

Robust Estimation for Computer Vision using Grassmann Manifolds

Saket Anand¹, Sushil Mittal² and Peter Meer³

Abstract Real-world visual data is often corrupted and requires the use of estimation techniques that are robust to noise and outliers. Robust methods are well studied for Euclidean spaces and their use has also been extended to Riemannian spaces. In this chapter, we present the necessary mathematical constructs for Grassmann manifolds, followed by two different algorithms that can perform robust estimation on them. In the first one, we describe a nonlinear mean shift algorithm for finding modes of the underlying kernel density estimate (KDE). In the second one, a user-independent robust regression algorithm, the generalized projection based M-estimator (gpbM) is detailed. We show that the gpbM estimates are significantly improved if KDE optimization over the Grassmann manifold is also included. The results for a few real-world computer vision problems are shown to demonstrate the importance of performing robust estimation using Grassmann manifolds.

1 Introduction

Estimation problems in geometric computer vision often require dealing with orthogonality constraints in the form of linear subspaces. Since orthogonal matrices representing linear subspaces of Euclidean space can be represented as points on Grassmann manifolds, understanding the geometric properties of these manifolds can prove very useful for solving many vision problems. Usually, the estimation process involves optimizing an objective function to find the regression coefficients that best describe the underlying constraints. Alternatively, given a distribution of sampled hypotheses of linear solutions, it could also be formulated as finding the cluster centers of those distributions as the dominant solutions to the underlying observations.

A typical regression problem in computer vision involves discovering multiple, noisy inlier structures present in the data corrupted with gross outliers. Usually, very little or no information is available about the number of inlier structures, the nature of the noise corrupting each one of them and the amount of gross outliers. The original RANdom SAMple Consensus (RANSAC) [5] and its several variants like MLESAC, LO-RANSAC,

1. IIIT-Delhi, New Delhi, India. e-mail: anands@iiitd.ac.in

2. Scibler Corporation, Santa Clara, CA, USA. e-mail: mittal@scibler.com

3. Dept. of ECE, Rutgers University, NJ, USA. e-mail: meer@cronos.rutgers.edu

PROSAC, QDEGSAC [15] were designed for problems with single inlier structure where either a *fixed* estimate of the scale of inlier noise is provided by the user beforehand or it is estimated using a simple heuristic. As the complexity of the problems grow, a scale estimate becomes harder to determine. Although Grassmann manifolds provide a continuous parameter space to optimize and refine the estimate returned by a robust algorithm, it is still essential for the original algorithm to perform well in situations where the data deviates from the underlying model in a variety of different ways. Accurate estimation of the scale of inlier noise, especially for multi-dimensional problems is an important step for all robust algorithms.

Robust methods applied to the more general Riemannian manifolds have appeared in computer vision literature. A short introduction to Riemannian manifolds, mainly from a computer vision point of view and some applications can be seen in [13] and [19], as well as the references therein. More recent work in this area could be found in [6, 8, 10, 16, 20, 21, 22]. In case of Grassmann manifolds, such techniques amount to clustering and finding the modes of the subspace distributions obtained from the data.

This chapter is organized as follows. We introduce in Section 2, the necessary tools for solving subspace estimation and clustering problems on Grassmann manifolds. Section 3 describes two robust subspace estimation algorithms - the nonlinear mean shift and the generalized projection based M-estimator (gpbM). A few applications of the two algorithms are presented in Section 4. We conclude with a discussion in Section 5.

2 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 $m > k$. The point \mathbf{X} is represented by an $m \times k$ orthogonal matrix, i.e., $\mathbf{X}^\top \mathbf{X} = \mathbf{I}_{k \times k}$ and is independent of the choice of any particular basis vectors. Hence, 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 [4].

Tangent vectors at \mathbf{X} to $\mathbf{G}_{m,k}$ are also represented as $m \times k$ matrices. Given a real valued function $f : \mathbf{G}_{m,k} \rightarrow \mathbb{R}$ on the manifold, $\Delta(f)$ is the magnitude of the derivative of f in the tangent direction Δ at \mathbf{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 horizontal and vertical spaces. The space normal to the tangent space is called the normal space. See Fig. 1a for an illustration of this decomposition. A 3×3 matrix of $\mathbf{G}_{3,1}$ has two-dimensional horizontal, one-dimensional vertical and six-dimensional normal space.

