



Conjugate gradient on Grassmann manifolds for robust subspace estimation[☆]

Sushil Mittal^{*}, Peter Meer

ECE Department, Rutgers University, Piscataway, NJ 08854-8058, USA

ARTICLE INFO

Article history:

Received 12 April 2011

Received in revised form 18 July 2011

Accepted 14 September 2011

Keywords:

Grassmann manifolds

Conjugate gradient algorithm

Generalized projection based M-estimator

ABSTRACT

Most geometric computer vision problems involve orthogonality constraints. An important subclass of these problems is subspace estimation, which can be equivalently formulated into an optimization problem on Grassmann manifolds. In this paper, we propose to use the conjugate gradient algorithm on Grassmann manifolds for robust subspace estimation in conjunction with the recently introduced generalized projection based M-Estimator (gpbM). The gpbM method is an elemental subset-based robust estimation algorithm that can process heteroscedastic data without any user intervention. We show that by optimizing the orthogonal parameter matrix on Grassmann manifolds, the performance of the gpbM algorithm improves significantly. Results on synthetic and real data are presented.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Orthogonality constraints arise frequently in geometric computer vision problems. Linear subspace estimation naturally falls into this category. Orthogonal matrices representing linear subspaces of Euclidean space can be represented as points on Grassmann manifolds. Studying the geometric properties of Grassmann manifolds can therefore prove very useful in solving many vision problems. In the recent past, the problem of subspace estimation has been formulated in many different ways. Important methods include robust regression based approaches [34, 22], spectral clustering based approaches [6, 41] and clustering random hypothesis on Grassmann manifolds [35, 2].

Robust regression has been an active field of research in computer vision. It corresponds to estimating multiple, noisy inlier structures present in the data corrupted with gross outliers. Following RANdom SAmple Consensus (RANSAC) [7], many algorithms like MLESAC, LO-RANSAC, PROSAC, QDEGSAC, have been proposed. See [27] for a brief description of these methods. Recently, the projection based M-estimator (pbM) of [34] was extended to the generalized pbM (gpbM) [22]. The main advantage of pbM and gpbM over RANSAC and RANSAC-like regression algorithms is that both pbM and gpbM do not require from the user an estimate of the scale of the noise corrupting inlier points. While pbM uses a MAD-based scale estimate that is dependent on the choice of a particular hypothesis, gpbM estimates the true scale of the noise beforehand.

In this paper, we extend the work of [22] to robustly estimate subspaces by using concepts of Riemannian geometry. While the main idea of work follows that of [34], it is different from [34] in a number

of ways. Like [34], we also refine the estimate of the subspace obtained from the gpbM algorithm by using conjugate gradient on Grassmann manifolds. But, in pbM [34], firstly, the scale of the inlier noise was estimated using a variant of the Median Absolute Deviation (MAD) based method. Being dependent on MAD-based scale estimate, the method is 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. Secondly, in [34], the scale of inlier noise being dependent on the choice of a particular model hypothesis, explicit assumptions had to be made for optimizing the objective function on Grassmann manifolds. In gpbM, since the scale is computed beforehand, it is independent of a particular model hypothesis and no such assumptions are required. Finally, in [34], only homoscedastic formulation of the method was presented. Our method can handle both homoscedastic and heteroscedastic data. We argue that by refining the subspace estimation using Grassmann manifolds, the performance of the original gpbM algorithm improves significantly. Experiments on challenging synthetic and real examples are presented.

The paper is organized as follows. In Section 2, we describe the necessary theoretical concepts related to Riemannian manifolds. We also review some of the recent work that uses of Riemannian manifolds for solving problems in computer vision. In Section 3, we describe, in detail, the properties of Grassmann manifolds and derive the necessary relations. In Section 4, we briefly explain the gpbM algorithm with a detailed focus on the conjugate gradient for Grassmann manifolds. In Section 5, we present results from experiments conducted on synthetic and real examples.

2. Riemannian manifolds

A Hausdorff space is defined as a topological space where for any two distinct points \mathbf{x} , \mathbf{y} , there exist neighborhoods \mathcal{U} around

[☆] This paper has been recommended for acceptance by special issue Guest Editor Rama Chellappa.

^{*} Corresponding author. Tel.: +1 732 447 7703.

E-mail addresses: smittal@caip.rutgers.edu (S. Mittal), meer@jove.rutgers.edu (P. Meer).

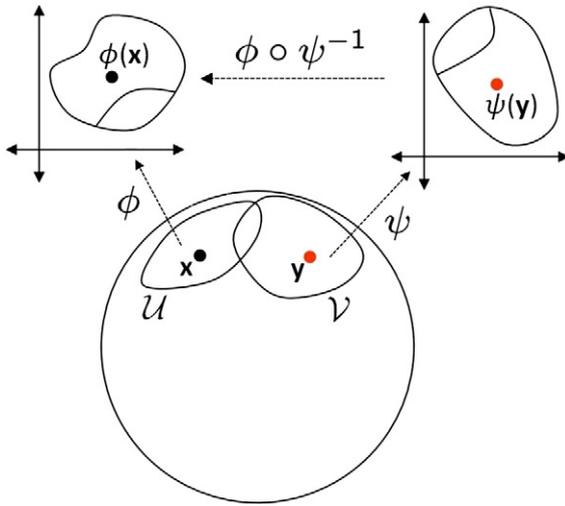


Fig. 1. Example of a two-dimensional manifold. Two overlapping coordinate charts are shown. If the manifold is analytic, the transition map $\phi \circ \psi^{-1}$ (and $\psi \circ \phi^{-1}$) from \mathbb{R}^2 to \mathbb{R}^2 should be analytic.

x and \mathcal{V} around y such that \mathcal{U} and \mathcal{V} are disjoint. A manifold \mathcal{M} is an d -dimensional Hausdorff topological space, that is locally homeomorphic to Euclidean space, i.e., for every point $x \in \mathcal{M}$ there exists a neighborhood $\mathcal{U} \subset \mathcal{M}$ containing x and an associated mapping ϕ from \mathcal{U} to some Euclidean space \mathbb{R}^d , such that $\phi(\mathcal{U})$ is an open set in \mathbb{R}^d . The neighborhood \mathcal{U} and its associated mapping ϕ together form a coordinate chart (\mathcal{U}, ϕ) .

Given two coordinate charts (\mathcal{U}, ϕ) and (\mathcal{V}, ψ) such that $\mathcal{U} \cap \mathcal{V}$ is non-empty, the transition map $\phi \circ \psi^{-1}$ is defined as a mapping from the open set $\psi(\mathcal{U} \cap \mathcal{V}) \in \mathbb{R}^d$ to the open set $\phi(\mathcal{U} \cap \mathcal{V}) \in \mathbb{R}^d$ (Fig. 1). The properties of calculus, like smoothness and differentiability, can be applied to the transition maps of a manifold.

An analytic manifold is a smooth, differentiable manifold such that for all coordinate charts (\mathcal{U}, ϕ) and (\mathcal{V}, ψ) , either $\mathcal{U} \cap \mathcal{V}$ is empty or $\mathcal{U} \cap \mathcal{V}$ is nonempty and the transition map $\phi \circ \psi^{-1}$ is analytic, i.e., has a convergent Taylor series expansion.

Consider a real valued function $f : \mathcal{M} \rightarrow \mathbb{R}$ on the manifold. Given a coordinate chart (\mathcal{U}, ϕ) , the function $\tilde{f} = f \circ \phi^{-1}$ maps the open set $\phi(\mathcal{U}) \in \mathbb{R}^d$ to \mathbb{R} . The function f is said to be continuous, if for all coordinate charts \tilde{f} is continuous. Similarly, f is said to be analytic if \tilde{f} is analytic for all coordinate charts defined on \mathcal{M} . The set of all continuous, real-valued functions on \mathcal{M} is denoted by $\mathfrak{F}(\mathcal{M})$.

Given a point x on the manifold \mathcal{M} , a tangent vector at x to \mathcal{M} is a real-valued function $\Delta : \mathfrak{F}(\mathcal{M}) \rightarrow \mathbb{R}$ that satisfies the properties of linearity and Leibniz product rule of derivatives. Let $\Delta_x(\mathcal{M})$ represent the vector space of all tangent vectors to \mathcal{M} at x . It can be shown that for d -dimensional manifolds, the tangent space is a d -dimensional vector space ([24], p.8). For a point x lying on \mathcal{M} , $\Delta(f)$ is the magnitude of the derivative of f in the tangent direction Δ at x . Intuitively, the tangent vector can be thought of as velocity of a point constrained to move on the manifold. The tangent space can be further divided into complementary spaces called the horizontal and vertical spaces. The space normal to the tangent space is called the normal space. See Fig. 2a.

The partial differentiation of f on \mathcal{M} can be achieved by first moving the function f back to the Euclidean space using the coordinate charts and then taking the usual directional derivative. Therefore, given a coordinate chart (\mathcal{U}, ϕ) , the partial differential of f at x can be written as

$$\partial_i(f) = \left. \frac{\partial(f \circ \phi^{-1})}{\partial u^i} \right|_{\phi(x)} \tag{1}$$

where, u^i is the i -th coordinate of the point $\phi(x)$ in \mathbb{R}^d . The mutually orthogonal tangent directions $\partial_i, i = 1, \dots, d$ form the basis of $\Delta_x(\mathcal{M})$.

A Riemannian manifold is a real analytic manifold \mathcal{M} of dimension d , which can be isometrically embedded into some Euclidean space \mathbb{R}^m where $m > d$. The tangent space at every point x on a Riemannian manifold is associated with an inner product g_x called the Riemannian metric ([24], p. 55). The manifold is denoted by the ordered pair (\mathcal{M}, g) . Two different inner product metrics on the same manifold would lead to two different Riemannian manifolds. However, in practice there exists a standard metric and the Riemannian manifold is denoted by the underlying analytic manifold \mathcal{M} .

Given two tangent vectors $\Delta, \Lambda \in \Delta_x(\mathcal{M})$, their inner product is written as $g_x(\Delta, \Lambda)$. This enables us to define angles between the two tangent vectors. For a tangent vector Δ the Riemannian metric is the length of Δ and given by $\sqrt{g_x(\Delta, \Delta)}$. Given a set of basis vectors for the tangent space $\Delta_x(\mathcal{M})$, g_x can always be represented as a symmetric positive definite matrix.

