds, which is also described in detail in [22].
- We introduce a new method for inlier/outlier dichotomy, also taking into account the discrimination between intersecting inlier structures.

In Section 2, we introduce 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 challenging synthetic and real-world applications. Finally, in Section 5, we conclude and provide some interesting directions for extending the presented work into the future.

2 ROBUST SUBSPACE ESTIMATION

Solving an estimation problem requires determining a set of parameters from noisy image measurements using constraints. Given n_1 true values of the measurements (inliers) $\mathbf{y}_{io}$, the vector of constraints generally represents a nonlinear relationship between $\mathbf{y}_{io}$ and ideal values of the parameters, $\boldsymbol{\beta}$:

$$\boldsymbol{\Psi}(\mathbf{y}_{io}, \boldsymbol{\beta}) = \mathbf{0}_k, \ i = 1, \ldots, n_1, \ \boldsymbol{\Psi}(\cdot) \in \mathbb{R}^k. \quad (2)$$

In computer vision problems, this nonlinear relationship (2) can be linearized (upto the first order) into a set of k linearly independent constraints parametrized by the matrix $\boldsymbol{\Theta} \in \mathbb{R}^{m \times k}$ and the intercept vector $\boldsymbol{\alpha} \in \mathbb{R}^k$:

$$\boldsymbol{\Phi}(\mathbf{x}_{io}^{[c]}, \boldsymbol{\Theta}, \boldsymbol{\alpha}) = \boldsymbol{\Theta}^\top \mathbf{x}_{io}^{[c]} - \boldsymbol{\alpha} = \mathbf{0}_k, \quad i = 1, \ldots, n_1. \quad (3)$$

The set of true carrier vectors is $\mathbf{x}_{io}^{[c]} \in \mathbb{R}^m$. In a multicarrier problem, a variable $\mathbf{y}$ provides $\zeta > 1$ carrier vectors $\mathbf{x}^{[c]}, c = 1, \ldots, \zeta$. The $m \times k$ $(k < m)$ 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 carrier data. The multiplicative ambiguity is resolved by requiring $\boldsymbol{\Theta}^\top \boldsymbol{\Theta} = \mathbf{I}_{k \times k}$. Due to this property, the $m \times k$ matrix $\boldsymbol{\Theta}$ can also be represented as a point on a Grassmann manifold of appropriate dimensions.

Given n $(> n_1)$ noisy measurements $\mathbf{y}_i$ (inliers and outliers) and constraints, we can compute all the carrier vectors $\mathbf{x}_i^{[c]}, i = 1, \ldots, n, c = 1, \ldots, \zeta$. The noise in the carriers corresponding to the inliers is assumed to be additive $\mathbf{x}_i^{[c]} = \mathbf{x}_{io}^{[c]} + \delta \mathbf{x}_i^{[c]}, \ \delta \mathbf{x}_i^{[c]} \sim GI(\mathbf{0}, \sigma^2 \mathbf{C}_{\mathbf{x}_i^{[c]}})$. The number of inliers n_1 is unknown. The problem of robust linear subspace estimation is to find the estimates of $\boldsymbol{\Theta}, \boldsymbol{\alpha}$, and n_1 such that

$$\boldsymbol{\Theta}^\top \mathbf{x}_i^{[c]} - \boldsymbol{\alpha} \approx \mathbf{0}_k, \quad i = 1, \ldots, n_1, \quad c = 1, \ldots, \zeta. \quad (4)$$

The corrected carriers $\widehat{\mathbf{x}}_i^{[c]}$ and the variables can then be recovered easily. The points $\mathbf{x}_i^{[c]}, i = n_1 + 1, \ldots, n$, 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.

We present two examples for computing carrier covariances heteroscedastically, one for a single and the other for a multicarrier problem. More examples, together with detailed experiments will be presented in Section 4.

In fundamental matrix estimation, $\zeta = 1$. Traditionally, the vector of measured variables is given by $\mathbf{y} = [x_1 \ y_1 \ x_2 \ y_2]^\top$ which 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 3×3 matrix $\mathbf{F}$ satisfies the 1D constraint in the homogeneous coordinates (for nonideal points only [17, p. 279]):

$$[x_2 \quad y_2 \quad 1] \mathbf{F} [x_1 \quad y_1 \quad 1]^\top = 0. \quad (5)$$

The carrier vector is given by

$$\mathbf{x} = [x_1 \ y_1 \ x_2 \ y_2 \ x_1 x_2 \ x_1 y_2 \ y_1 x_2 \ y_1 y_2]^\top,$$

which lies in $\mathbb{R}^8$, implying that $\boldsymbol{\theta} \in \mathbb{R}^8$. The constraint $\boldsymbol{\theta}^\top \boldsymbol{\theta} = 1$ eliminates the multiplicative ambiguity in (5). Assuming the variables $\mathbf{y}$ have covariance $\sigma^2 \mathbf{I}_{4 \times 4}$, the first order approximation of the covariance of $\mathbf{x}$ is computed from the Jacobian using error propagation [21]:

$$\mathbf{J}_{\mathbf{x}|\mathbf{y}} = \begin{bmatrix} 1 & 0 & 0 & 0 & x_2 & y_2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & x_2 & y_2 \\ 0 & 0 & 1 & 0 & x_1 & 0 & y_1 & 0 \\ 0 & 0 & 0 & 1 & 0 & x_1 & 0 & y_1 \end{bmatrix} = [\mathbf{I}_{4 \times 4} \ \mathbf{J}(\mathbf{y})], \quad (6)$$

$$\mathbf{C}_{\mathbf{x}} = \sigma^2 \mathbf{J}_{\mathbf{x}|\mathbf{y}}^\top \mathbf{I}_{4 \times 4} \mathbf{J}_{\mathbf{x}|\mathbf{y}} = \sigma^2 \begin{bmatrix} \mathbf{I}_{4 \times 4} & \mathbf{J}(\mathbf{y}) \\ \mathbf{J}(\mathbf{y})^\top & \mathbf{J}(\mathbf{y})^\top \mathbf{J}(\mathbf{y}) \end{bmatrix}. \quad (7)$$

Next, consider the problem depicted in Fig. 2. Given the vector of variables $\mathbf{y} = [x \; y \; z]^\top$, the $\zeta = 2$ carrier vectors are given by

$$\mathbf{x}^{[1]} = [x^2 \quad y^2 \quad z]^\top \quad \mathbf{x}^{[2]} = [x \quad y \quad z]^\top, \qquad (8)$$

which lie in $\mathbb{R}^3$, implying that $\boldsymbol{\theta} \in \mathbb{R}^3$. Assuming that the variables $\mathbf{y}$ have covariance $\sigma^2 \mathbf{I}_{3\times 3}$, the first order approximations of the covariances of $\mathbf{x}^{[1]}$ and $\mathbf{x}^{[2]}$ are computed as

$$\mathbf{J}_{\mathbf{x}^{[1]}|\mathbf{y}} = \begin{bmatrix} 2x & 0 & 0 \\ 0 & 2y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{J}_{\mathbf{x}^{[2]}|\mathbf{y}} = \mathbf{I}_{3\times 3}, \qquad (9)$$

$$\mathbf{C}_{\mathbf{x}^{[1]}} = \sigma^2 \mathbf{J}_{\mathbf{x}^{[1]}|\mathbf{y}}^\top \mathbf{I}_{3\times 3} \mathbf{J}_{\mathbf{x}^{[1]}|\mathbf{y}} = \sigma^2 \begin{bmatrix} 4x^2 & 0 & 0 \\ 0 & 4y^2 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{C}_{\mathbf{x}^{[2]}} = \sigma^2 \mathbf{J}_{\mathbf{x}^{[2]}|\mathbf{y}}^\top \mathbf{I}_{3\times 3} \mathbf{J}_{\mathbf{x}^{[2]}|\mathbf{y}} = \sigma^2 \mathbf{I}_{3\times 3}. \qquad (10)$$

In case of a multicarrier problem, for a given model hypothesis $[\boldsymbol{\Theta}, \boldsymbol{\alpha}]$, the projections of points $\mathbf{x}_i^{[c]}$, $i = 1, \dots, n$ and $c = 1, \dots, \zeta$, onto $\boldsymbol{\Theta}$ are given by $\mathbf{z}_i^{[c]} = \boldsymbol{\Theta}^\top \mathbf{x}_i^{[c]}$. A point $\mathbf{y}_i$ is considered an inlier if *all the corresponding carriers* $\mathbf{x}_i^{[c]}$, $c = 1, \dots, \zeta$, satisfy that the Mahalanobis distances

$$d_i^{[c]} = \sqrt{\left(\boldsymbol{\Theta}^\top \mathbf{x}_i^{[c]} - \boldsymbol{\alpha}\right)^\top \left(\mathbf{H}_i^{[c]}\right)^{-1} \left(\boldsymbol{\Theta}^\top \mathbf{x}_i^{[c]} - \boldsymbol{\alpha}\right)} \approx 0, \qquad (11)$$

where $\boldsymbol{\Theta}^\top \mathbf{x}_i^{[c]} - \boldsymbol{\alpha}$ are deviations of the points $\mathbf{x}_i^{[c]}$ from the model hypothesis and $\mathbf{H}_i^{[c]} = \boldsymbol{\Theta}^\top \mathbf{C}_i^{[c]} \boldsymbol{\Theta}$ are the covariance matrices of the projections $\mathbf{z}_i^{[c]}$. We use $\mathbf{C}_i^{[c]}$, $\mathbf{H}_i^{[c]}$ instead of $\mathbf{C}_{\mathbf{x}_i^{[c]}}$, $\mathbf{H}_{\mathbf{x}_i^{[c]}}$ for convenience. Thus, for the point $\mathbf{y}_i$ to be an inlier, it is sufficient to check if

$$d_i^{[\widetilde{c}]} \approx 0 \quad \text{where} \quad \widetilde{c} = \underset{c=1,\dots,\zeta}{\arg\max} \; d_i^{[c]}. \qquad (12)$$

The carrier vector corresponding to $\widetilde{c}$ and its projection onto $\boldsymbol{\Theta}$ are represented by $\widetilde{\mathbf{x}}_i$ and $\widetilde{\mathbf{z}}_i$ and their respective covariance matrices by $\widetilde{\mathbf{C}}_i$ and $\widetilde{\mathbf{H}}_i$. Most robust estimation algorithms, including RANSAC, use the euclidean distance instead of the Mahalanobis metric. This is similar to homoscedastic estimation, and in a lot of cases could lead to serious inaccuracies in distinguishing between the inliers and the outliers.