A *curve* in a Riemannian manifold \mathcal{M} is a smooth mapping α from an open interval \mathbf{T} of \mathbb{R} to \mathcal{M} . For a particular $t \in \mathbf{T}$, $\mathbf{X}(t)$ lies on the manifold

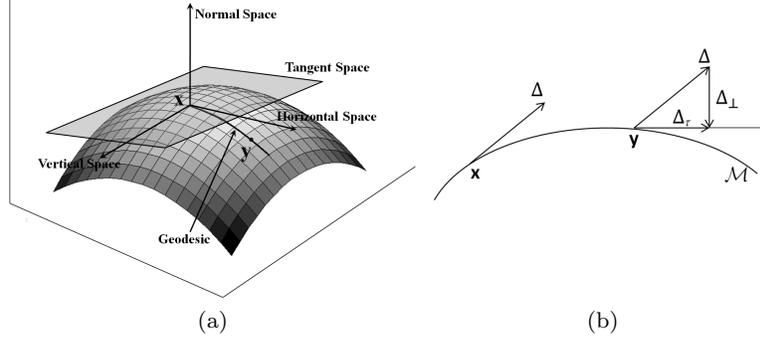


Fig. 1 (a) Tangent and normal spaces of Grassmann (Riemannian) manifold with the geodesic along the direction of the horizontal space. (b) Parallel transport of tangent Δ from \mathbf{X} to \mathbf{Y} on the manifold by removing the normal component Δ_{\perp} .

and $\mathbf{X}'(t)$ is a tangent vector at $\mathbf{X}(t)$. Given points \mathbf{X}, \mathbf{Y} on \mathcal{M} , the shortest curve connecting \mathbf{X} and \mathbf{Y} is called the geodesic. The length of the geodesic is defined to be the *Riemannian distance* between the two points. It can be shown that for Grassmann manifold $\mathbf{G}_{m,k}$, a geodesic from \mathbf{X} , in the direction of the tangent $\mathbf{\Lambda}$ ($m \times k$ matrix) can be written as

$$\mathbf{X}(t) = [\mathbf{X}\mathbf{V} \quad \mathbf{U}] \begin{bmatrix} \cos \mathbf{\Sigma} t \\ \sin \mathbf{\Sigma} t \end{bmatrix} \mathbf{V}^{\top} \quad (1)$$

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

In Euclidean space, a vector can be moved parallel to itself by just moving the base of the vector. Moving a tangent vector Δ from a point \mathbf{X} to \mathbf{Y} on the manifold \mathcal{M} also accumulates a normal component Δ_{\perp} at \mathbf{Y} , which is subtracted from the transported vector. This is called *parallel translation* and is illustrated in Fig. 1b. A tangent vector $\Delta \in \Delta_{\mathbf{X}}$ at $\mathbf{X} = \mathbf{X}(0)$ can be parallel-translated to another point $\mathbf{Y} \in \mathcal{M}$ by infinitesimally removing the normal component of the translated vector, Δ_{\perp} along the path between \mathbf{X} and \mathbf{Y} on the manifold. For Grassmann manifold $\mathbf{G}_{m,k}$, the parallel-translation of Δ along the geodesic in direction $\mathbf{\Lambda}$ is given by

$$\Delta(t) = \left([\mathbf{X}\mathbf{V} \quad \mathbf{U}] \begin{bmatrix} -\sin \mathbf{\Sigma} t \\ \cos \mathbf{\Sigma} t \end{bmatrix} \mathbf{U}^{\top} + [\mathbf{I} - \mathbf{U}\mathbf{U}^{\top}] \right) \Delta \quad (2)$$

where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$ is the compact SVD of $\mathbf{\Lambda}$, and $\Delta(0) = \Delta$.

The horizontal space of the tangent vectors and geodesics on the manifold are closely related. There is a unique geodesic curve $\alpha : [0, 1] \rightarrow \mathbf{G}_{m,k}$, starting at \mathbf{X} with tangent vector Δ , which has the initial velocity $\alpha'(0) = \Delta$.

The *exponential map*, $\exp_{\mathbf{X}}$, maps Δ from the tangent space to the point on the manifold reached by this geodesic

$$\exp_{\mathbf{X}}(\Delta) = \alpha(1) = \mathbf{X}(1). \quad (3)$$

where $\mathbf{X}(1)$ is computed using (1). 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 around the origin of $\Delta_{\mathbf{X}}$ that can be uniquely mapped to a neighborhood of \mathbf{X} via $\exp_{\mathbf{X}}$. The inverse mapping is achieved by 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 tangent direction $\mathbf{\Lambda}$ such that the geodesic from \mathbf{X} along $\mathbf{\Lambda}$ reaches \mathbf{Y} in unit time. With $[\mathbf{X} \ \mathbf{X}_{\perp}]^{\top} [\mathbf{X} \ \mathbf{X}_{\perp}] = \mathbf{I}_m$, let \mathcal{C} and \mathcal{S} represent the generalized singular values of $\mathbf{X}^{\top} \mathbf{Y}$ and $\mathbf{X}_{\perp}^{\top} \mathbf{Y}$ such that $\mathcal{C}^{\top} \mathcal{C} + \mathcal{S}^{\top} \mathcal{S} = \mathbf{I}_k$. With some computation, it can be shown that for $\mathbf{G}_{m,k}$, the logarithm operator can be written as