A curve in a manifold \mathcal{M} is a smooth mapping α from an open interval \mathbf{T} of \mathbb{R} to \mathcal{M} . For a particular $t \in \mathbf{T}$, $\alpha(t)$ lies on the manifold and $\alpha'(t)$ is a tangent vector at $\alpha(t)$. Given a function f on the manifold, the tangent vector $\alpha'(t)$ applied to f gives $\partial(f \circ \alpha)/\partial t$, i.e., the directional derivative of f in the direction $\alpha'(t)$. The vector $\alpha'(t)$ is the rate of change of α at $t \in \mathbf{T}$. The length of the curve is given by

$$\int_{t \in \mathbf{T}} \sqrt{g_{\alpha(t)}(\alpha'(t), \alpha'(t))} dt. \tag{2}$$

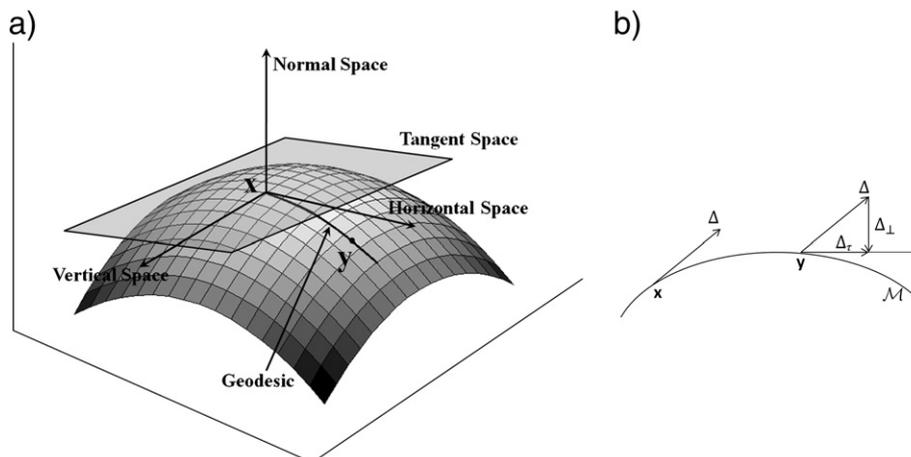


Fig. 2. (a) An illustration of tangent and normal spaces of Riemannian manifold. The geodesic is always in the direction of the horizontal space. (b) Parallel transport of a tangent vector Δ from a point x to another point y on the manifold.

Given points \mathbf{x}, \mathbf{y} on \mathcal{M} , the shortest curve connecting \mathbf{x} and \mathbf{y} is called the *minimal geodesic*. In this work, since we are always only interested in the minimal geodesics, for convenience, we call them just *geodesics*. The length of the geodesic is defined to be the *Riemannian distance* between the two points. Geodesics have the property that $g_{\alpha(t)}(\alpha'(t), \alpha'(t))$ is constant for all $t \in \mathbf{T}$, i.e., the speed is constant along the geodesic ([24], p. 67).

In Euclidean space, a vector can be moved parallel to itself by just moving the base of the arrow. While transporting a tangent vector Δ from a point \mathbf{x} to \mathbf{y} on the manifold \mathcal{M} , the normal component Δ_{\perp} at \mathbf{y} is subtracted from the transported vector. This is called *parallel translation* and is illustrated in Fig. 2b.

Riemannian manifolds appear frequently in computer vision due to the geometric nature of the problems. Different types of Riemannian manifolds like Lie groups, essential manifolds, symmetric manifolds and Grassmann manifolds have been used in the past. Following, we present an overview of frequently used Riemannian manifolds in computer vision and the related papers that have appeared in the past few years.

A Lie group is a group which is also a manifold such that the group operations are compatible with the smooth structure of the manifold. Examples include special orthogonal, special Euclidean and affine groups. In [40], the mean shift-based clustering algorithm was extended to cluster points lying on matrix Lie groups and applications on special Euclidean group and special orthogonal groups were presented. In [23], a method was developed to compute means and averages on the special orthogonal groups. Using this method, in [8], the algorithm for principal component analysis (PCA) was extended to similarity transformations and special orthogonal groups by defining principal geodesics which was applied for shape analysis of anatomical objects. By defining the averages on special orthogonal and special Euclidean groups, a method for estimating 3D motion was proposed in [10]. In [38], a learning based method was developed for performing regression on points lying on the affine group and an application of object detection and tracking was presented. Visual tracking using particle filtering on affine groups was studied in [14], while [16] proposed learning on affine groups to analyze visual flows in image sequences.

An essential manifold is a six-dimensional space of essential matrices and it encodes the epipolar geometry for a pair of calibrated cameras. The essential manifold was first introduced in [30]. Using its geometry, three different algorithms for rigid motion estimation were proposed and their performance was compared. It was shown in [31] that the essential manifold can be identified as the unit tangent bundle of the special orthogonal group. Based on the work of [20], Gauss-Newton type methods for essential manifolds were proposed in [12]. In [33], the non-linear mean shift algorithm of [40] was extended to cluster points lying on the essential manifold by developing a robust parametrization that bijectively maps an essential matrix to a point on the manifold.

The set of all $m \times m$ symmetric positive definite matrices forms the symmetric manifold. Geometric properties, like the mean of points and geodesics on symmetric manifolds, were introduced in [25]. These properties were then used to perform operations like interpolation, filtering and diffusion on symmetric manifolds. A computationally more efficient geometric mean called the Log-Euclidean mean was proposed in [1]. This mean was used in [15] to perform learning on symmetric manifold for visual tracking of objects. In [39], a method for human detection was introduced by performing learning on the space of covariance descriptors. This idea was further extended in [36] to perform a classification of video sequences into multiple classes for surveillance.

The set of $m \times k$, ($m > k$) orthonormal matrices representing the k -dimensional linear subspaces in \mathbb{R}^m form a Grassmann manifold. Geometric properties of Grassmann manifolds like geodesics, parallel translation are discussed in [5]. In [32], a method for tracking

subspaces using a Bayesian approach was presented. Projective reconstruction of feature points from their multiple, low-dimension projections using Grassmann tensors was proposed in [11]. In [3], an algorithm for finding approximate nearest neighbors on Grassmann manifolds was proposed to analyze human activity in videos. For the same application, the method of [19] performed classification on product manifold of Grassmann manifolds. A non-linear mean shift algorithm to intrinsically cluster points on Grassmann manifolds was proposed in [2], while the method of [37] used Karcher mean [13] to study inferences on Grassmann manifolds. In [17], the conjugate gradient algorithm was applied on Grassmann manifolds for the application of face recognition.

A general algorithm for nonlinear mean shift was proposed in [35], which could handle points lying on any Riemannian manifold. Further, it was also proved that the employed mean-shift method converges to the closest mode of the distribution. The geometric properties of Lie groups, Grassmann manifolds, essential manifolds and the manifolds of symmetric positive definite matrices were discussed in detail and many applications were also presented. For more details, see [35] and references therein.

While the concepts of geodesics and parallel translation are applicable to all Riemannian manifolds, specific implementations for Grassmann manifolds will be presented in Section 3. Note that points lying on manifolds are, in general, represented by small bold letters. However, since points lying on Grassmann manifolds are matrices, they are represented using the standard convention with capital bold letters.

In Section 5, we present two real computer vision applications that are formulated into the problem of robust subspace estimation. Note that in such applications, our goal is to only estimate that appropriate subspace spanned by the inlier points. The choice of a particular basis vectors for that subspace is not important. Since orthogonal matrices representing linear subspace with non-unique basis are also points on Grassmann manifolds, robust subspace estimation can be naturally formulated as the problem of optimizing an appropriate objective function on Grassmann manifolds. As opposed to this, since points on Stiefel manifolds represent orthogonal matrices with fixed set of basis, they are usually of lesser interest for the problems of subspace estimation in computer vision.

3. Grassmann manifolds

A point \mathbf{X} on the Grassmann manifold, $\mathbf{G}_{m,k}$, represents a k -dimensional linear subspace in \mathbb{R}^m , where $m > k$. Therefore, \mathbf{X} is represented by an $m \times k$ orthogonal matrix, i.e., $\mathbf{X}^T \mathbf{X} = \mathbf{I}_{k \times k}$. The point \mathbf{X} is independent of the choice of any particular basis vectors. Points on the Grassmann manifold are *equivalence classes* of $m \times k$ orthogonal matrices, where two matrices are equivalent if their columns span the same k -dimensional subspace in \mathbb{R}^m [5]. Due to the frequent occurrence of orthogonality constraints in many geometric computer vision problems, they can be solved by performing computations on the Grassmann manifolds.

As opposed to Grassmann manifold, in Stiefel manifold, each point is an $m \times k$ orthogonal matrix with a *fixed set* of basis vectors. While algorithms for computations on Stiefel manifolds were described in [2, 37], their performance on real applications in computer vision were only shown for $k = 1$. Since, the choice of basis vector is automatically fixed in that case, the utility of Stiefel manifolds for computer vision problems is still largely unexplored.

To develop the notions of the geodesics and parallel translation of tangent vectors on Grassmann manifolds, it is important to understand their quotient form representation. Note that tangents at points on $\mathbf{G}_{m,k}$ are also represented as $m \times k$ matrices. However, due to the convention, we still refer to them as vectors.

Let \mathbf{Q} be an $m \times m$ orthogonal matrix, represented as a point on the *special orthogonal group*, $\mathbf{SO}(m)$. Let the geodesic curve between the

two points $\mathbf{Q}_0, \mathbf{Q}_1 \in \mathbf{SO}(m)$ be given by $\mathbf{Q}(t)$ such that $\mathbf{Q}_0 = \mathbf{Q}(0)$ and $\mathbf{Q}_1 = \mathbf{Q}(t_1)$ where $0 \leq t \leq t_1, t \in \mathbb{R}$. For any point $\mathbf{Q} \in \mathbf{SO}(m)$, we know that

$$\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_{m \times m}. \quad (3)$$

Differentiating above equation yields $\mathbf{Q}^\top \mathbf{\Gamma}_\mathbf{Q} + \mathbf{\Gamma}_\mathbf{Q}^\top \mathbf{Q} = \mathbf{0}$, where $\mathbf{\Gamma}_\mathbf{Q}$ is the $m \times m$ matrix of differentials of \mathbf{Q} ([4], p. 14). Intuitively, $\mathbf{\Gamma}_\mathbf{Q}$ is the set of tangent vectors at \mathbf{Q} . The normal space $\mathbf{N}_\mathbf{Q}$ at \mathbf{Q} is the orthogonal complement of the tangent space (Fig. 2a), such that $\text{trace}(\mathbf{\Gamma}_\mathbf{Q}^\top \mathbf{N}_\mathbf{Q}) = 0$. It can be shown that $\mathbf{N}_\mathbf{Q} = \mathbf{Q}\mathbf{P}$ where \mathbf{P} is any $m \times m$ symmetric matrix. Then we will have the product of a skew-symmetric matrix $\mathbf{\Gamma}_\mathbf{Q}^\top \mathbf{Q}$ and a symmetric matrix \mathbf{P} , that satisfies the trace being zero. Since \mathbf{Q} is non-singular, orthogonal matrix

$$\mathbf{Q}^\top \mathbf{N}_\mathbf{Q} + \mathbf{N}_\mathbf{Q}^\top \mathbf{Q} = \mathbf{Q}^\top \mathbf{Q}\mathbf{P} + \mathbf{P}^\top \mathbf{Q}^\top \mathbf{Q} = 2\mathbf{P}. \quad (4)$$

Twice differentiating (3) w.r.t. t yields

$$\mathbf{Q}^\top \ddot{\mathbf{Q}} + 2\dot{\mathbf{Q}}^\top \dot{\mathbf{Q}} + \ddot{\mathbf{Q}}^\top \mathbf{Q} = \mathbf{0} \quad (5)$$

where $\dot{\mathbf{Q}}(t)$ and $\ddot{\mathbf{Q}}(t)$ are the instantaneous velocity and acceleration at $\mathbf{Q}(t)$. Since the velocity $\dot{\mathbf{Q}}(t)$, at any time t , is also the first derivative of \mathbf{Q} , $\dot{\mathbf{Q}}(t) \in \mathbf{\Gamma}_{\mathbf{Q}(t)}$ for $0 \leq t \leq t_1$. By the definition of the geodesic, since the speed (norm of the velocity), $\|\dot{\mathbf{Q}}(t)\| = \text{constant}$, for $0 \leq t \leq t_1$, the component of the acceleration, $\ddot{\mathbf{Q}}(t)$ in the tangent space $\mathbf{\Gamma}_{\mathbf{Q}(t)}$ is zero. Therefore, the acceleration vector $\ddot{\mathbf{Q}}(t)$ must lie in the normal space at $\mathbf{Q}(t)$ yielding

$$\ddot{\mathbf{Q}} = \mathbf{Q}(\mathbf{Q}^\top \ddot{\mathbf{Q}} + \ddot{\mathbf{Q}}^\top \mathbf{Q})/2 \quad (6)$$

where $(\mathbf{Q}^\top \ddot{\mathbf{Q}} + \ddot{\mathbf{Q}}^\top \mathbf{Q})$ is a symmetric $m \times m$ matrix. From (5) and (6) it can be shown that

$$\mathbf{Q}^\top \ddot{\mathbf{Q}} + \ddot{\mathbf{Q}}^\top \mathbf{Q} = \mathbf{0}. \quad (7)$$