We assume that data is nondegenerate. Degenerate data can first be processed similar to QDEGSAC [13], followed by the generalized projection-based M-estimation. Given all the k constraints, the m-dimensional carrier vectors can be easily computed. No other prior knowledge about the inlier structures is used.

3 GENERALIZED PROJECTION-BASED M-ESTIMATOR

We define the heteroscedastic objective function as

$$[\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{\alpha}}] = \underset{\boldsymbol{\Theta}, \boldsymbol{\alpha}}{\arg\max} \frac{1}{n} \sum_{i=1}^{n} \frac{K\left(\left(\left(\widetilde{\mathbf{z}}_i - \alpha\right)^\top \widetilde{\mathbf{B}}_i^{-1}\left(\widetilde{\mathbf{z}}_i - \alpha\right)\right)^{\frac{1}{2}}\right)}{\sqrt{\det \widetilde{\mathbf{B}}_i}}, \quad (13)$$

where $\widetilde{\mathbf{z}}_i = \mathbf{z}_i^{[\widetilde{c}]} = \boldsymbol{\Theta}^\top \widetilde{\mathbf{x}}_i$ is the projection of $\mathbf{x}_i^{[\widetilde{c}]}$ on to $\boldsymbol{\Theta}$. Note that for each point i, the choice of $\widetilde{c}$ is dependent on a particular model hypothesis $[\boldsymbol{\Theta}, \boldsymbol{\alpha}]$ and is determined using (12). Assuming that the data is nondegenerate, the bandwidth matrix $\widetilde{\mathbf{B}}_i = \mathbf{S}^\top \widetilde{\mathbf{H}}_i \mathbf{S}$ is positive definite. The $k \times k$ diagonal scale matrix $\mathbf{S}$ is computed beforehand (see Section 3.1) and corresponds to the scale of inlier noise in k dimensions. The division by $\sqrt{\det \widetilde{\mathbf{B}}_i}$ ensures that points with larger covariances contribute less than the ones with smaller covariances. Note that in [29], since the hypothesis-dependent scale matrix $\mathbf{S}_{\boldsymbol{\Theta}}$ is the same for all the points, it cannot be used for general heteroscedastic problems.

The gpbM algorithm derives its name due the relation of the kernel function, $K(u)$ to the M-estimator loss function, $\rho(u)$ which is given by $K(u) = 1 - \rho(u)$. The function $\rho(u)$ is a redescending M-estimator and is nonnegative, symmetric, and nondecreasing with $|u|$. It has a unique minimum of $\rho(0) = 0$ and a maximum of one for $|u| > 1$. We will use Epanechnikov or Gaussian kernels in our experiments.

The gpbM algorithm falls in the category of elemental subset-based estimation methods. Each inlier structure is estimated in three steps. In the *first* step, we estimate the $k \times k$ diagonal scale matrix $\mathbf{S}$, heteroscedastically. In the *second* step, the parameter pair $[\boldsymbol{\Theta}, \boldsymbol{\alpha}]$ is estimated using the scale computed in the first step. We show that maximizing the objective function of (13) is equivalent to maximizing a heteroscedastic kernel density function over the projections, $\widetilde{\mathbf{z}}_i$. In the *third* step, we perform inlier/outlier dichotomy. After *all* the inlier structures are recovered iteratively, a final reclassification of the points using the discrimination between the estimated models can also be performed.

3.1 Step One: Heteroscedastic Scale Estimation

For detecting multiple inlier structures, if the scale of inlier noise is overestimated, two or more inlier structures might get merged into one. On the other hand, if the scale is underestimated, multiple models might get fitted within one inlier structure. The inliers are always densely packed around the regression surface and, to estimate the scale, we capture the difference in density between the inliers and outliers. It is impossible to estimate the scale if both inliers and ouliers have similar densities.

As opposed to [27], the scale is estimated heteroscedastically. A homoscedastic scale estimate could lead to gross errors, especially in multicarrier problems. The scale of an inlier structure is computed by first estimating the approximate fraction of data points belonging to it. We generate M elemental subset-based model hypotheses $[\boldsymbol{\Theta}, \boldsymbol{\alpha}]$. The value of M is specific to the problem and will be given in Section 4. For each hypothesis, we vary the value of fraction between $(0, 1]$ in Q steps, such that for $q = 1, \dots, Q$, the fraction $\eta_q = q/Q = n_q/n$. In all our experiments, the value of Q was conservatively set to 40.

For a given q, with a slight abuse of the notation, let

$$vol^q(\boldsymbol{\Theta}, \boldsymbol{\alpha}) = \sqrt{\sum_{l=1}^{n_q} \left(\widetilde{\mathbf{z}}_l - \boldsymbol{\alpha}\right)^\top \widetilde{\mathbf{H}}_l^{-1}\left(\widetilde{\mathbf{z}}_l - \boldsymbol{\alpha}\right)}, \qquad (14)$$
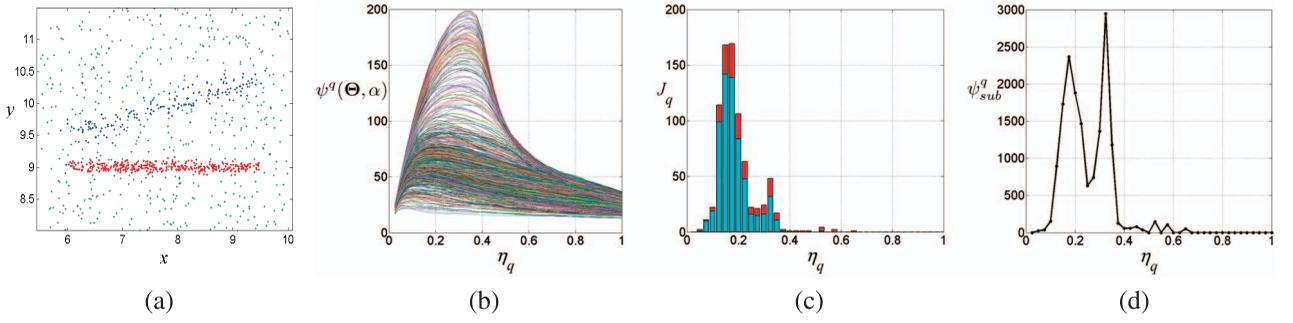
Fig. 3. Inlier fraction estimation. (a) Original data with two inlier structures. (b) Density plots for all 800 randomly generated hypotheses. (c) Histogram of peaks J_q. The red region in the top of each bar corresponds to the highest $\eta_q J_q$ peaks. (d) Sum of the highest $\eta_q J_q$ peaks, ψ^q_{sub}. The location of highest peak corresponds to the estimated fraction of the inlier structure with $\eta_{\widehat{q}} = 0.35$.

be the volume around the intercept $\boldsymbol{\alpha}$ containing n_q points. Since we incorporate the Mahalanobis distance into the volume (14), our method is more robust to inlier noise than [27]. The corresponding density for the fraction η_q is proportional to the inverse of the volume and is given by

$$\psi^q(\boldsymbol{\Theta}, \boldsymbol{\alpha}) = \frac{n_q}{vol^q(\boldsymbol{\Theta}, \boldsymbol{\alpha}) + \epsilon}, \tag{15}$$

where a small constant ϵ is added to suppress extremely high values of densities obtained for small fractions. Usually, the value of ϵ is set 0.01 or 0.1.

For each of the M hypotheses, we are only interested in the maximum (peak) density, $\max_q[\psi^q(\boldsymbol{\Theta}, \boldsymbol{\alpha})]$. Let J_q be the number of times a peak occurs at η_q fraction. It is clear that the sum $\sum_{q=1}^{Q} J_q = M$. For every q, let ψ^q_{sub} be the *sum* of $\eta_q J_q$ peak density values. For example, for $q = 4$ ($\eta_q = 0.1$), only the sum of highest $0.1 J_4$ peaks is considered. This way, the estimation becomes more robust for data containing multiple inlier structures, especially when various inlier structures have very different number of points. Our estimate of the fraction of the inlier points is determined as $\eta_{\widehat{q}}$ where

$$\widehat{q} = \arg\max_q \psi^q_{sub}. \tag{16}$$

Consider the two inlier structures in Fig. 3a lying along two different lines in 2D. The blue and red structures contain 150 points and 350 points, respectively, and 500 unstructured outliers were also added. The x and y coordinates of the two structures were independently corrupted with zero-mean Gaussian noise with standard deviation of 0.1 for the red and 0.05 for the blue structure.