$$\log_{\mathbf{X}}(\mathbf{Y}) = \mathbf{\Lambda} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^{\top} = \mathbf{U} \cos^{-1}(\mathcal{C}_1) \mathbf{V}^{\top} = \mathbf{U} \sin^{-1}(\mathcal{S}_1) \mathbf{V}^{\top} \quad (4)$$

where $\mathcal{C}_1 = \begin{bmatrix} \mathcal{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{(m-k)} \end{bmatrix}$ and $\mathcal{S}_1 = \begin{bmatrix} \mathcal{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{(m-k) \times (m-k)} \end{bmatrix}$ and \cos^{-1} and \sin^{-1} act element-by-element along the diagonals of \mathcal{C}_1 and \mathcal{S}_1 . The exponential and logarithm operators vary as the point \mathbf{X} moves on $\mathbf{G}_{m,k}$ which is made explicit above by the subscript.

The distance between two points on the Grassmann manifold using (4) is given by

$$d(\mathbf{X}, \mathbf{Y}) = \|\log_{\mathbf{X}}(\mathbf{Y})\|_F \quad (5)$$

where $\|\cdot\|_F$ is the matrix Frobenius norm. The gradient of the squared Riemannian distance for Grassmann manifolds [19] is

$$\nabla_{\mathbf{X}} d^2(\mathbf{X}, \mathbf{Y}) = -2 \log_{\mathbf{X}}(\mathbf{Y}). \quad (6)$$

For a real-valued, scalar function $f : \mathbf{G}_{m,k} \rightarrow \mathbb{R}$ on $\mathbf{G}_{m,k}$, let $\partial \mathbf{f}_{\mathbf{X}}$ be the $m \times k$ Jacobian of f w.r.t. \mathbf{X} such that $\partial \mathbf{f}_{\mathbf{X}}(i, j) = \partial f / \partial \mathbf{X}(i, j)$, $i = 1, \dots, m$, and $j = 1, \dots, k$. 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 the tangent space $\Delta_{\mathbf{X}}$. 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

$$\nabla \mathbf{f}_{\mathbf{X}} = \partial \mathbf{f}_{\mathbf{X}} - \mathbf{X} \mathbf{X}^{\top} \partial \mathbf{f}_{\mathbf{X}} = \mathbf{X}_{\perp} \mathbf{X}_{\perp}^{\top} \partial \mathbf{f}_{\mathbf{X}}. \quad (7)$$

It is easy to verify that $\mathbf{X}^{\top}(\nabla \mathbf{f}_{\mathbf{X}}) = \mathbf{0}$. See [13] for proofs of most of the above equations.

3 Robust Estimation using Grassmann Manifolds

First, we describe the nonlinear-mean shift algorithm, which takes a clustering-based approach to identify dominant subspace hypotheses over the Grassmann manifold. Next, we describe the generalized projection-based M-estimator (gpbM), which improves the subspace estimate over the Grassmann manifold by using conjugate gradient optimization method.

3.1 Nonlinear Mean Shift on Grassmann Manifolds

The mean shift algorithm [3] takes an iterative approach for identifying local modes of the underlying kernel density estimate (KDE). In Euclidean space, this is achieved by computing weighted means of sample points in a local neighborhood. As opposed to this, the mean of points lying on a Grassmann manifold itself may not lie on the manifold. However, the tangent vectors of these points exist in a vector space, where their weighted mean can be computed and used to update the mode estimate. This method has been generalized to many Riemannian manifolds [19].

Given n points on the Grassmann manifold, $\mathbf{X}_i, i = 1, \dots, n$, the kernel density estimate with a kernel profile κ and bandwidth h is

$$\hat{f}_\kappa(\mathbf{Y}) = \frac{c_{\kappa,h}}{n} \sum_{i=1}^n \kappa \left(\frac{d^2(\mathbf{Y}, \mathbf{X}_i)}{h^2} \right) \quad (8)$$

where $c_{\kappa,h}$ is the normalization constant and $d(\cdot, \cdot)$ represents the Riemannian distance for Grassmann manifolds computed using (5). The kernel profile κ is related to the kernel function as $K_h(u) = c_{\kappa,h} \kappa(u^2)$ [3]. The bandwidth h , which is a tuning parameter in the application, is provided by the user. It can also be thought of as the scale parameter in the distance function.