Let $\mathbf{A} = \mathbf{Q}^\top \dot{\mathbf{Q}}$ be an $m \times m$ matrix. Differentiating \mathbf{A} w.r.t. t , we get $\dot{\mathbf{A}} = \dot{\mathbf{Q}}^\top \dot{\mathbf{Q}} + \mathbf{Q}^\top \ddot{\mathbf{Q}} = \mathbf{0}$, yielding $\mathbf{A} = \text{constant}$. The solution of (7) is of the form

$$\dot{\mathbf{Q}}(t) = \mathbf{Q}(t)\mathbf{A} \quad (8)$$

$$\mathbf{Q}(t) = \mathbf{Q}(0)e^{\mathbf{A}t}. \quad (9)$$

Since the points along the geodesic always lie on the manifold, therefore $\mathbf{Q}(t) \in \mathbf{SO}(m)$ for $0 \leq t \leq t_1$. It follows that $e^{\mathbf{A}t}$ is a rotation matrix and that \mathbf{A} has to be skew-symmetric.

A Grassmann manifold $\mathbf{G}_{m,k}$ can be represented as quotient space within the special orthogonal group $\mathbf{SO}(m)$ using the equivalence classes. Given a point \mathbf{Q} on $\mathbf{SO}(m)$, its equivalence class $[\mathbf{Q}]$, for the Grassmann manifold $\mathbf{G}_{m,k}$, is the set of all $m \times m$ orthogonal matrices whose first k columns span the same subspace. A point on the Grassmann manifold can be represented by the equivalence class

$$[\mathbf{Q}] = \left\{ \mathbf{Q} \begin{bmatrix} \mathbf{Q}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{m-k} \end{bmatrix} : \mathbf{Q}_k \in \mathbf{SO}(k), \mathbf{Q}_{m-k} \in \mathbf{SO}(m-k) \right\}. \quad (10)$$

The tangent space $\mathbf{\Gamma}_\mathbf{Q}$ at a point \mathbf{Q} , can be further divided into linear subspaces called the vertical and horizontal spaces which are orthogonal complements of each other (Fig. 2a). The vertical space consists of the tangent vectors at \mathbf{Q} , movements along which keep the point in the same equivalence class, $[\mathbf{Q}]$. Movements only along the horizontal space at \mathbf{Q} actually move a point on the Grassmann manifold. Formally, vertical space is defined as the set of vectors tangent to the entire equivalence class $[\mathbf{Q}]$.

From Eqs. (8) and (10), the horizontal tangent vectors at \mathbf{Q} are defined as

$$\Delta_\mathbf{Q} = \mathbf{Q}\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{0} & -\mathbf{B}^\top \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \quad (11)$$

where \mathbf{A} is an $m \times m$ skew-symmetric matrix and \mathbf{B} is an arbitrary $(m-k) \times k$ matrix. The dimension of the horizontal space and vertical space being $(m-k)k$ and $m(m-1)/2 - (m-k)k$, the dimension of entire tangent space $\mathbf{\Gamma}_\mathbf{Q}$ is $m(m-1)/2$. The dimension of the normal space $\mathbf{N}_\mathbf{Q}$ is $m^2 - m(m-1)/2 = m(m+1)/2$. Note that the $m \times k$ matrix $\mathbf{X} = \mathbf{Q}\mathbf{I}_{m \times k}$ is the projection of the entire equivalence class $[\mathbf{Q}]$ on $\mathbf{G}_{m,k}$, where $\mathbf{I}_{m \times k} = \begin{bmatrix} \mathbf{I}_{k \times k} \\ \mathbf{0} \end{bmatrix}$. Writing $\mathbf{Q} = [\mathbf{X} \mathbf{X}_\perp]$, where \mathbf{X}_\perp is the orthogonal complement of \mathbf{X} , the horizontal space at \mathbf{X} is the set of vectors of the form

$$\Delta_\mathbf{X} = [\mathbf{X} \mathbf{X}_\perp] \begin{bmatrix} \mathbf{0} & -\mathbf{B}^\top \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \mathbf{I}_{m \times k} = \mathbf{X}_\perp \mathbf{B}. \quad (12)$$

Since the horizontal space is equivalent to the tangent space of the quotient, the tangent space at \mathbf{X} consists of $m \times k$ matrices of the form $\Delta_\mathbf{X}$ which satisfy $\mathbf{X}^\top \Delta_\mathbf{X} = \mathbf{0}$. From (9), the corresponding geodesic for the Grassmann manifold is given by

$$\mathbf{X}(t) = \mathbf{Q}(0)e^{\mathbf{A}t} \mathbf{I}_{m \times k}. \quad (13)$$

Similar to [5], it can be proved that given the initial conditions, $\mathbf{X}(0) = \mathbf{X}$, $\dot{\mathbf{X}}(0) = \mathbf{X}_\perp \mathbf{B} = \mathbf{A}$, the geodesic Eq. (13) can be rewritten as

$$\mathbf{X}(t) = [\mathbf{X} \mathbf{V} \mathbf{U}] \begin{bmatrix} \cos \Sigma t \\ \sin \Sigma t \end{bmatrix} \mathbf{V}^\top \quad (14)$$

where $\mathbf{U}\Sigma\mathbf{V}^\top$ is the compact SVD of \mathbf{A} (only the first k columns of \mathbf{U} are computed) and the operators \sin and \cos act element-by-element along the diagonal of Σ .

Given the tangent space $\Delta_\mathbf{X}$ at a point \mathbf{X} , the normal space at \mathbf{X} consists of the set of matrices $\mathbf{N}_\mathbf{X}$ that satisfy $\text{trace}(\Delta_\mathbf{X}^\top \mathbf{N}_\mathbf{X}) = \mathbf{0}$. Similar to (4), the projection of an arbitrary $m \times k$ matrix \mathbf{Z} on $\mathbf{N}_\mathbf{X}$ is given by

$$\mathbf{Z}_{\mathbf{N}_\mathbf{X}} = \mathbf{X}(\mathbf{X}^\top \mathbf{Z} + \mathbf{Z}^\top \mathbf{X})/2. \quad (15)$$

A tangent vector $\Delta \in \Delta_\mathbf{X}$ at $\mathbf{X} = \mathbf{X}(0)$ can be parallel-translated to another point $\mathbf{Y} \in \mathbf{G}_{m,k}$ by using the technique illustrated in Fig. 2b, i.e., by infinitesimally removing the normal component of the translated vector, Δ_\perp along the path between \mathbf{X} and \mathbf{Y} on the manifold. By differentiating (15), substituting $\mathbf{Z} = \Delta$ and using $\mathbf{X}^\top \Delta + \Delta^\top \mathbf{X} = \mathbf{0}$, we get

$$\Delta_\perp = \dot{\Delta}_\tau = -\mathbf{X}(\dot{\mathbf{X}}^\top \Delta + \Delta^\top \dot{\mathbf{X}})/2. \quad (16)$$

The negative sign in the above equation appears due to the inverted direction of Δ_\perp in Fig. 2. Similar to (12), let the solution of (16) be of the form

$$\Delta_\tau(t) = \mathbf{Q}(t)\mathbf{C}\mathbf{I}_{m \times k} = \mathbf{X}_\perp(t)\mathbf{D} \quad (17)$$

where \mathbf{C} is another skew-symmetric matrix of the form

$$\mathbf{C} = \begin{bmatrix} \mathbf{0} & -\mathbf{D}^\top \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \quad (18)$$

and \mathbf{D} is an arbitrary $(m-k) \times k$ matrix. The parallel translation of the tangent vector (17) along the geodesic $\mathbf{X}(t) = \mathbf{Q}(0)e^{\mathbf{A}t} \mathbf{I}_{m \times k}$ is given by

$$\Delta_\tau(t) = \mathbf{Q}(0)e^{\mathbf{A}t} \mathbf{C}\mathbf{I}_{m \times k}. \quad (19)$$

Similar to [5], the parallel translation of Δ along the geodesic in direction $\dot{\mathbf{X}}(0) = \mathbf{X}_\perp(0)\mathbf{B} = \mathbf{\Lambda}$ can be rewritten as

$$\Delta_\tau(t) = \left(\mathbf{X}\mathbf{V} \mathbf{U} \begin{bmatrix} -\sin\Sigma t \\ \cos\Sigma t \end{bmatrix} \mathbf{U}^\top + [\mathbf{I} - \mathbf{U}\mathbf{U}^\top] \right) \Delta \quad (20)$$

where $\mathbf{U}\Sigma\mathbf{V}^\top$ is the compact SVD of $\mathbf{\Lambda}$. It is easy to observe that $\Delta_\tau(0) = \Delta$.

For Grassmann manifold, tangent vectors in the horizontal space and geodesics on the manifold are closely related. For a tangent vector Δ at \mathbf{X} , there is a unique geodesic curve $\alpha : [0, 1] \rightarrow \mathbf{G}_{m,k}$, starting at \mathbf{X} with initial velocity $\alpha'(0) = \Delta$. The exponential map, $\exp_{\mathbf{X}}$, maps Δ to the point on the manifold reached by this geodesic

$$\exp_{\mathbf{X}}(\Delta) = \alpha(1) = \mathbf{X}(1). \quad (21)$$

where $\mathbf{X}(1)$ is computed using (14). Consequently, the origin of the tangent space is mapped to the point itself, $\exp_{\mathbf{X}}(\mathbf{0}) = \mathbf{X}(0)$. For each point $\mathbf{X} \in \mathbf{G}_{m,k}$, there exists a neighborhood $\tilde{\mathcal{U}}$ of the origin in $\Delta_{\mathbf{X}}$, such that $\exp_{\mathbf{X}}$ is a mapping from $\tilde{\mathcal{U}}$ onto a neighborhood \mathcal{U} of \mathbf{X} , where \mathcal{U} is called the normal neighborhood. Over the neighborhood \mathcal{U} , we can define the inverse of the exponential and this mapping from \mathcal{U} to $\tilde{\mathcal{U}}$ is known as the logarithm map, $\log_{\mathbf{X}} = \exp_{\mathbf{X}}^{-1}$. Given two points $\mathbf{X}, \mathbf{Y} \in \mathbf{G}_{m,k}$, the logarithm map finds a constant, $m \times m$ skew-symmetric matrix \mathbf{A} (and consequently the tangent direction $\mathbf{\Lambda}$), such that the geodesic along $\mathbf{\Lambda}$ and starting at \mathbf{X} , reaches \mathbf{Y} in unit time.