The histogram in Fig. 3c shows that more peaks occur at $\eta_q = 0.15$ than at $\eta_q = 0.35$. For a given q, some peaks could correspond to incorrect hypotheses and their number depends both on the number of inlier structures and the fraction of each inlier structure. The smaller inlier fraction ($\eta_q = 0.15$) has more peaks since more incorrect hypotheses can produce peaks at these fractions. Also, some hypotheses taken from the larger inlier structure can correspond to peaks at $\eta_q = 0.15$. For each q, summing over just $\eta_q J_q$ peaks ensures that only the most reliable peaks are considered.

Fig. 3d plots ψ^q_{sub} for each fraction. The factor η_q also biases the estimation toward detecting the inlier structure containing more points. It is computationally more advantageous to detect and remove this inlier structure first so that fewer points are left for the subsequent iterations of the algorithm. If the data contains multiple inlier structures

with similar noise and number of points, then the scale of one of them is estimated first. Similar results were also obtained for the example in [27, Fig. 2].

The projections $\widetilde{\mathbf{z}}_i$ of the data points are computed by projecting them to the $\boldsymbol{\Theta}$ that gives the *highest* peak at the estimated fraction, $\eta_{\widehat{q}}$. The dimensions of the smallest rectangular region around $\boldsymbol{\alpha}$ in the projected space enclosing $n_{\widehat{q}}$ points, divided by two, gives the estimate of the scale *in k dimensions* which forms the diagonal of $\mathbf{S}$. The corresponding points form an initial 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 choosing elemental subsets from this set, thus making it very efficient.

We generate N elemental subset-based model hypothesis. The value of N is specific to the problem and will be given in Section 4. Given a hypothesis $[\boldsymbol{\Theta}, \boldsymbol{\alpha}]$, we first compute the k-dimensional projections $\widetilde{\mathbf{z}}_i = \boldsymbol{\Theta}^\top \widetilde{\mathbf{x}}_i$, $i = 1, \ldots, n$. The value of $\widetilde{c}$, such that $\widetilde{\mathbf{z}}_i = \mathbf{z}_i^{[c]}$, is determined using (12). Since the kernel $K(u)$ is radial-symmetric, we can write it using the profile $\kappa(u^2) = K(u)$. The multivariate kernel density function based on the k-dimensional projections $\widetilde{\mathbf{z}}_i$ is then

$$f_{\boldsymbol{\Theta}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\kappa((\mathbf{z} - \widetilde{\mathbf{z}}_i)^\top \widetilde{\mathbf{B}}_i^{-1} (\mathbf{z} - \widetilde{\mathbf{z}}_i))}{\sqrt{\det \widetilde{\mathbf{B}}_i}}. \tag{17}$$

The original robust estimation problem of (13) can be reformulated into the problem of maximizing the kernel density in k dimensions. Given a $\boldsymbol{\Theta}$, the estimated intercept $\boldsymbol{\alpha}_0$ corresponds to the location of the mode of the kernel density closest to $\boldsymbol{\alpha}$. This mode is usually also the highest mode of the kernel density $f_{\boldsymbol{\Theta}}(\mathbf{z})$:

$$\boldsymbol{\alpha}_0 = \arg\max_{\mathbf{z}} f_{\boldsymbol{\Theta}}(\mathbf{z}). \tag{18}$$

The final estimate of the intercept $\widehat{\boldsymbol{\alpha}}$ corresponds to the location of the highest mode over all N randomly chosen hypotheses, while the corresponding matrix $\widehat{\boldsymbol{\Theta}}$ is the estimate of $\boldsymbol{\Theta}$:

$$[\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{\alpha}}] = \arg\max_{\boldsymbol{\Theta}} \left(\arg\max_{\mathbf{z}} f_{\boldsymbol{\Theta}}(\mathbf{z}) \right). \tag{19}$$

As compared to this, RANSAC just uses the $\boldsymbol{\alpha}$ obtained from the elemental subset as an estimate of the intercept. This might be inaccurate, especially when the data contains asymmetric inlier noise.

Given a hypothesis $[\boldsymbol{\Theta}, \boldsymbol{\alpha}]$, the maximization problem (18) is solved by initializing mean-shift procedure from $\boldsymbol{\alpha}$. Taking the derivative of (17) w.r.t. $\mathbf{z}$:

$$\nabla f_{\boldsymbol{\Theta}}(\mathbf{z}) = \frac{2}{n} \sum_{i=1}^{n} \widetilde{\mathbf{B}}_i^{-1} \Delta \mathbf{z}_i \frac{g(\Delta \mathbf{z}_i^{\top} \widetilde{\mathbf{B}}_i^{-1} \Delta \mathbf{z}_i)}{\sqrt{\det \widetilde{\mathbf{B}}_i}} = 0, \quad (20)$$

where $\Delta \mathbf{z}_i = \mathbf{z} - \widetilde{\mathbf{z}}_i$ and $g(u^2) = -\kappa'(u^2)$. The mean shift vector can be written as

$$\delta \mathbf{z} = \left[\sum_{i=1}^{n} \frac{\widetilde{\mathbf{B}}_i^{-1} g(\ldots)}{\sqrt{\det \widetilde{\mathbf{B}}_i}} \right]^{-1} \left[\sum_{i=1}^{n} \frac{\widetilde{\mathbf{B}}_i^{-1} \widetilde{\mathbf{z}}_i g(\ldots)}{\sqrt{\det \widetilde{\mathbf{B}}_i}} \right] - \mathbf{z}. \quad (21)$$

Note that the bandwidth matrix $\widetilde{\mathbf{B}}_i$ is different for each point, making the problem heteroscedastic. The iteration $\widetilde{\mathbf{z}}^{(j+1)} = \delta \mathbf{z}^{(j)} + \widetilde{\mathbf{z}}^{(j)}$ is an ascent step converging to the closest mode, $\boldsymbol{\alpha}_0$.

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. This stopping criterion is merely a heuristic, but worked well for *all* the applications presented in this paper.

3.3   Local Optimization on Grassmann Manifolds

In general, $\boldsymbol{\Theta}$ is an $m \times k$ orthonormal matrix, which can be represented as a point on the Grassmann manifold, $\mathbf{G}_{m,k}$. The function (17) can then be optimized on $\mathbf{G}_{m,k} \times \mathbb{R}^k$ simultaneously with respect to $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$, which results in a better model estimate.

For optimization over Grassmann manifolds, the conjugate gradient method can be used [8]. In order to compute the derivatives of $f_{\boldsymbol{\Theta}}(\mathbf{z})$ w.r.t. $\boldsymbol{\Theta}$ and $\boldsymbol{\alpha}$, we assume that the covariance matrices $\widetilde{\mathbf{H}}_i = \boldsymbol{\Theta}^{\top} \widetilde{\mathbf{C}}_i \boldsymbol{\Theta}$ are independent of $\boldsymbol{\Theta}$. In practice, this assumption does not change the results significantly while saving a lot of computation time. Moreover, the optimization is performed only for those hypotheses for which the density obtained by maximizing (17) using mean-shift is greater than 0.9 times the highest density obtained over all previous hypotheses. This algorithm was used previously for pbM in [29], where the optimization function (17) was assumed to be independent of the scale matrix $\mathbf{S}_{\boldsymbol{\Theta}}$. This worked for the homoscedastic applications but is quite unrealistic in general.

Here, the estimated scale matrix is not dependent on $\boldsymbol{\Theta}$. The theory of Grassmann manifolds and the conjugate gradient algorithm for optimization on $\mathbf{G}_{m,k} \times \mathbb{R}^k$, along with several experiments for single-carrier problems, can be found in [22]. In this paper, we will evaluate this algorithm on a multicarrier application in Section 4.3.1.

3.4   Step Three: Inlier/Outlier Dichotomy

Once the model estimation is performed, the final step is to separate the inliers from the outliers. This separation is based

on the assumption that each inlier structure is corrupted with generally different, asymmetric and unimodal additive noise. Given the model estimate $[\widehat{\boldsymbol{\Theta}}, \widehat{\boldsymbol{\alpha}}]$, let $\widetilde{\mathbf{z}}_i$ (with bandwidth matrices $\widetilde{\mathbf{B}}_i$) be the projections of the data points onto $\widehat{\boldsymbol{\Theta}}$. Starting mean-shift iterations from *every point* $\widetilde{\mathbf{z}}_i$, $i = 1, \ldots, n$, the points for which the procedure converges at $\widehat{\boldsymbol{\alpha}}$ (with a small tolerance) are considered as inliers. 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.

Points lying close to the boundary of the basin of attraction should be dichotomized very carefully. Sometimes, even a small error in the estimation of model parameters could lead to misclassifications around the boundary, especially for multivariate problems. One way of solving the problem is to use additional information that can be reliably extracted after *all* models are estimated. For example, in multiple projective 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 4.2.2.

3.5   Discriminative Reclassification

In data containing multiple inlier structures, there are often points that lie at the intersection of two or more subspaces. Due to the iterative nature of random hypotheses-based algorithms, such points get associated with the structure that is detected first. This could lead to misclassification of points. To overcome this problem, we perform a reclassification of the detected inliers after *all* the inlier structures are estimated.

Since the unstructured outliers were already removed, we only have the estimated models and their respective set of inlier points. For each inlier point in every structure, we compute the kernel density for all the estimated models. The kernel densities are between 0 and 1 and a point is then assigned to the structure that gives the maximum density. This can improve the overall classification accuracy. Some examples are presented in Section 4.