The gradient of \hat{f}_κ at \mathbf{Y} is calculated as

$$\begin{aligned} \nabla \hat{f}_\kappa(\mathbf{Y}) &= \frac{1}{n} \sum_{i=1}^n \nabla \kappa \left(\frac{d^2(\mathbf{Y}, \mathbf{X}_i)}{h^2} \right) \\ &= -\frac{1}{n} \sum_{i=1}^n g \left(\frac{d^2(\mathbf{Y}, \mathbf{X}_i)}{h^2} \right) \frac{\nabla d^2(\mathbf{Y}, \mathbf{X}_i)}{h^2} \\ &= \frac{2}{n} \sum_{i=1}^n g \left(\frac{d^2(\mathbf{Y}, \mathbf{X}_i)}{h^2} \right) \frac{\log_{\mathbf{Y}}(\mathbf{X}_i)}{h^2} \end{aligned} \quad (9)$$

where, $g(\cdot) = -\kappa'(\cdot)$, and the relation (6) is used in the last equation of (9). The gradient of the distance function is taken here with respect to \mathbf{Y} . Like in the Euclidean case, the updated nonlinear mean shift vector is then

computed as

$$\delta\mathbf{Y} = \frac{\sum_{i=1}^n g\left(\frac{d^2(\mathbf{Y}, \mathbf{X}_i)}{h^2}\right) \log_{\mathbf{Y}}(\mathbf{X}_i)}{\sum_{i=1}^n g\left(\frac{d^2(\mathbf{Y}, \mathbf{X}_i)}{h^2}\right)}. \quad (10)$$

This expression is defined in a vector space since $\log_{\mathbf{Y}}(\mathbf{X}_i)$ terms lie in the tangent space $\Delta_{\mathbf{Y}}$ and the kernel terms $g(d^2(\mathbf{Y}, \mathbf{X}_i)/h^2)$ are scalars. Since the mean shift vector is computed in the tangent space, i.e., not on the manifold intrinsically, this algorithm was referred to as the *extrinsic* mean shift in [1]. On the other hand, the intrinsic mean shift (Int-MS) proposed in [1] operates directly on the manifold. As shown in Sec. 4.4, there is little difference in performance of the two mean shift procedures. The mean shift vector (10) computed in $\Delta_{\mathbf{Y}}$ is projected back to the Grassmann manifold using the exponential map for the j^{th} iteration

$$\mathbf{Y}^{(j+1)} = \exp_{\mathbf{Y}^{(j)}}(\delta\mathbf{Y}^{(j)}). \quad (11)$$

Each iteration of (11) updates the current mode estimate $\mathbf{Y}^{(j)}$ by moving along the geodesic defined by the mean shift vector to get the next estimate, $\mathbf{Y}^{(j+1)}$. A mean shift iteration is initialized at each data point by initializing $\mathbf{X} = \mathbf{X}_i$ and repeatedly updated until convergence. The complete nonlinear mean shift algorithm is shown in Fig. 2.

NONLINEAR MEAN SHIFT OVER GRASSMANN MANIFOLDS

Given: Points on Grassmann manifold $\mathbf{X}_i, i = 1, \dots, n$

for $i \leftarrow 1 \dots n$

Y $\leftarrow \mathbf{X}_i$

repeat

$$\delta\mathbf{Y} \leftarrow \frac{\sum_{i=1}^n g(d^2(\mathbf{Y}, \mathbf{X}_i)/h^2) \log_{\mathbf{Y}}(\mathbf{X}_i)}{\sum_{i=1}^n g(d^2(\mathbf{Y}, \mathbf{X}_i)/h^2)}$$

Y $\leftarrow \exp_{\mathbf{Y}}(\delta\mathbf{Y})$

until $\|\delta\mathbf{Y}\| < \epsilon$

Retain **Y** as a local mode

Report distinct local modes.

Fig. 2 Nonlinear mean shift over Grassmann manifolds.

The mean shift procedure is initiated from every hypothesis point on the manifold. Points whose iterations converges into a particular mode belong to its basin of attraction. Dominant modes are identified as the ones having

high kernel density estimates and a large number of points in their basins of attraction. Spurious modes are characterized by low kernel density estimates and fewer points in their basins of attraction and can be easily pruned.