Let $\mathbf{X}^\top \mathbf{Y} = \mathbf{V}\mathbf{C}\mathbf{W}^\top$ and $\mathbf{X}_\perp^\top \mathbf{Y} = \mathbf{U}_1\mathbf{S}\mathbf{W}^\top$ be the generalized SVD such that $\mathbf{C}^\top \mathbf{C} + \mathbf{S}^\top \mathbf{S} = \mathbf{I}_{k \times k}$. Similar to [9], the Cosine-Sine (CS) decomposition of $\mathbf{Q}^\top \mathbf{Y}$ can be written as

$$\mathbf{Q}^\top \mathbf{Y} = \begin{bmatrix} \mathbf{X}^\top \mathbf{Y} \\ \mathbf{X}_\perp^\top \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_1 \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ \mathbf{S} \end{bmatrix} \mathbf{W}^\top \quad (22)$$

Left-multiplying $\mathbf{X}_\perp^\top \mathbf{Y}$ by \mathbf{X}_\perp , we obtain

$$\mathbf{X}_\perp \mathbf{X}_\perp^\top \mathbf{Y} = \mathbf{X}_\perp \mathbf{U}_1 \mathbf{S} \mathbf{W}^\top = \mathbf{U} \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix} \mathbf{W}^\top = \mathbf{U} \mathbf{S}_1 \mathbf{W}^\top \quad (23)$$

where \mathbf{U} is an $m \times m$ orthogonal matrix with same first k columns as $\mathbf{X}_\perp \mathbf{U}_1$. Left multiplying $\mathbf{X}^\top \mathbf{Y}$ by \mathbf{X} and adding to (23) results

$$\mathbf{X}\mathbf{X}^\top \mathbf{Y} + \mathbf{X}_\perp \mathbf{X}_\perp^\top \mathbf{Y} = \mathbf{Y} = \mathbf{X}\mathbf{V}\mathbf{C}\mathbf{W}^\top + \mathbf{U}\mathbf{S}_1\mathbf{W}^\top \quad (24)$$

which is of the same form as (14), the only difference being the post-multiplication with \mathbf{W}^\top above instead of \mathbf{V}^\top in (14). However, since both \mathbf{W}^\top and \mathbf{V}^\top are $k \times k$ orthogonal matrices, multiplication with either of them does not move the point on $\mathbf{G}_{m,k}$ [5]. The logarithm map can then be computed as

$$\log_{\mathbf{X}}(\mathbf{Y}) = \mathbf{\Lambda} = \mathbf{U}\Sigma\mathbf{V}^\top = \mathbf{U} \begin{bmatrix} \sin^{-1} \mathbf{S}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{V}^\top = \mathbf{U} \begin{bmatrix} \cos^{-1} \mathbf{C}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{V}^\top \quad (25)$$

where $\mathbf{C}_1 = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{(m-k) \times (m-k)} \end{bmatrix}$ and \sin^{-1} and \cos^{-1} act element-by-element along the diagonal of \mathbf{S}_1 and \mathbf{C}_1 respectively.

The Grassmann manifold $\mathbf{G}_{m,k}$ can be represented as a submanifold of the Euclidean space by smoothly embedding it in an (mk) -dimensional Euclidean space. The tangent space $\Delta_{\mathbf{X}}$ at a point $\mathbf{X} \in \mathbf{G}_{m,k}$ has k^2 constraints imposed on it due to the condition $\mathbf{X}^\top \Delta_{\mathbf{X}} = \mathbf{0}$. Therefore, the tangent space $\Delta_{\mathbf{X}}$ has dimension $d = (m - k)k$. Given two tangent vectors $\Delta, \mathbf{\Lambda} \in \Delta_{\mathbf{X}}$ at $\mathbf{X} \in \mathbf{G}_{m,k}$, the Grassmann metric is defined as $g_{\mathbf{X}}(\Delta, \mathbf{\Lambda}) = \text{trace}(\Delta^\top \mathbf{\Lambda})$. Since both Δ and $\mathbf{\Lambda}$ are $m \times k$ matrices, the trace of their matrix inner product gives the scalar product of the two tangent vectors in \mathbb{R}^d .

Let $f : \mathbf{G}_{m,k} \rightarrow \mathbb{R}$ be a real-valued, scalar function defined on $\mathbf{G}_{m,k}$. We denote by $\partial \mathbf{f}_{\mathbf{X}}$, the $m \times k$ matrix of partial differentials (or the Jacobian w.r.t. \mathbf{X}) of f such that $\partial \mathbf{f}_{\mathbf{X}}(i, j) = \partial f / \partial \mathbf{X}(i, j)$, $1 \leq i \leq m$, $1 \leq j \leq k$. As discussed in Section 2, the j^{th} column vector in $\partial \mathbf{f}_{\mathbf{X}}$ gives the partial differential of f w.r.t. the j^{th} basis vector of \mathbf{X} . Since each entry of $\partial \mathbf{f}_{\mathbf{X}}$ is

computed independently, in general, $\partial \mathbf{f}_{\mathbf{X}}$ does not lie in $\Delta_{\mathbf{X}}$. The gradient of f at \mathbf{X} is the tangent vector $\nabla \mathbf{f}_{\mathbf{X}}$ obtained by subtracting from $\partial \mathbf{f}_{\mathbf{X}}$ its component in subspace spanned by the columns of \mathbf{X} yielding

$$\nabla \mathbf{f}_{\mathbf{X}} = \partial \mathbf{f}_{\mathbf{X}} - \mathbf{X}\mathbf{X}^\top \partial \mathbf{f}_{\mathbf{X}}. \quad (26)$$

It can be verified that $\mathbf{X}^\top (\nabla \mathbf{f}_{\mathbf{X}}) = \mathbf{0}$ and $\text{trace}(\nabla \mathbf{f}_{\mathbf{X}}^\top \Delta) = g_{\mathbf{X}}(\nabla \mathbf{f}_{\mathbf{X}}, \Delta) = \text{trace}(\partial \mathbf{f}_{\mathbf{X}}^\top \Delta)$.

4. Generalized projection based M-Estimator

The generalized projection based M-estimator (gpbM), introduced recently in [22], is a robust subspace estimation algorithm. Both the number of inlier structures and the scale of inlier noise are estimated automatically without any user intervention. The superior performance of the algorithm over other robust estimation methods on challenging synthetic and real world datasets was demonstrated. In [34], the conjugate gradient method was used on Grassmann manifolds as the post-processing step for refining the estimate of the parameter matrix obtained from projection based M-estimator (pbM). However, the optimization on Grassmann manifolds was performed using the scale of the inlier noise that was dependent on the choice of a particular hypothesis.

As opposed to [34], in our case, the employed scale does not depend on a particular hypothesis. In addition, the gpbM can handle heteroscedastic data for single or multiple constraints in a unified framework. Following, we briefly present the gpbM algorithm. For more details, please refer to [22].

Given n_1 measurements of inlier variables $\mathbf{y}_i \in \mathbb{R}^p$, let $\mathbf{x}_i \in \mathbb{R}^m$, $i = 1, \dots, n_1$ represent the corresponding carrier vectors that are usually monomials in a subset of the variables. For example, in the case of fitting an ellipse to measured data $\mathbf{y}_i = [y_1 \ y_2]^\top \in \mathbb{R}^2$, the one-dimensional constraint can be written as

$$\theta_1 y_1 + \theta_2 y_2 + \theta_3 y_1^2 + \theta_4 y_1 y_2 + \theta_5 y_2^2 - \alpha = 0 \quad (27)$$

where $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]^\top$ is the parameter vector and α is the scalar intercept. The corresponding carrier vector is given by $\mathbf{x} = [y_1 \ y_2 \ y_1^2 \ y_1 y_2 \ y_2^2]^\top \in \mathbb{R}^5$. In general, $\boldsymbol{\theta}$ can be an $m \times k$, ($k < m$) orthonormal matrix representing the k constraints satisfied by the inlier points in \mathbb{R}^m . Geometrically, $\boldsymbol{\theta}$ is the basis of the k -dimensional null space of the data. Correspondingly, $\boldsymbol{\alpha}$ is the k -dimensional vector of intercepts.

Given $n(>n_1)$ points \mathbf{x}_i , $i = 1, \dots, n$, 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

$$\boldsymbol{\theta}^\top \mathbf{x}_{i_0} - \boldsymbol{\alpha} = \mathbf{0}_k \quad (28)$$

where \mathbf{x}_{i_0} , $i = 1, \dots, n_1$, are the unknown true values of the inlier carrier points. The multiplicative ambiguity in the estimation of $\boldsymbol{\theta}$ is resolved by requiring $\boldsymbol{\theta}^\top \boldsymbol{\theta} = \mathbf{I}_{k \times k}$. The points \mathbf{x}_i , $i = n_1 + 1, \dots, n$ are outliers and no assumptions are made about their distribution.

The nonlinear relationship between the variables and the carriers makes the estimation problem heteroscedastic, i.e., each carrier vector has a different covariance matrix, and in general can even have different mean. Given $p \times p$ covariance matrices $\mathbf{C}_{\mathbf{y}_i}$ of the variables \mathbf{y}_i , $i = 1, \dots, n$, the first order approximation of the $m \times m$ covariance matrices of \mathbf{x}_i are computed using error propagation [21] as

$$\mathbf{C}_{\mathbf{x}_i} = \mathbf{J}_{\mathbf{x}_i|\mathbf{y}_i}^\top \mathbf{C}_{\mathbf{y}_i} \mathbf{J}_{\mathbf{x}_i|\mathbf{y}_i}, \quad i = 1, \dots, n \quad (29)$$

where $\mathbf{J}_{\mathbf{x}_i|\mathbf{y}_i}$ is the Jacobian of the carrier vector \mathbf{x} with respect to the vector of variables \mathbf{y} and evaluated at \mathbf{y}_i .

To estimate a k -dimensional linear subspace in \mathbb{R}^m , the gpbM algorithm uses elemental subset-based hypotheses generation scheme. Each hypothesis is represented by the pair $[\boldsymbol{\theta}, \boldsymbol{\alpha}]$. Similar to RANSAC, gpbM estimates $\boldsymbol{\theta}$ purely from an elemental subset. However, unlike RANSAC, gpbM refines the estimate of $\boldsymbol{\alpha}$ by using mean shift in the projected space. Similar to [22], the heteroscedastic objective function for a particular hypothesis $[\boldsymbol{\theta}, \boldsymbol{\alpha}]$ is defined as

$$[\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\alpha}}] = \arg \max_{\boldsymbol{\theta}, \boldsymbol{\alpha}} \frac{1}{n} \sum_{i=1}^n \frac{K\left(\left(\frac{(\boldsymbol{\theta}^\top \mathbf{x}_i - \boldsymbol{\alpha})^\top \mathbf{B}_i^{-1} (\boldsymbol{\theta}^\top \mathbf{x}_i - \boldsymbol{\alpha})}{\sqrt{\det \mathbf{B}_i}}\right)^{\frac{1}{2}}\right)}{\sqrt{\det \mathbf{B}_i}} \quad (30)$$

where the kernel function $K(u)$ is related to the M-estimator loss function $\rho(u)$ by $K(u) = 1 - \rho(u)$. The loss function $\rho(u)$ is a redescending M-estimator. It 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 $k \times k$ covariance matrix of each projection $\mathbf{z}_i = \boldsymbol{\theta}^\top \mathbf{x}_i$ is given by $\mathbf{H}_i = \boldsymbol{\theta}^\top \mathbf{C}_{\mathbf{x}_i} \boldsymbol{\theta}$. Note that each $m \times k$ matrix $\boldsymbol{\theta}$ results in a different set of $k \times k$ covariance matrices, $\mathbf{H}_i, i = 1, \dots, n$. The $k \times k$ bandwidth matrices \mathbf{B}_i are given by $\mathbf{B}_i = \mathbf{S}^\top \mathbf{H}_i \mathbf{S}$ where \mathbf{S} is the $k \times k$ diagonal scale matrix, with the diagonal entries corresponding to the value of scale in each dimension of the null space. Each inlier structure is estimated by using a three-step procedure:

- scale estimation,
- mean shift based robust model estimation,
- inlier/outlier dichotomy.