4   EXPERIMENTS

We present three groups of experiments. First, we show the performance of our algorithm on both single and multicarrier synthetic examples. Then, we present two single-carrier real-world applications—face image clustering and projective motion segmentation. Finally, we show two multicarrier applications—camera calibration and multiple homography estimation. Each application is formulated as a robust subspace estimation problem (4). The corresponding values for length of the carrier vector m, the dimensionality of the null space k, and the number of carriers ζ, are provided for each example. The multiplicative ambiguity in the estimation problem (4) is resolved by using the constraint $\boldsymbol{\Theta}^{\top} \boldsymbol{\Theta} = \mathbf{I}_{k \times k}$.

4.1   Synthetic Examples

4.1.1   Conic Lines ($m = 3, k = 2, \zeta = 1$)

A line in 3D can be represented as an intersection of two planes. The problem of line estimation is homoscedastic since the variables and carriers are identical. We consider
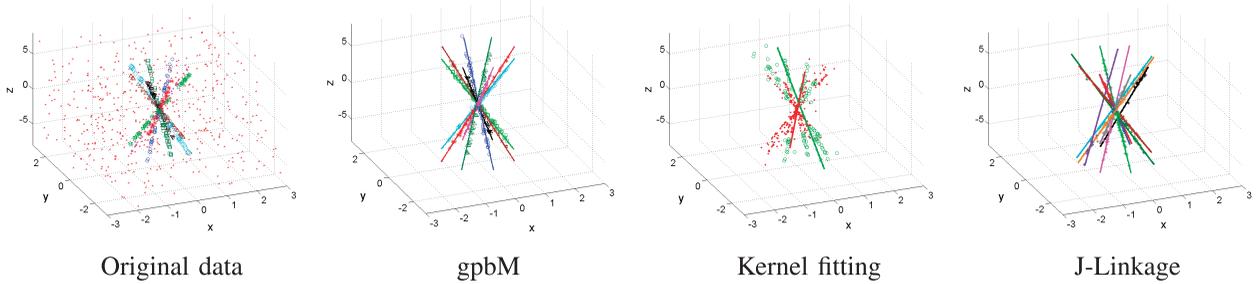
Fig. 4. Eight conic lines. There are 50 inliers point in every line and 500 unstructured outliers. See text. The inliers are independently corrupted with Gaussian noise with $\sigma = 0.02$. Only gpbM is able to recover all eight lines in practice.

eight different lines in 3D lying on surface of a double cone with its vertex at $(0, 0, 0)$ and axis aligned with z-axis. The angle between each consecutive pair of lines is about $7.3°$. Along each line, we generated 50 equally spaced inlier points with z-coordinate between $[-6, 6]$. The $x, y,$ and z coordinates of the inliers were independently corrupted with zero mean Gaussian noise with standard deviation of 0.02. In addition, 500 unstructured 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.

For each inlier structure, in the scale estimation $M = 1,000$ and in the model estimation $N = 200$ were used. The results were compared with RANSAC [12], the Kernel Fitting (KF) method [4], and the J-Linkage method [31].

We performed 100 runs of each algorithm. The KF and J-Linkage methods 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 did it detect seven lines instead of eight. For gpbM, the error in the estimation of Θ 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. Fig. 4 shows a comparison of the sample results obtained. RANSAC was able to find all eight structures only if the right parameters were given and is not shown in the figure. Note that the comparison with RANSAC and J-Linkage is not fair since both these methods 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.

4.1.2 Three Intersecting Lines $(m = 2, k = 1, \zeta = 1)$

We generated three inlier structures, each along a different line in 2D having 100, 200, and 300 points. The problem is homoscedastic. The x and y-coordinates of the inliers were independently corrupted with zero-mean Gaussian noise. The corresponding standard deviations of the noise were 0.1σ, 0.05σ, and 0.12σ, where σ was varied uniformly between $[0, 1.2]$ in steps of 0.15. Any other such setting for number of points and inlier noise yields similar results. The larger the value of σ, the more was the overlap between the three structures. In addition, 400 random outliers were also added between $[-1, -1]$ and $[3, 3]$, resulting in a total of 1,000 data points.

We compare of our algorithm with RANSAC [12] and kernel fitting method [4]. For each value of σ, the results of all the algorithms were averaged over 50 runs. RANSAC was always provided the *true scales* of the inlier noise. For gpbM, $M = 1,000$ and $N = 200$ were used for all values of σ.

The results of the experiments are summarized in Fig. 5. For $\sigma = 1.2$, Figs. 5b and 5c show the sample results obtained using gpbM, before and after the discriminative reclassification step. In Fig. 5d, the average number of misclassification errors (out of 1,000 points) for various values of σ are reported for each method. The gpbM algorithm outperforms the other methods for all values of the inlier noise even without the discriminative reclassification step, which further improves the results.

4.1.3 Ellipse Fitting $(m = 3, k = 1, \zeta = 2)$

We also tested our algorithm on the heteroscedastic, ellipse fitting problem of Fig. 2. In addition to the 100 noisy inlier points ($\sigma = 0.0707$), 500 uniformly distributed outliers were added between $[-2.5, -2.5, -2]$ and $[2.5, 2.5, 12]$, resulting in a total of 600 data points. We compared the performance of gpbM algorithm with that of RANSAC over 100 runs of both the algorithms. The scale provided to RANSAC was
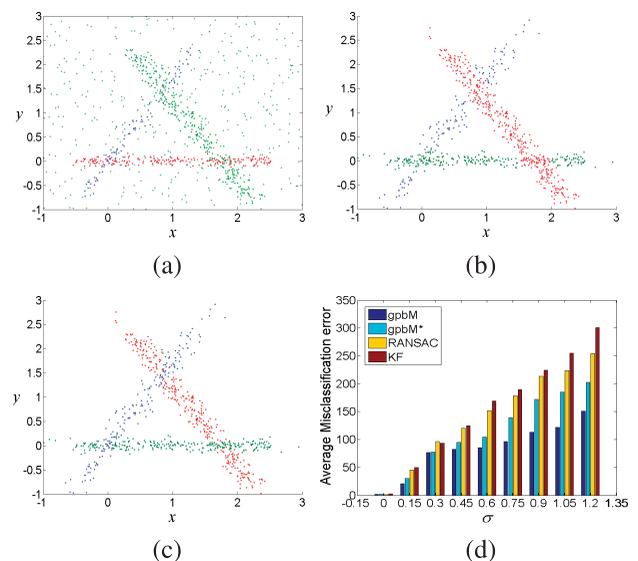

Fig. 5. Three intersecting lines. (a) Sample input data for $\sigma = 1.2$. (b) Detection results using the gpbM algorithm before discriminative reclassification. (c) Detection results after discriminative reclassification with 24 points corrected. (d) Number of average misclassification errors over 50 runs computed for different values of σ. The results of gpbM without discriminative reclassification are reported under gpbM*.
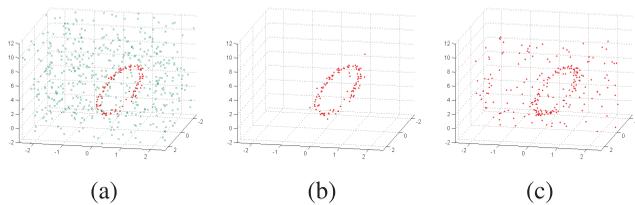
Fig. 6. Ellipse estimation in 3D. (a) Sample input data. (b) Detection results using gpbM. (c) Detection results using RANSAC.

computed as half of norm of homoscedastic scales, $\widehat{s}_{1hom}$ and $\widehat{s}_{2hom}$. (See Section 1.) Using the values of $M = 500$ and $N = 100$, on average, the gpbM algorithm misclassified 19.3 points, as opposed to RANSAC which misclassified 72.5 points. Fig. 6 shows sample results for both algorithms.

4.2 Real Single Carrier Applications

4.2.1 Face Image Clustering ($m = 20, k = 18, \zeta = 1$)

Clustering face images under varying illumination is a difficult problem. It has been proven that the set of images of the same object taken under varying illumination conditions forms a polyhedral cone in the image space which can be approximated by a low-dimensional linear subspace [15]. We test our algorithm on data containing 500 images—50 frontal face images of each of the *10 subjects* of Yale Face Database B. The number of subspaces (number of subjects) is *not known* a priori.

Clustering of faces using the spectral clustering method was first successfully addressed in [18]. We follow this work to compute the symmetric global affinity matrix with nonnegative entries and apply the gpbM algorithm to fit linear subspaces in its low-dimensional representation. The problem is homoscedastic since the variables and carriers are identical.

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}:

$$\mathbf{x}_i = \sum_{j, j \neq i} w_{ij} \mathbf{x}_j, \quad i, j = 1, \ldots, n. \quad (22)$$

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 500×500 symmetric affinity matrix is formed as $\mathbf{A} = (\mathbf{W} + \mathbf{W}^\top)/2$. 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 m largest eigenvalues of $\mathbf{P}$ form a $n \times m$ matrix $\mathbf{Q}$. Images of the same subject taken under varying lighting conditions generally lie in a $d < m$ dimensional subspace [18]. Therefore, the dimensionality of the null space is $k = m - d$.

The clustering algorithms proposed in [2] and [36] used frontal images of three out of the total 10 subjects for evaluation of their techniques. However, in all their experiments, they used uncropped face images where the presence of substantial background makes the problem relatively easier. For example, Fig. 7 shows the face data of
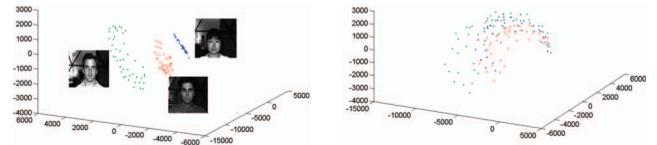

Fig. 7. Frontal faces of three subjects of Yale Database B projected to 3D. *Left*: Face images with background as used in [36] and [2]. *Right*: Same images projected to 3D after cropping faces.