3.2 Generalized Project Based M-estimators (gpbM)

The generalized projection based M-estimator (gpbM) [12] is a robust subspace estimation algorithm that works on the hypothesize and test principle. The scale of the inlier noise, the number of inlier structures and the associated model hypotheses are automatically estimated by gpbM without any user intervention. The gpbM can also handle heteroscedastic data for single or multiple carrier problems in a unified framework. The original pbM algorithm [17] performed scale estimation for each newly chosen elemental subset based hypothesis by computing the median absolute deviation (MAD) estimate in each dimension separately. More recently, a completely different scale estimation strategy was presented in [12] showing superior performance over competing methods. However, the model parameters were estimated by optimization over a discrete set of parameter matrices. In [13], these estimates were refined over the Grassmann manifold as a continuous space of parameter matrices, which led to significant performance improvements.

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 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 this setting, robust subspace estimation corresponds to performing a linear estimation in the carrier space of the data corrupted with outliers and containing an unknown number of noisy inlier points. We have $n (> n_1)$ points $\mathbf{x}_i, i = 1, \dots, n$, where $n - n_1$ points are outliers. The parameter matrix Θ is an $m \times k, (k < m)$, orthonormal matrix such that $\Theta^\top \Theta = \mathbf{I}_{k \times k}$, and therefore can be represented as a point on the Grassmann manifold $\mathbf{G}_{m,k}$. Geometrically, it is a basis of the k -dimensional *null* space of the inlier data representing the k constraints imposed on them. The $\alpha \in \mathbb{R}^k$ is the corresponding k -dimensional vector of intercepts. Therefore

$$\Theta^\top \mathbf{x}_{i_o} - \alpha = \mathbf{0}_k \quad (12)$$

where $\mathbf{x}_{i_o}, i = 1, \dots, n_1$, are the *unknown* true values of the inlier carrier points. The multiplicative ambiguity in the estimation of Θ is resolved by enforcing $\Theta^\top \Theta = \mathbf{I}_{k \times k}$. No assumptions are made about the distribution of the $n - n_1$ outlier points.

The carrier vectors are often nonlinear in the variables, thereby making the estimation problem *heteroscedastic*, i.e., each carrier vector has a different

noise covariance matrix, and in general can even have a different mean. Given the covariance matrices of the observed variables, $\mathbf{C}_{\mathbf{y}_i}$ and the *Jacobians* of the carrier vectors with respect to those variables, $\mathbf{J}_{\mathbf{x}_i|\mathbf{y}_i}$, the first order approximation of the $m \times m$ carrier covariances can be computed using error propagation as

$$\mathbf{C}_{\mathbf{x}_i} = \mathbf{J}_{\mathbf{x}_i|\mathbf{y}_i}^\top \mathbf{C}_{\mathbf{y}_i} \mathbf{J}_{\mathbf{x}_i|\mathbf{y}_i}, \quad i = 1, \dots, n. \quad (13)$$

The covariance of the vector of variables $\mathbf{C}_{\mathbf{y}_i}$ is often assumed to be the same for all i , but in general, can be different for each \mathbf{y}_i . Since some carriers are associated with outliers, their covariances are computed incorrectly.

For each projected point $\mathbf{z}_i = \mathbf{\Theta}^\top \mathbf{x}_i$, the $k \times k$ covariance matrix is computed as $\mathbf{H}_i = \mathbf{\Theta}^\top \mathbf{C}_{\mathbf{x}_i} \mathbf{\Theta}$. The $k \times k$ point-dependent bandwidth matrices \mathbf{B}_i are computed as $\mathbf{B}_i = \mathbf{S}^\top \mathbf{H}_i \mathbf{S}$ using the $k \times k$ diagonal scale matrix \mathbf{S} , with the diagonal entries corresponding to the value of scale of inlier noise in each dimension of the null space. Therefore, in order to compute \mathbf{B}_i , we need to estimate the k -dimensional scale first. In gpbM, each inlier structure is estimated by using a three-step procedure:

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

In case of multiple structures, the set of inlier points associated with each detected structure is removed iteratively from the data and the three step procedure is repeated until no more significant inlier structures are found.

The gpbM algorithm follows a hypothesize-then-test strategy such that an estimate of the parameter pair $[\mathbf{\Theta}, \boldsymbol{\alpha}]$ is computed from a randomly selected elemental subset, i.e., minimal set of points necessary to generate a subspace hypothesis. The hypothesis corresponding to the best model over the set of all randomly generated hypotheses is selected by maximizing the following heteroscedastic objective function

$$[\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\alpha}}] = \arg \max_{\boldsymbol{\Theta}, \boldsymbol{\alpha}} \frac{1}{n} \sum_{i=1}^n \frac{K \left(\left((\boldsymbol{\Theta}^\top \mathbf{x}_i - \boldsymbol{\alpha})^\top \mathbf{B}_i^{-1} (\boldsymbol{\Theta}^\top \mathbf{x}_i - \boldsymbol{\alpha}) \right)^{\frac{1}{2}} \right)}{\sqrt{\det \mathbf{B}_i}} \quad (14)$$

where $K(u)$ is the kernel function and is related to a redescending M-estimator loss function by $K(u) = 1 - \rho(u)$. The loss function $\rho(u)$ is non-negative, symmetric, non-decreasing with $|u|$ and has a unique minimum of $\rho(0) = 0$ and a maximum of one for $|u| > 1$.