Multiple inlier structures are estimated iteratively by removing the points associated with previously detected structures from the data. The algorithm stops once no more significant inlier structures are detected.

4.1. Heteroscedastic scale estimation

To estimate the approximate fraction of data points belonging to an inlier structure, we generate M elemental subset-based model hypotheses. We vary the value of fraction between $(0, 1]$ in Q steps, such that for $q = 1, \dots, Q$, the fraction is $\eta_q = q/Q = n_q/n$. In all our experiments, the value of Q was set to 40. The scale of the inlier noise is estimated by taking into account the heteroscedasticity of the carrier vector. Given a hypothesis $[\boldsymbol{\theta}, \boldsymbol{\alpha}]$, let

$$\text{vol}^q(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \sqrt{\sum_{i=1}^{n_q} (\mathbf{z}_i - \boldsymbol{\alpha})^\top \mathbf{H}_i^{-1} (\mathbf{z}_i - \boldsymbol{\alpha})} \quad (31)$$

be the volume around the intercept $\boldsymbol{\alpha}$ containing n_q points, where $\mathbf{z}_i, i = 1, \dots, n_q$ are the Mahalanobis distance based n_q nearest neighbors of $\boldsymbol{\alpha}$ in the null space. For each q , the density for the fraction η_q is computed using the volume $\text{vol}^q(\boldsymbol{\theta}, \boldsymbol{\alpha})$ as $\psi^q(\boldsymbol{\theta}, \boldsymbol{\alpha}) = n_q / (\text{vol}^q(\boldsymbol{\theta}, \boldsymbol{\alpha}) + \epsilon)$, where a small constant ϵ is added to suppress extremely high values of densities at lower fractions.

The densities for all M hypotheses and all Q fractions are then stacked into an $M \times Q$ matrix $\boldsymbol{\psi}$. For every q , let the number of rows of $\boldsymbol{\psi}$ that have the maximum density in the q^{th} column be J_q . The corresponding set of density values are ψ_{\max}^q with $\psi_{\max}^q \subset \psi_{\max}^q$ containing the η_q/J_q highest densities. For example, for $\eta_q = 0.1$, only the highest 0.1 $J_{0.1}$ densities are taken into account. The sum of highest density values for every q over the subset ψ_{\max}^q is then computed as

$$\hat{\psi}_{\text{sub}}^q = \sum_{\psi \in \psi_{\max}^q} \psi. \quad (32)$$

For every q , by summing over $\hat{\psi}_{\text{sub}}^q$, 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 inlier fraction $\eta_{\hat{q}}$ is computed using

$$\hat{q} = \arg \max_q \hat{\psi}_{\text{sub}}^q. \quad (33)$$

The projections \mathbf{z}_i of the data points \mathbf{x}_i are then computed for the hypothesis that gives the highest density at $\eta_{\hat{q}}$. The dimensions of the smallest rectangular region in the k -dimensional null space containing $n_{\hat{q}}$ points, divided by two, gives the estimate of the scale which forms the diagonal of \mathbf{S} . The corresponding $n_{\hat{q}}$ points form an initial estimate of the inliers. The algorithm for scale estimation is summarized in the following steps.

1. Generate M model hypotheses using elemental subsets.
2. For each hypothesis $[\boldsymbol{\theta}, \boldsymbol{\alpha}]$,
 - 2.1. Compute the projections $\mathbf{z}_i = \boldsymbol{\theta}^\top \mathbf{x}_i, i = 1, \dots, n$.
 - 2.2. Vary the value of unknown fraction η_q between $(0, 1]$ in Q uniform steps such that for $q = 1, \dots, Q$, the fraction is $\eta_q = q/Q = n_q/n$.
 - 2.3. For each q , compute the volume $\text{vol}^q(\boldsymbol{\theta}, \boldsymbol{\alpha})$ around the intercept $\boldsymbol{\alpha}$ using (31).
 - 2.4. Compute corresponding density $\psi^q(\boldsymbol{\theta}, \boldsymbol{\alpha}) = n_q / (\text{vol}^q(\boldsymbol{\theta}, \boldsymbol{\alpha}) + \epsilon)$.
3. For each fraction η_q and all M hypotheses,
 - 3.1. Compute the number of peak density values over all hypotheses J_q .
 - 3.2. From the set of peak density values ψ_{\max}^q , compute the sum of η_q/J_q highest densities $\hat{\psi}_{\text{sub}}^q$ using (32).
4. Using (33), compute the index \hat{q} . The estimate of the unknown fraction is given by $\eta_{\hat{q}}$.
5. Compute the projections of the data points $\mathbf{x}_i, i = 1, \dots, n$ to the hypothesis that maximizes the density at $\eta_{\hat{q}}$.
6. The dimensions of the smallest k -dimensional rectangle in the null space containing $n_{\hat{q}}$ points, divided by two, gives the estimate of the scale.

4.2. Model estimation

We generate N elemental subset based model hypotheses. However, the selection of elemental subsets is restricted over the initial set of inliers returned by the scale estimation step, making the model estimation very efficient. For a given hypothesis, the original estimation problem of (30) is reformulated into a problem of estimating the kernel density in k dimensions. The adaptive kernel density function over the projections $\mathbf{z}_i = \boldsymbol{\theta}^\top \mathbf{x}_i \in \mathbb{R}^k, i = 1, \dots, n$ is defined as

$$f(\boldsymbol{\theta}, \mathbf{z}) = \frac{1}{n} \sum_{i=1}^n \frac{\kappa\left(\frac{(\mathbf{z} - \mathbf{z}_i)^\top \mathbf{B}_i^{-1} (\mathbf{z} - \mathbf{z}_i)}{\sqrt{\det \mathbf{B}_i}}\right)}{\sqrt{\det \mathbf{B}_i}} \quad (34)$$

where $\kappa(u^2) = K(u)$ is the profile of the kernel $K(u)$. Taking the derivative of (34) w.r.t. \mathbf{z} ,

$$\frac{df(\boldsymbol{\theta}, \mathbf{z})}{d\mathbf{z}} = \frac{2}{n} \sum_{i=1}^n \mathbf{B}_i^{-1} (\mathbf{z} - \mathbf{z}_i) \frac{g\left(\frac{(\mathbf{z} - \mathbf{z}_i)^\top \mathbf{B}_i^{-1} (\mathbf{z} - \mathbf{z}_i)}{\sqrt{\det \mathbf{B}_i}}\right)}{\sqrt{\det \mathbf{B}_i}} = 0 \quad (35)$$

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

$$\delta \mathbf{z} = \left[\sum_{i=1}^n \frac{\mathbf{B}_i^{-1} g(\dots)}{\sqrt{\det \mathbf{B}_i}} \right]^{-1} \left[\sum_{i=1}^n \frac{\mathbf{B}_i^{-1} \mathbf{z}_i g(\dots)}{\sqrt{\det \mathbf{B}_i}} \right] - \mathbf{z}. \quad (36)$$

The mean shift procedure is initiated from $\mathbf{z}^{(0)} = \boldsymbol{\alpha}^{(0)}$ (with a slight abuse of the notation) which is the projection of the elemental subset points on $\boldsymbol{\theta}$. The iteration $\mathbf{z}^{(j+1)} = \delta \mathbf{z}^{(j)} + \mathbf{z}^{(j)}$, is a gradient ascent step converging to the closest mode, $\boldsymbol{\alpha}$.

4.2.1. Conjugate gradient on the Grassmann manifold

The estimation of the intercept α is performed using mean shift in the projected space. On the contrary, the parameter matrix Θ is selected purely from the set of random hypotheses. This estimate can be inaccurate, for example, if the selected elemental subset points have significant inlier noise. Since, each Θ is an $m \times k$ orthogonal matrix, it can be represented as a point on the Grassmann manifold, $\mathbf{G}_{m,k}$. Optimization techniques to maximize the objective function of (34) over $\mathbf{G}_{m,k}$ can therefore be employed. Note that since we are only interested in the subspace associated with Θ and not in any particular set of k basis vectors, optimization on Stiefel manifold is not required.

The conjugate gradient algorithm is a popular non-linear optimization method. It has fast convergence since it does not compute the Hessian of the function. The method finds a solution by iteratively optimizing the objective function in linearly independent directions called the *conjugate directions*. A very good introduction to the method is given in [29] and ([26], pp. 420–425). Like many numerical optimization methods, function optimization along a particular direction is performed using *line search* methods. Two such methods are given in ([28], pp. 213–217). We use Brent's method ([26], pp. 402–405) which can find a minimum along a line after it has been bracketed.

In [5], a new approach of conjugate gradient algorithm to maximize a function $f : \mathbf{G}_{m,k} \rightarrow \mathbb{R}$ on the Grassmann manifold $\mathbf{G}_{m,k}$ was proposed. In our case, since the optimization is performed over both Θ and α the domain of our objective function of (34) is $\mathbf{G}_{m,k} \times \mathbb{R}^k$. The choice of conjugate gradient algorithm over standard gradient descent is justified due to the unknown degree of nonlinearity of the objective function on Grassmann manifolds. Choosing an appropriate value for step-size in gradient descent algorithm is therefore very difficult. It can vary with the dimensionality of the manifold, the choice of a particular hypothesis and the distribution of data points. Each iteration of conjugate gradient is computationally more expensive than that of gradient descent. However, since we are only interested in locally optimum point, the algorithm usually converges in very few iterations.

Similar to [34], we extend the original algorithm of [5] over the new product space $\mathbf{G}_{m,k} \times \mathbb{R}^k$. However, in [34], due to the employed Θ -dependent scale matrix \mathbf{S}_Θ , the optimization function (34) was explicitly assumed being independent of \mathbf{S}_Θ . Since the scale matrix \mathbf{S}_Θ was critically dependent on the hypothesis $[\Theta, \alpha]$, this assumption was quite unrealistic for computing the necessary conjugate directions. Contrary to this, in our case, the estimated scale matrix is independent of Θ and no such assumption is required. Also, in [34], only homoscedastic applications were presented. Our method can handle both homoscedastic and heteroscedastic data.

Conjugate gradient method originally being a function minimization algorithm, we seek to minimize $f_\diamond([\Theta, \alpha]) = -f([\Theta, \alpha])$. Since $f_\diamond([\Theta, \alpha])$ is minimized in the product space $\mathbf{G}_{m,k} \times \mathbb{R}^k$, both Θ and α need to be updated simultaneously in each iteration of the algorithm. The matrix Θ is updated on the Grassmann manifold, $\mathbf{G}_{m,k}$, while the vector α is updated independently in \mathbb{R}^k .