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.

We evaluate our algorithm over 500 cropped frontal face images of *all the 10* subjects. Each image was then down-sampled to 64×64 size for faster computation. Fig. 8 shows three examples for each of the 10 subjects. The $\mathbf{Q}$ matrix was formed by using first $m = 20$ significant eigenvectors. With $M = 5,000$ and $N = 500$, we used the gpbM algorithm to fit 2D subspaces ($k = 18$). Each subspace corresponds to one subject in this 20D space. For 100 runs, the average and median errors over 500 images were 3.42 and 3.4 percent.

Similar performance was also achieved on data containing images of three to nine subjects (50 per subject), with the value of m increasing from 10 to 18. In all experiments, the performance was slightly worse for $d = 3$. For $d = 2$ and $m = 20$, we tested the method of [2] on data containing the same 500 cropped images and it gave an error of 61.4 percent.

4.2.2 Projective Motion Factorization
$(m = 15, k = 12, \zeta = 1)$

We use point trajectories across multiple frames to segment *multiple* rigid body motions in a video sequence. Several approaches have been proposed which can be categorized into factorization-based [32], [34], clustering-based [9], [19], [11], robust estimation-based [12], [26], [3], [29], algebraic [36], and statistical methods [30], [16]. A brief review of most of these techniques can be found in [9]. Except for [3] and [29], all other methods assume that the number of motions is known a priori. We compare the performance of our algorithm with other state-of-the-art robust subspace estimation methods on the *Hopkins155* dataset. We also show the performance on the *parking lot* sequence that also contains unstructured outliers.

Assume that n_1 rigidly moving inlier points lying on a *single motion* are tracked over all F frames. The $2F$ image coordinates are used to define a feature vector for each point in $\mathbb{R}^{2F}$. In general, these n_1 vectors lie in a 4D subspace of $\mathbb{R}^{2F}$ [32]. In all the experiments, we use $F = 5$ frames. For the *Hopkins155* dataset, this corresponds to selecting every sixth or seventh frame for each sequence.

In homogeneous coordinates, the ith noisy, inlier image point in the jth frame, $\mathbf{q}_i^j$ and its corresponding noisy 3D world point $\mathbf{Q}_i$ are related as

Fig. 8. Example frontal face images of Yale Database B used in our experiments.

$$\lambda_i^j \mathbf{q}_i^j \approx \mathbf{P}^j \mathbf{Q}_i, \quad i = 1, \ldots, n_1, \quad j = 1, \ldots, F, \qquad (23)$$

where λ_i^j is the projective depth of $\mathbf{q}_i^j$ and $\mathbf{P}^j$ is the 3×4 camera matrix for jth frame. A homogeneous image 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 $\mathbf{y}_i = [(x_i^1, y_i^1, \lambda_i^1), \ldots (x_i^F, y_i^F, \lambda_i^F)]^\top$. The corresponding carrier vector is given by $\mathbf{x}_i = [(\lambda_i^1 x_i^1, \lambda_i^1 y_i^1, \lambda_i^1) \ldots (\lambda_i^F x_i^F, \lambda_i^F y_i^F, \lambda_i^F)]^\top$. In this case both $\mathbf{y}_i$ and $\mathbf{x}_i$ lie in $\mathbb{R}^{3F}$. In the absence of any prior knowledge, we assume that the noise corrupting the depth and the image coordinates is identical and known only up to a common scale σ^2, which is a coarse approximation. 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

$$\mathbf{J}_{\mathbf{x}_i|\mathbf{y}_i} = \begin{bmatrix} \mathcal{J}_i^1 & 0 & 0 & \ldots & 0 \\ 0 & \mathcal{J}_i^2 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \mathcal{J}_i^F \end{bmatrix}, \mathcal{J}_i^j = \begin{bmatrix} \lambda_i^j & 0 & 0 \\ 0 & \lambda_i^j & 0 \\ x_i^j & y_i^j & 1 \end{bmatrix}. \tag{24}$$

For data containing multiple, nondegenerate motions and unstructured outliers, the unknown motion subspaces are estimated from the $3F \times n$ carrier data matrix in two steps. In the *first* step, assuming all the unknown depths to be equal to one $(\lambda_i^j = 1)$, an iterative affine motion estimation is performed using gpbM. Since the variables and carriers are identical, this step is homoscedastic with $m = 2F = 10$. The size of the elemental subset is four and due to centering of the elemental subset data for hypotheses generation, the dimensionality of the null space is $k = 2F - 3 = 7$. This step returns the initial estimates of *all* the motions and their associated 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 first estimate the unknown projective depths. The carrier vector is heteroscedastic due to the multiplication of the image points with their depths. The data corresponding to each inlier structure obtained using affine motion estimation can be factorized as

$$\mathbf{T} = \begin{bmatrix} \lambda_1^1 \mathbf{q}_1^1 & \lambda_2^1 \mathbf{q}_2^1 & \ldots & \lambda_{n_1}^1 \mathbf{q}_{n_1}^1 \\ \lambda_1^2 \mathbf{q}_1^2 & \lambda_2^2 \mathbf{q}_2^2 & \ldots & \lambda_{n_1}^2 \mathbf{q}_{n_1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^F \mathbf{q}_1^F & \lambda_2^F \mathbf{q}_2^F & \ldots & \lambda_{n_1}^F \mathbf{q}_{n_1}^F \end{bmatrix} = \mathcal{M}\mathcal{S}, \tag{25}$$

where $\mathcal{M}$ is the $3F \times 4$ motion matrix and $\mathcal{S}$ is the $4 \times n_1$ structure matrix. The unknown depths λ_i^j are estimated using the iterative method of [34]. The algorithm starts with all the depth values equal to one. The rank-four approximation $\check{\mathbf{T}}$ of $\mathbf{T}$ is computed using SVD. The least-squares estimates of the depths are then obtained from $\check{\mathbf{t}}_i^j$, the entries of $\check{\mathbf{T}}$ corresponding to $\lambda_i^j \mathbf{q}_i^j$ as

$$\lambda_i^j = \left(\check{\mathbf{t}}_i^j \right)^\top \mathbf{q}_i^j / \|\mathbf{q}_i^j\|^2, \tag{26}$$

where $\mathbf{q}_i^j$ is the original image point and λ_i^j, $\check{\mathbf{t}}_i^j$ change in each iteration. The iterations for estimating $\check{\mathbf{T}}$ and λ_i^j end when $\check{\mathbf{T}}$ is within a small tolerance ($< 10^{-6}$) of $\mathbf{T}$. The method usually converges in very few iterations. After obtaining the depths of all inlier points and rejecting the

outliers, the $3F \times n$ carrier data matrix is also updated. Similarly to the first step, an iterative projective motion estimation is then performed using gpbM with the new values of $m = 3F = 15$ and $k = 3F - 3 = 12$.

At the end of both affine and projective steps, we use fundamental matrices to dichotomize the points lying *close* to the boundary of the basin of attraction [37]. Only the inlier points that lie within half a scale margin of the boundary of basin of attraction in *all* dimensions are considered. 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 (5) computed over all pairs of frames is *minimum*. For this simple classification problem, the eight point algorithm for estimating fundamental matrices is sufficient because the data contains mostly inliers and we just want to classify the point to its closest motion.

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 all these examples, the values both M and N were conservatively set to 500. This is the only application where we use Gaussian kernel instead of Epanechnikov for density estimation because it gives slightly improved performance.

The *Hopkins155* dataset is available online at http://www.vision.jhu.edu/data/hopkins155. It consists of 120 two-motion and 35 three-motion sequences which are divided into three categories—traffic, articulated, and checkerboard. We compared the performance of our algorithm with five other methods—Generalized PCA [36], RANSAC, Local Subspace Affinity (LSA) [40], pbM [29], and the Ordered Residual Kernel method [3]. The classification error is computed similarly to [3] and [36]:

$$\text{classification error} = \frac{\text{number of mislabeled points}}{\text{total number of points}}. \tag{27}$$

As opposed to pbM, ORK, and gpbM, the methods GPCA, RANSAC, and LSA 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.

Tables 1 and 2 compare the results obtained by various methods on two and three-motion sequences. The results of REF, the reference/control method generated for benchmarking, GPCA, LSA, and RANSAC were obtained from [35]. We used our own implementation of pbM. The results of ORK [3] were not reported for individual categories. For each sequence, the results for gpbM were averaged over 100 runs, while those of pbM were averaged over 20 runs. Additionally, we obtained a median error of 5.6 percent for two-motion and 6.2 percent for three-motion sequences. Results can be further improved by handling degeneracies present in the data.

The *parking lot* sequence contains four motions (background and three moving cars). The points across various frames were matched using [14]. 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

TABLE 1
Percent Classification Errors for Two-Motion Sequences

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**

Only pbM, ORK, and gpbM are user independent.

TABLE 2
Percent Classification Errors for Three-Motion Sequences

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**

Only pbM, ORK, and gpbM are user independent.

62 unstructured outliers. Fig. 9 shows sample motion segmentation results using gpbM along with the corresponding confusion matrix.

4.3 Real Multicarrier Applications

For both the applications presented in this section, we assume that the noise is only present in the *measured* image points. No explicit assumptions for the model of the carrier noise are made.