Step 1: Heteroscedastic Scale Estimation. The estimation of the scale of inlier noise is equivalently posed as the problem of estimating the approximate fraction of data points belonging to an inlier structure. For estimating the fraction, M elemental subset-based model hypotheses are generated. For each hypothesis $[\boldsymbol{\Theta}, \boldsymbol{\alpha}]$, the value of fraction is varied between $(0, 1]$ in Q steps,

such that for $q = 1, \dots, Q$, the fraction is $\eta_q = q/Q = n_q/n$. A value of $Q = 40$ was used for all applications shown in [12, 13]. The scale of the inlier noise is estimated by taking into account the heteroscedasticity of the carrier vector. In the projected space defined by Θ , the volume around the intercept α containing n_q points is computed as

$$\text{vol}^q(\Theta, \alpha) = \sqrt{\sum_{l=1}^{n_q} (\mathbf{z}_l - \alpha)^\top \mathbf{H}_l^{-1} (\mathbf{z}_l - \alpha)} \quad (15)$$

where $\mathbf{z}_l, l = 1, \dots, n_q$ are Mahalanobis distance-based n_q nearest neighbors of α . Given a hypothesis $[\Theta, \alpha]$, the density at each fraction $\eta_q, q = 1, \dots, Q$ is computed as $n_q/(\text{vol}^q(\Theta, \alpha) + \epsilon)$, where a small constant ϵ is added to suppress extremely high numerical values of the densities at low fractions. The computed density values are used to populate an $M \times Q$ matrix Ψ .

From the matrix Ψ , the peak density values along each of the M hypotheses (rows) and their corresponding fractions (columns) are retained. Typically, some columns are associated with several density peaks while other columns are not. At each column q , a fraction η_q of the highest peak density values are summed up. The largest sum of density peaks corresponds to $\eta_{\hat{q}}$, i.e., the estimate of the fraction of points comprising an inlier structure. This approach makes the fraction estimate more robust, especially when multiple inlier structures exist, each comprising very different number of points.

The hypothesis $[\Theta^*, \alpha^*]$ that gives the highest density at $\eta_{\hat{q}}$ is used to project the data points \mathbf{x}_i . The dimensions of the smallest k -dimensional rectangle enclosing the $n_{\hat{q}}$ nearest neighbors of α^* provide the final estimate of the scale, which forms the diagonal of \mathbf{S} . The corresponding $n_{\hat{q}}$ points enclosed inside the rectangle form an initial estimate of the inliers.

Step 2: Model Estimation. Model estimation is performed by generating N elemental subset based model hypotheses. However, only the initial set of inliers returned by the scale estimation step are used for the selection of elemental subsets, making the model estimation very efficient. For a given hypothesis $[\Theta, \alpha]$, the problem 14 can be re-written as that of estimating the kernel density of the data points projected to the k -dimensional null space. The *adaptive* kernel density function over the projections $\mathbf{z}_i = \Theta^\top \mathbf{x}_i \in \mathbb{R}^k, i = 1, \dots, n$ is defined as

$$\hat{f}_\kappa(\Theta, \mathbf{z}) = \frac{1}{n} \sum_{i=1}^n \frac{\kappa((\mathbf{z} - \mathbf{z}_i)^\top \mathbf{B}_i^{-1} (\mathbf{z} - \mathbf{z}_i))}{\sqrt{\det \mathbf{B}_i}} \quad (16)$$

where $\kappa(u^2) = K(u)$ is the profile of the kernel function $K(u)$. Differentiating (16) w.r.t. \mathbf{z} ,

$$\frac{d\hat{f}_\kappa(\Theta, \mathbf{z})}{d\mathbf{z}} = \frac{2}{n} \sum_{i=1}^n \mathbf{B}_i^{-1}(\mathbf{z} - \mathbf{z}_i) \frac{g((\mathbf{z} - \mathbf{z}_i)^\top \mathbf{B}_i^{-1}(\mathbf{z} - \mathbf{z}_i))}{\sqrt{\det \mathbf{B}_i}} = 0 \quad (17)$$

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

$$\delta\mathbf{z} = \left[\sum_{i=1}^n \frac{\mathbf{B}_i^{-1}g(\dots)}{\sqrt{\det \mathbf{B}_i}} \right]^{-1} \left[\sum_{i=1}^n \frac{\mathbf{B}_i^{-1}\mathbf{z}_i g(\dots)}{\sqrt{\det \mathbf{B}_i}} \right] - \mathbf{z}. \quad (18)$$

The mean shift procedure is initiated from $\mathbf{z}^{(0)}$, i.e., the projection of the elemental subset points on Θ . The update $\mathbf{z}^{(j+1)} = \delta\mathbf{z}^{(j)} + \mathbf{z}^{(j)}$, is a gradient ascent step converging to α , the closest mode of the KDE (16).