Given an estimated pair $[\Theta, \alpha]$, with a slight abuse of the notation, the initial gradient of the objective function $f_\diamond([\Theta, \alpha])$ w.r.t. Θ on $\mathbf{G}_{m,k}$ is computed using (26) as

$$\nabla \mathbf{f}_\Theta = \partial \mathbf{f}_\Theta - \Theta \Theta^T \partial \mathbf{f}_\Theta \tag{37}$$

where $\partial \mathbf{f}_\Theta$ is the Jacobian of $f_\diamond([\Theta, \alpha])$ w.r.t. Θ . The corresponding gradient w.r.t. α is given by

$$\nabla \mathbf{f}_\alpha = \partial \mathbf{f}_\alpha \tag{38}$$

where $\partial \mathbf{f}_\alpha$ is the Jacobian of $f_\diamond([\Theta, \alpha])$ w.r.t. α .

The Jacobians $\partial \mathbf{f}_\Theta$ and $\partial \mathbf{f}_\alpha$ depend on the choice of the kernel function and are computed by assuming an *explicit independence* among

Θ , α and the covariance matrices $\mathbf{H}_i, i = 1, \dots, n$. While computing $\partial \mathbf{f}_\Theta$ and $\partial \mathbf{f}_\alpha$, the covariance matrices \mathbf{H}_i are not differentiated w.r.t. Θ and α . It can be shown that making this assumption is equivalent to performing a first order approximation of the Jacobians. For Epanechnikov kernel defined as

$$K(u) \approx \begin{cases} 1 - u^2 & \text{if } |u| \leq 1 \\ 0 & \text{if } |u| > 1 \end{cases} \tag{39}$$

and for $j = 1, \dots, k, l = 1, \dots, m$, the entries of the $m \times k$ matrix $\partial \mathbf{f}_\Theta$ are

$$\partial \mathbf{f}_\Theta(l, j) = -\frac{1}{n} \sum_{i=1}^n \frac{1}{\sqrt{\det \mathbf{B}_i}} \left(\frac{\partial K(u)}{\partial \Theta(l, j)} \right) = \frac{2}{n} \sum_{i=1}^n \mathbf{p}_i(j) \mathbf{x}_i(l) \tag{40}$$

where $u = \left((\Theta^T \mathbf{x}_i - \alpha)^T \mathbf{B}_i^{-1} (\Theta^T \mathbf{x}_i - \alpha) \right)^{\frac{1}{2}}$ and $\mathbf{p}_i = \frac{\mathbf{B}_i^{-1} (\Theta^T \mathbf{x}_i - \alpha)}{\sqrt{\det \mathbf{B}_i}}$. The entries of the k -dimensional vector $\partial \mathbf{f}_\alpha$ are given as

$$\partial \mathbf{f}_\alpha(j) = -\frac{1}{n} \sum_{i=1}^n \frac{1}{\sqrt{\det \mathbf{B}_i}} \left(\frac{\partial K(u)}{\partial \alpha(j)} \right) = -\frac{2}{n} \sum_{i=1}^n \mathbf{p}_i(j) \quad j = 1, \dots, k. \tag{41}$$

Similar expressions of the Jacobians $\partial \mathbf{f}_\Theta$ and $\partial \mathbf{f}_\alpha$ of $f_\diamond([\Theta, \alpha])$ for other kernels can be derived.

Given the parameter pair $[\Theta^{(0)}, \alpha^{(0)}]$ estimated using mean shift, let $\Gamma^{(0)} = \nabla \mathbf{f}_{\Theta^{(0)}}$ and $\gamma^{(0)} = \nabla \mathbf{f}_{\alpha^{(0)}}$ be the gradient directions obtained using (37) and (38). The initial search directions are set as the negative of the gradients giving $\Lambda^{(0)} = -\Gamma^{(0)}$ and $\lambda^{(0)} = -\gamma^{(0)}$. At the j^{th} iteration of the conjugate gradient algorithm, the following steps are executed.

1. Using line search, minimize $f_\diamond([\Theta^{(j)}, \alpha^{(j)}])$ along the geodesic

$$\Theta^{(j)}(t) = \Theta^{(j)} \mathbf{V} \cos(\Sigma t) \mathbf{V}^T + \mathbf{U} \sin(\Sigma t) \mathbf{V}^T \tag{42}$$

where $\mathbf{U} \Sigma \mathbf{V}^T$ is the compact SVD of the conjugate direction $\Lambda^{(j)}$. This is equivalent to finding t_Θ , the value of the parameter t for which the function $f_\diamond([\Theta^{(j)}(t), \alpha^{(j)}])$ is minimum.

2. Using line search in \mathbb{R}^k , minimize $f_\diamond([\Theta^{(j)}, \alpha^{(j)}(t)])$ along the conjugate direction

$$\alpha^{(j)}(t) = \alpha^{(j)} + t \lambda^{(j)}. \tag{43}$$

This is equivalent to finding t_α , the value of the parameter t for which the function $f_\diamond([\Theta^{(j)}, \alpha^{(j)}(t)])$ is minimum.

3. Set $\Theta^{(j+1)} = \Theta^{(j)}(t_\Theta)$ and $\alpha^{(j+1)} = \alpha^{(j)}(t_\alpha)$.
4. Using (20), parallel transport the tangent vectors $\Lambda^{(j)}$ and $\Gamma^{(j)}$ to the point $\Theta^{(j+1)}$ as

$$\Gamma_r^{(j)} = \Gamma^{(j)} - \left(\Theta^{(j)} \mathbf{V} \sin \Sigma t_\Theta + \mathbf{U} [1 - \cos \Sigma t_\Theta] \right) \mathbf{U}^T \Gamma^{(j)} \tag{44}$$

$$\Lambda_r^{(j)} = \Lambda^{(j)} - \left(\Theta^{(j)} \mathbf{V} \sin \Sigma t_\Theta + \mathbf{U} [1 - \cos \Sigma t_\Theta] \right) \mathbf{U}^T \Lambda^{(j)}. \tag{45}$$

By substituting $\Lambda^{(j)} = \mathbf{U} \Sigma \mathbf{V}^T$

$$\Lambda_r^{(j)} = \left(-\Theta^{(j)} \mathbf{V} \sin \Sigma t_\Theta + \mathbf{U} \cos \Sigma t_\Theta \right) \Sigma \mathbf{V}^T. \tag{46}$$

5. Compute the new gradient directions as

$$\Gamma^{(j+1)} = \nabla \mathbf{f}_{\Theta^{(j+1)}} = \partial \mathbf{f}_{\Theta^{(j+1)}} - \left(\Theta^{(j+1)} \right) \left(\Theta^{(j+1)} \right)^T \partial \mathbf{f}_{\Theta^{(j+1)}} \tag{47}$$

$$\gamma^{(j+1)} = \nabla \mathbf{f}_{\alpha^{(j+1)}} = \partial \mathbf{f}_{\alpha^{(j+1)}}. \tag{48}$$

6. Compute the new conjugate search directions as

$$\mathbf{\Lambda}^{(j+1)} = -\mathbf{\Gamma}^{(j+1)} + \omega^{(j)} \mathbf{\Lambda}_r^{(j)} \quad (49)$$

$$\boldsymbol{\lambda}^{(j+1)} = -\boldsymbol{\gamma}^{(j+1)} + \omega^{(j)} \boldsymbol{\lambda}^{(j)} \quad (50)$$

where

$$\omega^{(j)} = \frac{\text{trace}\left(\left[\mathbf{\Gamma}^{(j+1)} - \mathbf{\Gamma}_r^{(j)}\right]^\top \mathbf{\Gamma}^{(j+1)}\right) + \left[\boldsymbol{\gamma}^{(j+1)} - \boldsymbol{\gamma}^{(j)}\right]^\top \boldsymbol{\gamma}^{(j+1)}}{\text{trace}\left(\left(\mathbf{\Gamma}^{(j)}\right)^\top \mathbf{\Gamma}^{(j)}\right) + \left(\boldsymbol{\gamma}^{(j)}\right)^\top \boldsymbol{\gamma}^{(j)}}. \quad (51)$$

Note that for every j , the covariance matrices $\mathbf{H}_i^{(j)} = \left(\boldsymbol{\theta}^{(j)}\right)^\top \mathbf{C}_{\mathbf{x}_i} \boldsymbol{\theta}^{(j)}$, $i = 1, \dots, n$ should ideally be recomputed and inverted to estimate $f_\diamond(\boldsymbol{\theta}, \boldsymbol{\alpha})$. However, due to high computational complexity, the covariance matrices are kept constant $\mathbf{H}_i^{(j)} = \mathbf{H}_i^{(0)}$, $j = 1, 2, \dots$, $i = 1, \dots, n$. In practice, inverting each \mathbf{H}_i at every iteration of the algorithm does not change the final estimate significantly. The estimated intercept $\hat{\boldsymbol{\alpha}}$ corresponds to the location of the highest mode over all N hypotheses, while the corresponding matrix $\hat{\boldsymbol{\theta}}$ is the estimate of $\boldsymbol{\theta}$.

4.2.2. Stopping criterion

To decide if all the inliers structures present in the data have been discovered, we compute the strength of the current inlier structure as $\xi = f(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\alpha}}) / \|\mathbf{S}\|^2$ where \mathbf{S} is the corresponding estimated scale matrix. 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 gross outliers.

4.3. Inlier/outlier dichotomy

Given the estimated model $[\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\alpha}}]$, we first normalize the deviation of each point from the mode by its point-dependent covariance. For $\mathbf{z}_i = \boldsymbol{\theta}^\top \mathbf{x}_i$, $i = 1, \dots, n$, the heteroscedastic projections around the mode are computed as

$$\tilde{\mathbf{z}}_i = \hat{\boldsymbol{\alpha}} + \frac{(\mathbf{z}_i - \hat{\boldsymbol{\alpha}})}{\|\mathbf{z}_i - \hat{\boldsymbol{\alpha}}\|_2} \sqrt{(\mathbf{z}_i - \hat{\boldsymbol{\alpha}})^\top \mathbf{H}_i^{-1} (\mathbf{z}_i - \hat{\boldsymbol{\alpha}})}. \quad (52)$$

The inlier/outlier dichotomy is then performed by starting *homoscedastic* mean shift iterations from each point $\tilde{\mathbf{z}}_i$ with $\tilde{\mathbf{H}}_i = \mathbf{I}_{k \times k}$, $i = 1, \dots, n$. The bandwidth matrices $\tilde{\mathbf{B}}_i = \mathbf{S}^\top \mathbf{I}_{k \times k} \mathbf{S} = \mathbf{S}^\top \mathbf{S}$ also become constant for all points. The points for which the mean shift procedure converges to $\hat{\boldsymbol{\alpha}}$ (with small tolerance) are considered inliers.

5. Experimental results

We present three groups of experiments. First we show the performance of our algorithm on a synthetic multiple line fitting example. Then we present two real-world applications: fundamental matrix estimation and affine motion factorization. While the problems of line fitting and affine factorization are homoscedastic due to linear relationship between the carriers and variables, the problem of fundamental matrix estimation is heteroscedastic. Other heteroscedastic applications can be found in the program <http://coewww.rutgers.edu/riul/research/code/GPBM/index.html>.

In [22], it was shown that the gpbM algorithm performs better than other robust estimation methods. In this paper, we emphasize on the importance of optimization on Grassmann manifolds. It should also be noted that comparing gpbM to RANSAC based methods is also quite tricky, especially for multi-dimensional problems, since RANSAC requires an accurate estimate of the scale of inlier noise in all dimensions of the residual space. Also, similar to [34], RANSAC like methods do not estimate the model heteroscedastically [27]. To recapitulate, M and N are the number of elemental subset-based random

hypotheses used in the scale estimation and model estimation steps. Also, $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ represent the subspace parameter matrix (a point on a Grassmann manifold) and the intercept, respectively.