4.3.1 Camera Calibration $(m = 12, k = 1, \zeta = 2)$

Camera calibration is the task of estimating the 3×4 camera projection matrix $\mathbf{P}$ that relates the noisy 3D scene points $\mathbf{Q}_i$ to their corresponding noisy 2D image points $\mathbf{q}_i$ as

$$\mathbf{q}_i \approx \mathbf{P}\mathbf{Q}_i, \quad i = 1, \ldots, n_1, \tag{28}$$

where $\mathbf{q}_i$ and $\mathbf{Q}_i$ are in homogeneous coordinates. Note that the projection matrix is same as $\boldsymbol{\Theta}$, but written as $\mathbf{P}$ following the convention.

Given n point correspondences (inliers and outliers), let $[x_i \; y_i]^\top$ be the *measured* noisy image points and $[X_i \; Y_i \; Z_i]^\top$ be the corresponding *measured* world points for $i = 1, \ldots, n$. The vector of variables is given by $\mathbf{y}_i = [x_i \; y_i \; X_i \; Y_i \; Z_i]^\top$. Writing $\mathbf{Q}_i = [X_i \; Y_i \; Z_i \; 1]^\top$, the robust estimation problem is to find the projection matrix $\mathbf{P} = [\mathbf{p}_1 \; \mathbf{p}_2 \; \mathbf{p}_3]^\top$ such that, for each $i = 1, \ldots, n_1$,
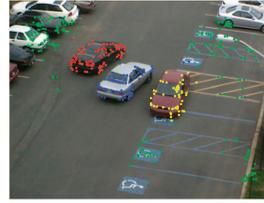
$$\mathbf{A}_i \mathbf{p} = \begin{bmatrix} -\mathbf{Q}_i^\top & \mathbf{0}_4^\top & x_i \mathbf{Q}_i^\top \\ \mathbf{0}_4^\top & -\mathbf{Q}_i^\top & y_i \mathbf{Q}_i^\top \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} \approx \mathbf{0}_2. \tag{29}$$

For each point, $\mathbf{A}_i$ is the 2×12 carrier matrix where the two rows of $\mathbf{A}_i$ are the two carrier vectors $\mathbf{x}_i^{[1]}$ and $\mathbf{x}_i^{[2]}$ which are
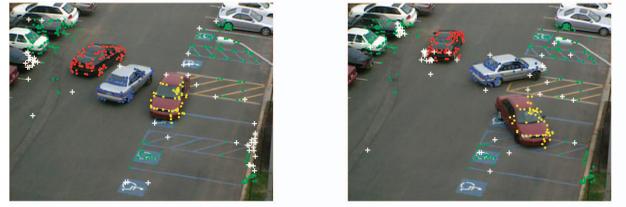
(a)   (b)

(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)

Fig. 9. 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.

heteroscedastic due to multiplicative terms. Therefore, $m = 12$, $k = 1$, and $\zeta = 2$.

Given six 2D-3D point correspondences, the 12×12 data matrix is formed by stacking the corresponding six carrier matrices. The hypothesis for the camera matrix is then computed by using the Direct Linear Transformation (DLT) algorithm [17, Algorithm 7.1]. Since our data also contains outliers, we only compute the initial linear solution here. While sophisticated techniques for detecting degenerate configurations for camera calibration exist, in our case, checking if this data matrix is rank-11 was sufficient.

We assume that the noise is only present in the x and y-coordinates of the 2D image points and the 3D points are noiseless. The 5×5 covariance matrices of the variables $\mathbf{y}_i$ are defined as

$$\mathbf{C}_{\mathbf{y}_i} = \sigma^2 \begin{bmatrix} \mathbf{I}_{2\times2} & \mathbf{0}_{2\times3} \\ \mathbf{0}_{3\times2} & \mathbf{0}_{3\times3} \end{bmatrix}, \quad i = 1, \ldots, n. \tag{30}$$

For a point correspondence between $\mathbf{q}_i$ and $\mathbf{Q}_i$, the 5×12 Jacobians for the two carriers are given by

$$\mathbf{J}_{\mathbf{x}_i^{[1]}|\mathbf{y}_i} = \begin{bmatrix} \mathbf{0}_3^\top & & \mathbf{Q}_i^\top \\ \mathbf{0}_3^\top & \mathbf{0}_{5\times5} & \mathbf{0}_3^\top & 0 \\ -\mathbf{I}_{3\times3} & x_i \mathbf{I}_{3\times3} & \mathbf{0}_3 \end{bmatrix},$$

$$\mathbf{J}_{\mathbf{x}_i^{[2]}|\mathbf{y}_i} = \begin{bmatrix} \mathbf{0}_3^\top & \mathbf{0}_3^\top & 0 \\ \mathbf{0}_{5\times4} & \mathbf{0}_3^\top & \mathbf{0}_5 & \mathbf{Q}_i^\top \\ -\mathbf{I}_{3\times3} & y_i \mathbf{I}_{3\times3} & \mathbf{0}_3 \end{bmatrix}. \tag{31}$$

The corresponding 12×12 covariance matrices are then computed as

$$\mathbf{C}_i^{[c]} = \mathbf{J}_{\mathbf{x}_i^{[c]}|\mathbf{y}_i}^\top \mathbf{C}_{\mathbf{y}_i} \mathbf{J}_{\mathbf{x}_i^{[c]}|\mathbf{y}_i}, \; c = 1, 2.$$

We use a Merton College image obtained from http://www.robots.ox.ac.uk/~vgg/data/data-mview.html for

evaluation of our algorithm. The same image is also used for homography estimation in Section 4.3.2 (Fig. 11a-top). Here, there are 575 inlier 2D-3D point correspondences available. Since the 3D points in the scene have comparable distances from the camera, both 2D and 3D points were normalized using the isotropic scaling as described in [17, p. 180].

We present three sets of experiments. Given an estimated projection matrix $\widehat{\mathbf{P}}$, we use the average geometric error of the 575 true inlier points $\mathbf{q}_i$ and their computed reprojections $\widehat{\mathbf{q}}_i = \widehat{\mathbf{P}}\mathbf{Q}_i$ as the metric for comparing various methods. While increasing either the amount of inlier noise or the number of outliers in gpbM, the value of M was varied uniformly between 500 to 2,000 and that of N between 100 to 200. The results of each algorithm were averaged over 50 runs.

In the *first* set, uniformly distributed outlier point correspondences were added and their number was varied from 0 to 1,500 in steps of 100. In the *second* experiment, we added zero-mean Gaussian noise independently both in x and y-coordinates of the image points. The standard deviation of this noise, σ_{2D} was varied between one to four pixels in steps of 0.4 pixels. In both the experiments, we compared our algorithm with RANSAC [12] and the kernel fitting method [4]. For each inlier structure, RANSAC had to be provided the true scale of the inlier noise. For every setting, the two algorithms used $M + N$ random hypotheses. All the algorithms were very sensitive to noise added to 3D points, so no noise in 3D points was added. The results of these two experiments are summarized in Figs. 10a and 10b. In both the experiments, gpbM outperformed the other two methods.

In the *third* experiment, we tested the gpbM algorithm by varying both the number of outliers and the scale of the inlier noise (Fig. 10c). We also tested gpbM by additionally performing model optimization over Grassmann manifolds (Fig. 10d). Details of this step can be found in [22, Section 4.2.1]. It is clear that this step further improves that the performance of gpbM. The gpbM algorithm started to break down when number of outliers approached 1,500 (about three times the number of inliers) even when 20,000 hypotheses were used.

4.3.2 Multiple Planar Homography Estimation
$(m = 9, k = 1, \zeta = 2)$

A planar homography is a general 2D transformation between two projective planes. Let $\mathbf{q}_i$ and $\mathbf{q}'_i$ be a corresponding pair of homogeneous, noisy inlier points. Given a 3×3 homography matrix $\mathbf{H}$ between the two planes, $\mathbf{q}_i$ can be mapped to $\mathbf{q}'_i$ using the relation

$$\mathbf{q}'_i \approx \mathbf{H}\mathbf{q}_i, \quad i = 1, \ldots, n_1. \tag{32}$$

Note that the homography matrix is same as $\boldsymbol{\Theta}$, but written as $\mathbf{H}$ following the convention. Given n noisy point correspondences (inliers and outliers), let $[x_i \; y_i]^\top$ and $[x'_i \; y'_i]^\top$, $i = 1, \ldots, n$, be the *measured* noisy points in the two images. Each point correspondence gives a vector of variables $\mathbf{y}_i = [x_i \; y_i \; x'_i \; y'_i]^\top$, $i = 1, \ldots, n$. Writing $\mathbf{q}_i = [x_i \; y_i \; 1]^\top$, the robust estimation problem is to find the homography matrix $\mathbf{H} = [\mathbf{h}_1 \mathbf{h}_2 \mathbf{h}_3]^\top$ such that, for each $i = 1, \ldots, n_1$,
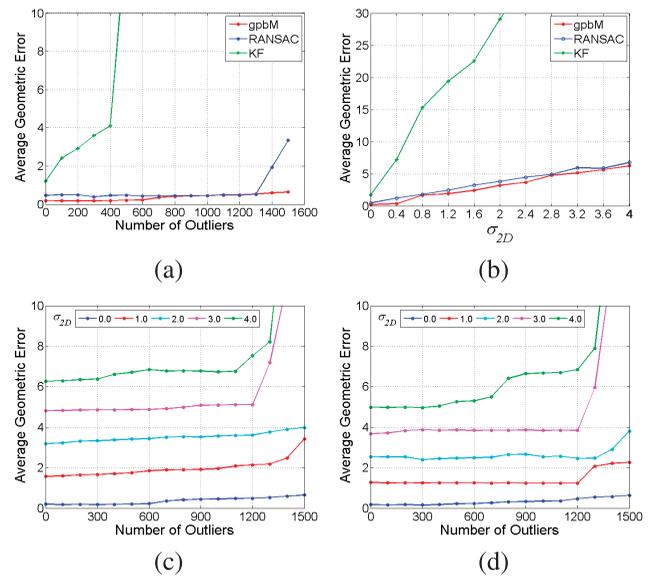