Step 2.1: Conjugate Gradient on the Grassmann manifold. Each Θ is an $m \times k$ orthogonal matrix and can be represented as a point on the Grassmann manifold, $\mathbf{G}_{m,k}$. Continuous optimization techniques to maximize the objective function of (16) over $\mathbf{G}_{m,k}$ can therefore be employed.

The conjugate gradient algorithm is widely used for optimization of non-linear objective functions defined over Euclidean spaces. This popularity is due to fast convergence rates achieved by iteratively moving along *linearly independent* directions in the solution space. Moreover, it avoids computing the Hessian, thus making each iteration less expensive than other alternatives like Newton's method. The optimization along a chosen direction is performed using *line search* methods and in this case Brent's method [14, pp. 402–405] is used. Edelman et al. [4] adapted the conjugate gradient algorithm to minimize a function $f : \mathbf{G}_{m,k} \rightarrow \mathbb{R}$ over the Grassmann manifold $\mathbf{G}_{m,k}$.

Conjugate gradient method originally being a function *minimization* algorithm, the function $f_\diamond(\Theta, \alpha) = -\hat{f}_\kappa(\Theta, \alpha)$ is optimized. The function $f_\diamond(\Theta, \alpha)$ is jointly minimized over its domain $\mathbf{G}_{m,k} \times \mathbb{R}^k$ with each iteration of conjugate gradient simultaneously updating both $\Theta \in \mathbf{G}_{m,k}$ and $\alpha \in \mathbb{R}^k$. Given an estimated pair $[\Theta, \alpha]$, the initial gradient of the objective function $f_\diamond(\Theta, \alpha)$ w.r.t. Θ on $\mathbf{G}_{m,k}$ is computed using (7) as

$$\nabla \mathbf{f}_\Theta = \partial \mathbf{f}_\Theta - \Theta \Theta^\top \partial \mathbf{f}_\Theta \quad (19)$$

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

$$\nabla \mathbf{f}_\alpha = \partial \mathbf{f}_\alpha \quad (20)$$

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 to a first order approximation. This is equivalent to assuming an *explicit independence* among Θ , α and the covariance matrices $\mathbf{H}_i, i = 1, \dots, n$, i.e, the Jacobian computation does not involve differentiating \mathbf{H}_i w.r.t. Θ and α . For the Epanechnikov kernel defined as

$$K(u) \simeq \begin{cases} 1 - u^2 & \text{if } |u| \leq 1 \\ 0 & \text{if } |u| > 1 \end{cases} \quad (21)$$

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

$$\partial \mathbf{f}_{\Theta}(l, j) = -\frac{1}{n} \sum_{i=1}^n \frac{1}{\sqrt{\det \mathbf{B}_i}} \left(\frac{\partial \kappa(u^2)}{\partial \Theta(l, j)} \right) = \frac{2}{n} \sum_{i=1}^n \mathbf{p}_i(j) \mathbf{x}_i(l) \quad (22)$$

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

$$\partial \mathbf{f}_{\alpha}(j) = -\frac{1}{n} \sum_{i=1}^n \frac{1}{\sqrt{\det \mathbf{B}_i}} \left(\frac{\partial \kappa(u^2)}{\partial \alpha(j)} \right) = -\frac{2}{n} \sum_{i=1}^n \mathbf{p}_i(j) \quad j = 1, \dots, k. \quad (23)$$

The conjugate gradient algorithm is initialized by setting the optimization variables to $[\Theta^{(0)}, \alpha^{(0)}]$ as estimated in the model estimation step. The initial search directions are taken to be the negative gradient direction, i.e., $\Lambda^{(0)} = -\nabla \mathbf{f}_{\Theta^{(0)}}$ and $\lambda^{(0)} = -\nabla \mathbf{f}_{\alpha^{(0)}}$, computed using (19) and (20). For each iteration j , Brent's method is applied for minimization of f_{\diamond} in directions along $[\Lambda^{(j)}, \alpha^{(j)}]$ and the variables $[\Theta^{(j+1)}, \alpha^{(j+1)}]$ are updated to this directional minimum. Both the search and the gradient directions on the Grassmann manifold are parallel-translated to the newly updated location $\Theta^{(j+1)}$ using (2) and are denoted by $\Lambda_{\tau}^{(j)}$ and $\nabla_{\tau} \mathbf{f}_{\Theta^{(j)}}$ respectively. The equivalent operations for $\lambda^{(j)}$ and $\mathbf{f}_{\alpha^{(j)}}$ are simply the Euclidean translations in \mathbb{R}^k . The new gradient directions $[\nabla \mathbf{f}_{\Theta^{(j+1)}}, \nabla \mathbf{f}_{\alpha^{(j+1)}}]$ are computed at the updated points and the resulting conjugate directions are