5.1. Multiple line fitting

In this example, the kernel density is estimated using the Epanechnikov kernel. We generated data points along three different lines in 2D having 100 (blue), 150 (red) and 200 (green) points (Fig. 3). For all three lines, the x and y coordinates were independently corrupted with zero-mean Gaussian noise with standard deviations of 0.2σ (blue), 0.15σ (red) and 0.1σ (green). A total of seven different

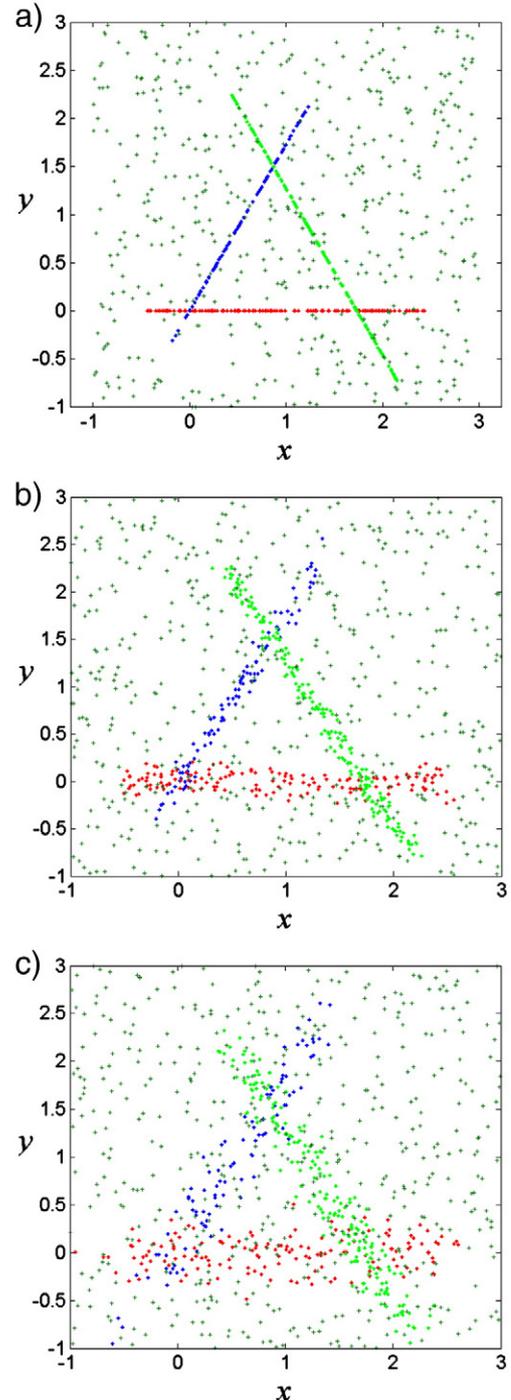


Fig. 3. Three intersecting lines. Sample input data for (a) $\sigma=0$, (b) $\sigma=0.6$, and (c) $\sigma=1.2$.

Table 1

Average error in the estimation of Θ and α over 100 runs relative to the ground truth values. CG stands for conjugate gradient on Grassmann manifolds.

σ	Average error in Θ (degrees)		Average error in α (units)	
	gpbM	gpbM + CG	gpbM	gpbM + CG
0.0	0.2515	0.2494	0.0627	0.0159
0.2	0.3563	0.3239	0.0771	0.0208
0.4	0.4919	0.4337	0.1375	0.0266
0.6	0.6417	0.6395	0.1694	0.0429
0.8	1.0327	0.8761	0.2330	0.1240
1.0	1.6126	1.5859	0.2924	0.2726
1.2	2.3383	2.1741	0.3332	0.2749

values of the parameter σ were tried by varying σ uniformly between [0, 1.2] in steps of 0.2. In addition, 500 random outliers were also added between [-1,-1] and [3, 3]. The values of $M=500$ and $N=200$ were used. For each σ , we compare the performance of the 100 runs of gpbM algorithm with and without performing the optimization on Grassmann manifolds. Table 1 summarizes the results.

It is clear from Table 1 that the performance of the gpbM algorithm improves if optimization on Grassmann manifolds is performed. The amount of improvement in general depends on the amount of inlier noise. For Θ , the average error increases quasi-linearly with the increasing noise and the improvement is independent of the slope of the lines. The corresponding change in α is not linear and is more sensitive to the estimated Θ .

5.2. Fundamental matrix estimation

In fundamental matrix estimation $\Theta \in \mathbb{R}^8$. 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. Using the epipolar constraint with the homogeneous image coordinates (without the points at infinity)

$$[x_2 \ y_2 \ 1] \mathbf{F}_{3 \times 3} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = 0 \tag{53}$$

the carrier vector is written as $\mathbf{x} = [x_1 \ y_1 \ x_2 \ y_2 \ x_1 x_2 \ x_1 y_2 \ y_1 x_2 \ y_1 y_2]^\top$ which lies in \mathbb{R}^8 . Assuming the variables \mathbf{y} have covariance $\sigma^2 \mathbf{I}_{4 \times 4}$, the first order approximation of the covariance matrix of \mathbf{x} is computed from the Jacobian using error propagation [21]

$$\mathbf{J}_{\mathbf{x}|\mathbf{y}} = \begin{bmatrix} 1 & 0 & 0 & 0 & x_2 & y_2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & x_2 & y_2 \\ 0 & 0 & 1 & 0 & x_1 & 0 & y_1 & 0 \\ 0 & 0 & 0 & 1 & 0 & x_1 & 0 & y_1 \end{bmatrix} = [\mathbf{I}_{4 \times 4} \ \mathbf{J}(\mathbf{y})] \tag{54}$$

$$\mathbf{C}_{\mathbf{x}} = \sigma^2 \mathbf{J}_{\mathbf{x}|\mathbf{y}}^\top \mathbf{I}_{4 \times 4} \mathbf{J}_{\mathbf{x}|\mathbf{y}} = \sigma^2 \begin{bmatrix} \mathbf{I}_{4 \times 4} & \mathbf{J}(\mathbf{y}) \\ \mathbf{J}(\mathbf{y})^\top & \mathbf{J}(\mathbf{y}) \mathbf{J}(\mathbf{y})^\top \end{bmatrix}. \tag{55}$$

We test our algorithm on two pairs of images. In both cases, the Epanechnikov kernel is employed.

5.2.1. Oxford corridor images

In this experiment, the first and tenth frames from the Oxford corridor sequence obtained from <http://www.robots.ox.ac.uk/~vgg/data/data-mview.html> were used (Fig. 4). Notice the large disparity between the two images. Using the SIFT algorithm [18], 127 point matches were obtained, out of which 57 were true inliers. We compared the performance of the gpbM algorithm over 100 runs with and without performing the optimization on Grassmann manifolds. The values of $M=400$ and $N=200$ were used. On average, the gpbM algorithm misclassified 6.14 (out of 127) points, while only 4.7 points were classified wrongly after using the conjugate gradient algorithm. The average absolute residual error for the 57 true inlier points using gpbM algorithm was 1.79 pixels, while it was 1.74 pixels after using the conjugate gradient algorithm. The corresponding median absolute residual error was 1.76 pixels without and 1.71 pixels with optimization on the Grassmann manifold.

5.2.2. Moulin Rouge images

In this experiment, two images of the Moulin Rouge building from the Paris dataset obtained from <http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/index.html> were used (Fig. 5). Again, the disparity between the two images is large. Using the SIFT algorithm [18], 182 point matches were obtained, out of which 114 were true inliers. We compared the performance of the gpbM algorithm over 100 runs with and without performing the optimization on Grassmann manifolds. Again, the values of $M=400$ and $N=200$ were used. On average, the gpbM algorithm misclassified 30.91 (out of 182) points, while only 23.22 points were classified wrongly after using the conjugate gradient algorithm. The relatively large number of misclassifications is due to scalar nature of the epipolar constraint. The average absolute residual error for the 114 true inlier points using gpbM algorithm was 0.6520 pixels while it was 0.6320 pixels after using the conjugate gradient algorithm. The corresponding median absolute residual error was 0.6794 without and 0.6707 pixels with optimization on the Grassmann manifold.

5.3. Affine motion factorization

If n_1 rigidly moving points, all lying on an affine motion, are tracked over F images, then the $2F$ image coordinates can be used to define feature vectors in \mathbb{R}^{2F} . Factorization is based on the fact that,

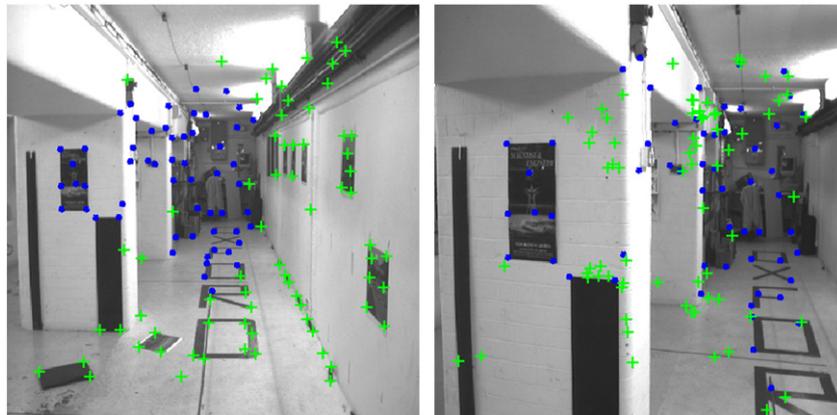


Fig. 4. Two images from the Oxford Corridor sequence. The true inliers are marked with blue dots and the outliers with green '+' signs.

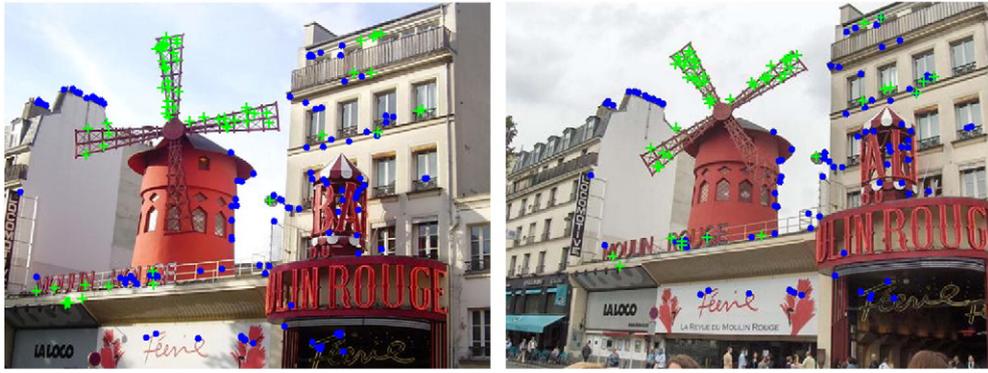


Fig. 5. Two images of the *Moulin Rouge* building from *Paris* dataset. The true inliers are marked with blue dots and the outliers with green '+' signs.

these vectors lie only in a four-dimensional subspace of \mathbb{R}^{2F} . If the data is centered then the dimension of the subspace is only three. For more details, see ([11], pp. 436–439). In [22], the gpbM algorithm was used for projective motion estimation. Since that requires a lot of theoretical introduction, in this paper, we present experiments for affine motion factorization since our aim is just to demonstrate that optimization on Grassmann manifolds improves the performance of gpbM.