Fig. 10. Camera calibration experiments performed on Fig. 11a-top. (a) Average geometric error computed by varying the number of outliers with zero inlier noise. (b) Average geometric error computed by varying the inlier noise when no outliers were added. Note that the comparisons with RANSAC are not fair because it needs true parameters as input. (c) Average geometric error computed for gpbM when both, the inlier noise and the number of outliers were varied. (d) Performance of gpbM when Grassmann manifold optimization was added to the model estimation step.

$$\mathbf{A}_i\mathbf{h} = \begin{bmatrix} -\mathbf{q}_i^\top & \mathbf{0}_3^\top & x'_i\mathbf{q}_i^\top \\ \mathbf{0}_3^\top & -\mathbf{q}_i^\top & y'_i\mathbf{q}_i^\top \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} \approx \mathbf{0}_2. \tag{33}$$

For each point correspondence, $\mathbf{A}_i$ is the 2×9 carrier matrix. The rows of $\mathbf{A}_i$ correspond to the two carrier vectors $\mathbf{x}_i^{[1]}$ and $\mathbf{x}_i^{[2]}$ obtained from each point correspondence and are heteroscedastic due to multiplicative terms. Therefore, $m = 9$, $k = 1$, and $\zeta = 2$.

Given four point correspondences across two planes, the 8×9 data matrix is formed by stacking the corresponding four carrier matrices. For a point correspondence pair $\mathbf{q}_i$ and $\mathbf{q}'_i$, since the noise is only present in the x and y coordinates, the 4×4 covariance matrix of the variable $\mathbf{y}_i$ is given as $\mathbf{C}_{\mathbf{y}_i} = \sigma^2\mathbf{I}_{4 \times 4}$, $i = 1, \ldots, n$. The 4×9 Jacobians of the two carriers are given by

$$\mathbf{J}_{\mathbf{x}_i^{[1]}|\mathbf{y}_i} = \begin{bmatrix} -\mathbf{I}_{2 \times 2} & & x'_i\mathbf{I}_{2 \times 2} & \mathbf{0}_2 \\ \mathbf{0}_2^\top & \mathbf{0}_{4 \times 4} & \mathbf{q}_i^\top \\ \mathbf{0}_2^\top & & \mathbf{0}_2^\top & 0 \end{bmatrix},$$

$$\mathbf{J}_{\mathbf{x}_i^{[2]}|\mathbf{y}_i} = \begin{bmatrix} & -\mathbf{I}_{2 \times 2} & y'_i\mathbf{I}_{2 \times 2} & \mathbf{0}_2 \\ \mathbf{0}_{4 \times 3} & \mathbf{0}_2^\top & \mathbf{0}_4 & \mathbf{0}_2^\top & 0 \\ & \mathbf{0}_2^\top & & \mathbf{q}_i^\top \end{bmatrix}. \tag{34}$$

The corresponding 9×9 covariance matrices are computed as $\mathbf{C}_i^{[c]} = \sigma^2 \, \mathbf{J}_{\mathbf{x}_i^{[c]}|\mathbf{y}_i}^\top \, \mathbf{C}_{\mathbf{y}_i} \mathbf{J}_{\mathbf{x}_i^{[c]}|\mathbf{y}_i}$, $c = 1, 2$.

We perform experiments on two pairs of Merton College images obtained from http://www.robots.ox.ac.uk/~vgg/data/data-mview.html. The number of inlier point correspondences in the two pairs was 361 and 429, which were distributed among four significant planes. The number of homographies in each pair of images was not known a priori.
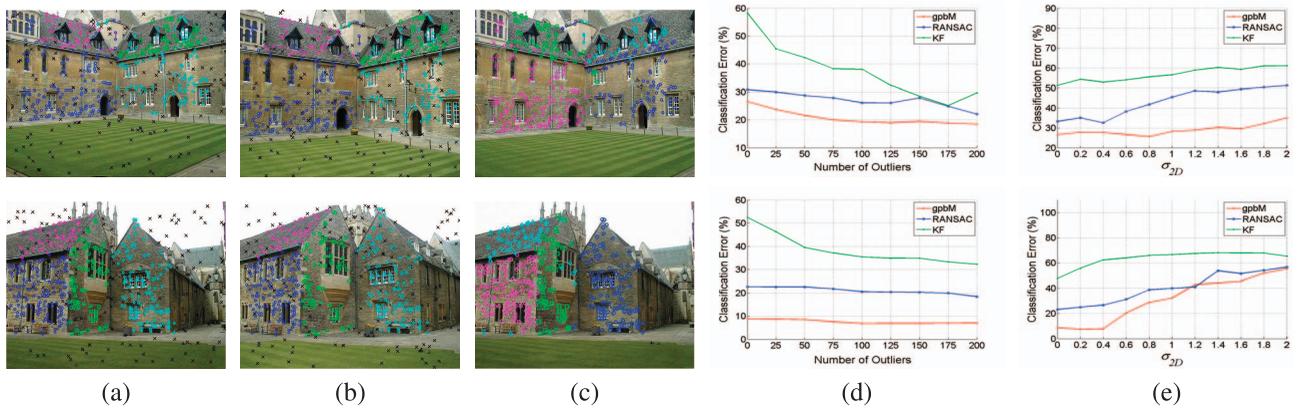
Fig. 11. Multiple homography estimation. (a) and (b) Original left and right images. Each image pair contains inliers corresponding to four planar homographies and 150 random outliers. (c) Results of the gpbM algorithm after discriminative reclassification of inliers. (d) Classification error computed by varying the number of outliers with zero inlier noise. (e) Classification error computed by varying the inlier noise when no outliers were added. Note that the comparisons with RANSAC are not fair because it needs true parameters as input.

In all these experiments, the points in the two images were normalized using isotropic scaling [17, p. 107].

For each image pair two sets of experiments were performed. In the *first* experiment, uniformly distributed, random outlier point correspondences were added and their number was varied between 0 to 200 in steps of 25. Beyond 200 outliers, it became very difficult for the algorithm to estimate homographies of all the planes. In the *second* experiment, we added zero-mean Gaussian noise independently in x and y coordinates of the inliers in both the images. The standard deviation of the noise σ_{2D} was varied between 0 to 2 pixels in steps of 0.2. We compared the results of our algorithm with RANSAC [12] and the kernel fitting method [4]. RANSAC always needed the true scale of the inlier noise and the number of inlier structures. The results of all the experiments are summarized in Fig. 11. For each inlier structure, for gpbM, the value of M varied uniformly between 2,000 to 8,000 for increasing number of outliers or the amount of inlier noise. The value of N varied between 400 to 1,000. The other two methods used $M + N$ iterations. The results of each algorithm were averaged over 50 runs.

Since these examples have multiple inlier structures, computing the error in point classification similar to (27) is more informative. It penalizes inaccuracies in model estimation, inlier/outlier dichotomy and discriminating one inlier structure from the others. The gpbM algorithm outperformed both RANSAC and KF for all settings of inlier noise and number of outliers.

5   DISCUSSION

The gpbM algorithm can handle multivariate constraints for single and multicarrier problems in a unified framework. The scale of the inlier noise is estimated automatically for each structure. We have shown its performance on numerous datasets covering a wide variety of applications in computer vision.

The execution time of the algorithm depends on various factors like number of outliers, dimensionality of the problem (m, k), amount of inlier noise, and number of inlier structures. Since each of the first two stages of the gpbM algorithm evaluates multiple elemental subset-based

hypotheses, those can be implemented in parallel. The execution time can thus be decreased by using multiple cores. As an example, consider the problem of affine factorization ($m = 10$, $k = 7$, and $\zeta = 1$) with data containing a few hundred points with one inlier structure and approximately equal number of inliers and outliers. A C implementation of the algorithm takes about 0.33, 0.09, and 0.05 seconds for the first, second, and third steps, respectively (without optimization on Grassmann manifolds), when executed on an Intel quad-core machine with 8 Gb of memory. If the optimization is also performed, then the execution time of the second step increases to 0.18 seconds. Also, since the hypotheses in the second step are generated from the initial set of inliers obtained in the first step of the algorithm, typically, the value of N is much smaller than M. Because the computation time per hypothesis in the second step is more than that of the first step, this leads to an overall increase in the speed of the algorithm.

We have also highlighted the importance of estimating models under heteroscedasticity, which allows us to account for the nonlinearities in the data. The resulting algorithm is theoretically more sound and also led to improvements in the actual performance of the computer vision systems. In the future, the algorithm can be extended into various directions, e.g., detecting degeneracies in some or all the inlier structures [13], optimizing hypothesis generation using guided sampling [6], [24], and performing regression in non-euclidean spaces [29].

Even after these enhancements, there will still be some problems which probably cannot be solved by using only bottom-up information. In the absence of any knowledge about the number of structures or the scale of the inlier noise, it is hard to decide if two very close lying inlier structures should be merged into one or not. Similarly, points lying at the intersection or boundary of two inlier structures cannot be dichotomized perfectly unless we know something about the underlying process that generates them. Without any information about the scale of the inlier structures, successfully detecting 20-30 inlier structures in an image, a realistic assumption for human vision, cannot be achieved with current estimation methods that use only bottom-up information. We believe that bringing automatic estimation algorithms for computer vision to human vision-like

performance would require a human vision-like optimal mix of bottom-up and top-down information.

The code for our algorithm is written in MATLAB and C and can be downloaded from http://coewww.rutgers.edu/riul/research/code/GPBM/index.html.

REFERENCES

[1] A. Bab-Hadiashar and D. Suter, "Robust Segmentation of Visual Data Using Ranked Unbiased Scale Estimate," *Robotica,* vol. 17, no. 6, pp. 649-660, 1999.

[2] G. Chen and G. Lerman, "Spectral Curvature Clustering (SCC)," *Int'l J. Computer Vision,* vol. 81, no. 3, pp. 317-330, 2009.

[3] T.J. Chin, H. Wang, and D. Suter, "The Ordered Residual Kernel for Robust Motion Subspace Clustering," *Proc. Advances in Neural Information Processing Systems Conf.,* pp. 333-341, 2009.

[4] T.J. Chin, H. Wang, and D. Suter, "Robust Fitting of Multiple Structures: The Statistical Learning Approach," *Proc. 12th IEEE Int'l Conf. Computer Vision,* pp. 413-420, 2009.

[5] J. Choi and G.G. Medioni, "StaRSaC: Stable Random Sample Consensus for Parameter Estimation," *Proc. IEEE Conf. Computer Vision,* pp. 675-682, 2009.

[6] O. Chum and J. Matas, "Matching with PROSAC—Progressive Sample Consensus," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 220-226, 2005.

[7] O. Chum, J. Matas, and J. Kittler, "Locally Optimized RANSAC," *Proc. DAGM Symp. Pattern Recognition,* pp. 236-243, 2003.

[8] A. Edelman, T.A. Arias, and S.T. Smith, "The Geometry of Algorithms with Orthogonality Constraints," *SIAM J. Matrix Analysis and Application,* vol. 20, no. 2, pp. 303-353, 1998.

[9] E. Elhamifar and R. Vidal, "Sparse Subspace Clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2790-2797, 2009.

[10] L. Fan and T. Pylvänäinen, "Robust Scale Estimation from Ensemble Inlier Sets for Random Sample Consensus Methods," *Proc. 10th European Conf. Computer Vision,* pp. 182-195, 2008.

[11] P. Favaro, R. Vidal, and A. Ravichandran, "A Closed Form Solution to Robust Subspace Estimation and Clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1801-1807, 2011.

[12] 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,* vol. 24, no. 6, pp. 381-395, 1981.

[13] J.M. Frahm and M. Pollefeys, "RANSAC for (Quasi-)Degenerate Data (QDEGSAC)," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 453-460, 2006.

[14] B. Georgescu and P. Meer, "Point Matching under Large Image Deformations and Illumination Changes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 6, pp. 674-689, June 2004.

[15] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 643-660, June 2001.

[16] A. Gruber and Y. Weiss, "Multibody Factorization with Uncertainty and Missing Data Using the EM Algorithm," *Proc. IEEE Computer Vision and Pattern Recognition Conf.,* vol. 1, pp. 707-714, 2004.

[17] R.I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision,* second ed. Cambridge Univ. Press, 2004.

[18] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering Appearances of Objects under Varying Illumination Conditions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 11-18, 2003.

[19] F. Lauer and C. Schnorr, "Spectral Clustering of Linear Subspaces for Motion Segmentation," *Proc. 12th IEEE Int'l Conf. Computer Vision,* pp. 678-685, 2009.

[20] K.-M. Lee, P. Meer, and R.-H. Park, "Robust Adaptive Segmentation of Range Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 2, pp. 200-205, Feb. 1998.

[21] B. Matei and P. Meer, "Estimation of Nonlinear Errors-in-Variables Models for Computer Vision Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 10, pp. 1537-1552, Oct. 2006.

[22] S. Mittal and P. Meer, "Conjugate Gradient on Grassmann Manifolds for Robust Subspace Estimation," *J. Image and Vision Computing,* vol. 30, nos. 6-7, pp. 417-427, June 2012.

[23] R. Raguram and J.-M. Frahm, "RECON: Scale-Adaptive Robust Estimation via Residual Consensus," *Proc. 13th IEEE Int'l Conf. Computer Vision,* pp. 1299-1306, 2011.

[24] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus," *Proc. 10th European Conf. Computer Vision,* pp. 500-513, 2008.

[25] R. Raguram, J.-M. Frahm, and M. Pollefeys, "Exploiting Uncertainty in Random Sample Consensus," *Proc. 12th IEEE Int'l Conf. Computer Vision,* pp. 2074-2081, 2009.

[26] 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," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2008.

[27] S. Mittal, S. Anand, and P. Meer, "Generalized Projection Based M-Estimator: Theory and Applications," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2689-2696, 2011.

[28] R. Subbarao and P. Meer, "Beyond RANSAC: User Independent Robust Regression," *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshop,* June 2006.

[29] R. Subbarao and P. Meer, "Subspace Estimation Using Projection Based M-Estimators over Grassmann Manifolds," *Proc. European Conf. Computer Vision,* vol. 1, pp. 301-312, 2006.

[30] Y. Sugaya and K. Kanatani, "Geometric Structure of Degeneracy for Multi-Body Motion Segmentation," *Proc. Second Workshop Statistical Methods in Video Processing,* pp. 13-25, 2004.

[31] R. Toldo and A. Fusiello, "Robust Multiple Structures Estimation with J-Linkage," *Proc. 10th European Conf. Computer Vision,* pp. 537-547, 2008.

[32] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography: A Factorization Method," *Int'l J. Computer Vision,* vol. 9, no. 2, pp. 137-154, 1992.

[33] P.H.S. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," *Computer Vision and Image Understanding,* vol. 78, pp. 138-156, 2000.

[34] B. Triggs, "Factorization Methods for Projective Structure and Motion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 845-851, 1996.

[35] R. Tron and R. Vidal, "A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2007.

[36] R. Vidal, Y. Ma, and S. Sastry, "Generalized Principal Component Analysis (GPCA)," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 12, pp. 1-15, 2005.

[37] R. Vidal, Y. Ma, S. Soatto, and S. Sastry, "Two-View Multibody Structure from Motion," *Int'l J. Computer Vision,* vol. 68, no. 1, pp. 7-25, 2006.

[38] H. Wang, D. Mirota, and D.H. Gregory, "A Generalized Kernel Consensus-Based Robust Estimator," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 1, pp. 178-184, Jan. 2010.

[39] H. Wang and D. Suter, "Robust Fitting by Adaptive-scale Residual Consensus," *Proc. Eighth European Conf. Computer Vision,* vol. 3, pp. 107-118, 2004.

[40] J. Yan and M. Pollefeys, "A General Framework for Motion Segmentation: Independent, Articulated, Rigid, Non-Rigid, Degenerate and Nondegenerate," *Proc. Ninth European Conf. Computer Vision,* pp. 94-106, 2006.

Sushil Mittal received the BTech degree in electronics and communication engineering from the National Institute of Technology, Warangal, India, in 2004, and the MS and PhD degrees in electrical and computer engineering from Rutgers University, New Jersey, in 2008 and 2011, respectively. From 2004 to 2005, he was in the Department of Electrical Engineering at the Indian Institute of Science as a research associate. Currently, he is working in the Department of Statistics at Columbia University as a postdoctoral research scientist. His research interests include computer vision, machine learning, and applied statistics. He is a member of the IEEE.

Saket Anand received the BE degree in electronics engineering from the University of Pune, India, and the MS degree in electrical and computer engineering from Rutgers University, New Jersey, in 2003 and 2006, respectively. From 2007 to 2009, he worked in handwriting recognition as a research engineer at Read-Ink Technologies, Bangalore, India. Currently, he is working toward the PhD degree in electrical and computer engineering at Rutgers University, New Jersey. His research interests include computer vision, statistical pattern recognition, and machine learning applied to computer vision. He is a student member of the IEEE.

Peter Meer received the Dipl Engn degree from the Bucharest Polytechnic Institute, Romania, and the DSc degree from the Technion-Israel Institute of Technology, Haifa, Israel, both in electrical engineering, in 1971 and 1986, respectively. From 1971 to 1979, he was with the Computer Research Institute, Cluj, Romania, working on R&D of digital hardware. Between 1986 and 1990, he was an assistant research scientist at the Center for Automation Research, University of Maryland at College Park. In 1991, he joined the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, New Jersey and is currently a professor. He has held visiting appointments in Japan, Korea, Sweden, Israel, and France, and was on the organizing committees of numerous international workshops and conferences. He was an associate editor of the *IEEE Transaction on Pattern Analysis and Machine Intelligence* between 1998 and 2002, a member of the editorial board of *Pattern Recognition* between 1990 and 2005, and was a guest editor of *Computer Vision and Image Understanding* for a special issue on robustness in computer vision in 2000. His research interests include application of modern statistical methods to image understanding problems. He was coauthor of an award winning paper in *Pattern Recognition* in 1989, the best student paper in the 1999, the best paper in the 2000, and the runner-up paper in 2007 of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). With coauthors Dorin Comaniciu and Visvanathan Ramesh he received the Longuet-Higgins prize for fundamental contributions in computer vision in the past 10 years at CVPR in 2010. He is a fellow of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.