We present two different experiments. In both experiments, for estimating the kernel density, we used *Tukey's biweight* kernel which is given as

$$K(u) \approx \begin{cases} (1-u^2)^3 & \text{if } |u| \leq 1 \\ 0 & \text{if } |u| > 1 \end{cases}. \quad (56)$$

Table 2

Average percentage of misclassified points. CG stands for conjugate gradient on Grassmann manifolds. The number in the parenthesis in the first column shows the true number of motions for each sequence. Ext-MS and Int-MS stand for extrinsic and intrinsic non-linear mean shift methods. The results of pbM + CG, gpbM and gpbM + CG were averaged over 50 runs while those of Ext-MS and Int-MS were averaged over 20 runs.

Sequence	Ext-MS [35]	Int-MS [2]	pbM + CG [34]	gpbM	gpbM + CG
arm(2)	30.65%	27.73%	61.03%	7.99%	7.79%
articulated(3)	30.17%	24.50%	40.26%	6.90%	6.70%
cars1(2)	20.07%	23.00%	26.38%	6.51%	5.96%
cars2(2)	11.90%	9.08%	12.65%	3.58%	3.55%
cars4(2)	21.60%	11.94%	31.56%	7.55%	7.31%
cars5(3)	19.94%	19.41%	21.48%	8.93%	8.05%
cars6(2)	5.68%	7.09%	5.82%	1.86%	1.85%
cars8(2)	42.71%	35.29%	27.86%	7.31%	6.97%
truck1(2)	28.56%	13.24%	50.47%	6.27%	6.09%
2RT3RC(3)	12.52%	7.40%	26.50%	10.92%	10.06%
Overall	17.91%	14.64%	23.75%	6.58%	6.18%

For each sequence, $F=5$ frames were used by picking every sixth or seventh frame from the sequence. Similar to [2], we first test our algorithm on ten video sequences containing multiple motions from *Hopkins155* dataset. This dataset is available online at <http://www.vision.jhu.edu/data/hopkins155> and contains no gross outliers. For this experiment, the values of $M=1000$ and $N=200$ were used. Table 2 compares the performance of the gpbM algorithm with and without using conjugate gradient on Grassmann manifolds over 50 runs. For comparison, we present the results of the pbM algorithm [34] and of using the intrinsic [2] and extrinsic [35] mean shift methods for subspace clustering over Grassmann manifolds. The results of [34] were average over 50 runs while those from [2, 35] were averaged over 20 runs. None of the methods assume the knowledge of the true number of motions.

In the second experiment, we compare the performance of two algorithms on the *parking lot* sequence of [22]. In [22] results from only a single run of the experiment were presented. Here we present the results of more detailed experiments performed over this sequence both using the original gpbM [22] and its proposed extension. This sequence contains four motions and gross outliers. There were a total of 474 points – 213 (background), 78 (black car), 75 (silver car), 46 (maroon car) and 62 gross outliers. Fig. 6 shows the first and last frames of the sequence with points lying on various motions marked with different colors. For this experiment, the values of $M=2000$ and $N=400$ were used. Over 50 runs, the gpbM algorithm misclassified 11.94% of the total points on average, which was reduced to 10.27% when optimization on Grassmann manifolds was performed. Table 3 shows the corresponding confusion matrices for the two cases averaged over 50 runs.

6. Conclusions

The generalized projection based M-estimator (gpbM) can robustly estimate multiple, noisy inlier structures in the presence of a lot of outliers. The scale of the inlier noise and the number of inlier



Fig. 6. First and last frames of the *parking lot* sequence with four motions and gross outliers. White points marked '+' show gross outliers.

Table 3

Confusion matrices averaged over 50 runs – without (top) and with (bottom) using conjugate gradient algorithm on Grassmann manifolds. M1, M2, M3 and M4 correspond to points lying on black car, silver car, maroon car and background respectively. The detection results are shown along rows.

	M1	M2	M3	M4	Outliers
M1	66.1	7.2	3.6	1.0	1.1
M2	0.0	67.1	0.0	0.6	7.3
M3	0.0	0.0	44.1	0.0	1.9
M4	0.6	0.0	5.3	185.7	20.4
Outliers	2.3	0.0	3.3	2.0	54.4
M1	69.5	7.4	0.0	1.0	1.1
M2	0.0	70.9	0.0	0.1	4.0
M3	0.0	0.0	42.6	0.0	3.4
M4	0.7	0.0	7.4	183.0	20.9
Outliers	0.0	0.0	0.7	2.0	59.3

structures are also estimated automatically without any user intervention. By exploiting the geometric properties of Riemannian manifolds, the original robust subspace estimation problem of gpbM is extended to a problem of maximizing the kernel density function on Grassmann manifolds. We demonstrated that by using this extension the performance of the original gpbM algorithm can be improved. As opposed to [34], where explicit independence was assumed between the objective function and the scale of inlier noise, in our case this assumption is automatically satisfied due to the way in which the scale is estimated. Also, contrary to [34], our algorithm is able to estimate the model heteroscedastically.

References

- [1] V. Arsigny, P. Pillard, X. Pennec, N. Ayache, Geometric means in a novel vector space structure on symmetric positive-definite matrices, *SIAM J. Matrix Anal. Appl.* 29 (2006) 328–347.
- [2] H. Çetingül, R. Vidal, Intrinsic mean shift for clustering on Stiefel and Grassmann manifolds, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 2009, pp. 1896–1902.
- [3] R. Chaudhry, Y. Ivanov, Fast approximate nearest neighbor methods for non-Euclidean manifolds with applications to human activity analysis in videos, Proceedings of the European Conference on Computer Vision, Heraklion, Crete, Greece, 2010, pp. 735–748.
- [4] Y. Chikuse, *Statistics on Special Manifolds*, Springer, 2003.
- [5] A. Edelman, T.A. Arias, S.T. Smith, The geometry of algorithms with orthogonality constraints, *SIAM J. Matrix Anal. Appl.* 20 (1998) 303–353.
- [6] Z. Fan, J. Zhou, Y. Wu, Multibody grouping by inference of multiple subspaces from high-dimensional data using oriented-frames, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (2006) 91–105.
- [7] M.A. Fischler, R.C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM* 24 (1981) 381–395.
- [8] P.T. Fletcher, C. Lu, S. Joshi, Statistics of shape via principal component analysis on Lie groups, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Madison, WI, vol. 1, 2003, pp. 95–101.
- [9] K. Gallivan, A. Srivastava, X. Liu, P.V. Dooren, Efficient algorithms for inferences on Grassmann manifolds, Proceedings of the 12th IEEE Workshop on Statistical Signal Processing, St. Louis, MO, 2003, pp. 315–318.
- [10] V.M. Govindu, Lie-algebraic averaging for globally consistent motion estimation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, vol. 1, 2004, pp. 684–691.
- [11] R.I. Hartley, F. Schaffalitzky, Reconstruction from projections using Grassmann tensors, *Int. J. Comput. Vision* 83 (2009) 274–293.
- [12] U. Helmke, K. Hüper, P.Y. Lee, J. Moore, Essential matrix estimation using Gauss-Newton iterations on a manifold, *Int. J. Comput. Vision* 74 (2007) 117–136.
- [13] H. Karcher, Riemannian center of mass and mollifier smoothing, *Commun. Pure Appl. Math.* 30 (1977) 509–541.
- [14] M. Li, W. Chen, K. Huang, T. Tan, Visual tracking via incremental self-tuning particle filtering on the affine group, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 1315–1322.
- [15] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, J. Cheng, Visual tracking via incremental log-Euclidean Riemannian subspace learning, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008.
- [16] D. Lin, E. Grimson, J. Fisher, Learning visual flows: A Lie algebraic approach, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 2009, pp. 747–754.
- [17] D. Lin, S. Yan, X. Tang, Pursuing informative projection on Grassmann manifold, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, 2006, pp. 1727–1734.
- [18] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vision* 60 (2004) 91–110.
- [19] Y. Lui, J. Beveridge, M. Kirby, Action classification on product manifolds, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 833–839.
- [20] Y. Ma, J. Košecká, S. Sastry, Optimization criteria and geometric algorithms for motion and structure estimation, *Int. J. Comput. Vision* 44 (2001) 219–249.
- [21] B. Matei, P. Meer, Estimation of nonlinear errors-in-variables models for computer vision applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (2006) 1537–1552.
- [22] S. Mittal, S. Anand, P. Meer, Generalized projection based M-estimator: theory and applications, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, 2011, pp. 2689–2696.
- [23] M. Moakher, Means and averaging in the group of rotations, *SIAM J. Matrix Anal. Appl.* 24 (2002) 1–16.
- [24] B. O’Neill, *Semi-Riemannian Geometry with Applications to Relativity*, Academic Press, 1983.
- [25] X. Pennec, P. Fillard, N. Ayache, A Riemannian framework for tensor computing, *Int. J. Comput. Vision* 66 (2006) 41–66.
- [26] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes*, second edition, Cambridge University Press, 1992.
- [27] R. Raguram, J.M. Frahm, M. Pollefeys, A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus, Proceedings of the European Conference on Computer Vision, Marseille, France, 2008, pp. 500–513.
- [28] A. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, 2006.
- [29] J.R. Shewchuk, An introduction to the conjugate gradient method without the agonizing pain, Technical Report CMU-CS-94-125, School of Computer Science, Carnegie Mellon University, 1994.
- [30] S. Soatto, R. Frezza, P. Perona, Recursive motion estimation on the essential manifold, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 1993, pp. 428–433.
- [31] S. Soatto, R. Frezza, P. Perona, Motion estimation via dynamic vision, *IEEE Trans. Autom. Control* 41 (1996) 393–413.
- [32] A. Srivastava, E. Klassen, Bayesian and geometric subspace tracking, *Adv. Appl. Probab.* 36 (2004) 43–56.
- [33] R. Subbarao, Y. Genc, P. Meer, Robust unambiguous parametrization of the essential manifold, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008.
- [34] R. Subbarao, P. Meer, Subspace estimation using projection based M-estimators over Grassmann manifolds, Proceedings of the European Conference on Computer Vision, Graz, Austria, vol. 1, 2006, pp. 301–312.
- [35] R. Subbarao, P. Meer, Nonlinear mean shift over Riemannian manifolds, *Int. J. Comput. Vision* 84 (2009) 1–20.
- [36] D. Tosato, M. Farenzena, M. Cristani, M. Spera, V. Murino, Multi-class classification on Riemannian manifolds for video surveillance, Proceedings of the European Conference on Computer vision, Heraklion, Crete, Greece, 2010, pp. 378–391.
- [37] P. Turaga, A. Veeraraghavan, A. Srivastava, R. Chellappa, Statistical computations on Grassmann and Stiefel manifolds for image and video based recognition, To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.
- [38] O. Tuzel, F. Porikli, P. Meer, Learning on Lie groups for invariant detection and tracking, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008.
- [39] O. Tuzel, F. Porikli, P. Meer, Pedestrian detection via classification on Riemannian manifolds, *IEEE Trans. Pattern Anal. Mach. Intell.* 30 (2008) 1713–1727.
- [40] O. Tuzel, R. Subbarao, P. Meer, Simultaneous multiple 3D motion estimation via mode finding on Lie groups, Proceedings of the International Conference of Computer Vision, Beijing, China, 2005, pp. 18–25.
- [41] J. Yan, M. Pollefeys, A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and nondegenerate, Proceedings of the European Conference on Computer Vision, Graz, Austria, 2006, pp. 94–106.