$$\begin{aligned} \Lambda^{(j+1)} &= -\nabla \mathbf{f}_{\Theta^{(j+1)}} + \omega^{(j)} \Lambda_{\tau}^{(j)} \\ \lambda^{(j+1)} &= -\nabla \mathbf{f}_{\alpha^{(j+1)}} + \omega^{(j)} \lambda^{(j)} \end{aligned} \quad (24)$$

where

$$\omega^{(j)} = \frac{\text{trace} \left([\nabla \mathbf{f}_{\Theta^{(j+1)}} - \nabla_{\tau} \mathbf{f}_{\Theta^{(j)}}]^\top \nabla \mathbf{f}_{\Theta^{(j+1)}} \right) + [\nabla \mathbf{f}_{\alpha^{(j+1)}} - \nabla \mathbf{f}_{\alpha^{(j)}}]^\top \nabla \mathbf{f}_{\alpha^{(j+1)}}}{\text{trace} \left((\nabla \mathbf{f}_{\Theta^{(j)}})^\top \nabla \mathbf{f}_{\Theta^{(j)}} \right) + (\nabla \mathbf{f}_{\alpha^{(j)}})^\top \nabla \mathbf{f}_{\alpha^{(j)}}}.$$

While the covariance matrices $\mathbf{H}_i^{(j)} = (\Theta^{(j)})^\top \mathbf{C}_{\mathbf{x}_i} \Theta^{(j)}$, $i = 1, \dots, n$, should ideally be recomputed in each iteration, it was shown in [13] that maintaining $\mathbf{H}_i^{(j)} = \mathbf{H}_i^{(0)}$, $j = 1, 2, \dots$, $i = 1, \dots, n$ reduced the computational cost significantly without noticeable changes in the final estimates. After convergence of the conjugate gradient, the optimization variables $[\hat{\Theta}, \hat{\alpha}]$ gives the final estimate of the parameter matrix and the intercept.

Step 3: Inlier/Outlier Dichotomy. Given the estimated model $[\hat{\Theta}, \hat{\alpha}]$, the deviation of each point from the mode is normalized by its point-dependent covariance. For $\mathbf{z}_i = \hat{\Theta}^\top \mathbf{x}_i, i = 1, \dots, n$, the heteroscedastic projections around the mode are computed as

$$\tilde{\mathbf{z}}_i = \hat{\alpha} + \frac{(\mathbf{z}_i - \hat{\alpha})}{\|\mathbf{z}_i - \hat{\alpha}\|_2} \sqrt{(\mathbf{z}_i - \hat{\alpha})^\top \mathbf{H}_i^{-1} (\mathbf{z}_i - \hat{\alpha})} \quad (25)$$

This reduces the inlier/outlier dichotomy to homoscedastic mean shift clustering problem since 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 iterations converge to $\hat{\alpha}$ (within a small tolerance) are considered inliers.

Using the estimated scale matrix \mathbf{S} , the strength of the detected inlier structure is computed as $\xi = f(\hat{\Theta}, \hat{\alpha}) / \|\mathbf{S}\|^2$. The algorithm stops if the strength drops by a factor of 20 compared to the maximum strength among the previously computed inlier structures, indicating that the remaining points comprise only gross outliers.

4 Applications

A result of the nonlinear mean shift for chromatic noise filtering is presented, followed by the application of generalized project based M-estimators (gpbM) algorithm applied to fundamental matrix and homography estimation. Finally, a quantitative comparison of both methods with related robust estimation techniques is presented using a subset of the Hopkins155 dataset.

4.1 Chromatic Noise Filtering

In general, pixels in images are represented by m -dimensional vectors, e.g., RGB values encoded as a 3-vector. Chromatic image noise affects only a pixel's chromaticity, i.e., the direction of the color vector but not its intensity. Here we restrict the filtering application to RGB images, therefore the chromaticity can be represented by unit vectors in \mathbb{R}^3 , which lies on the Grassmann manifold $\mathbf{G}_{3,1}$. The original mean shift has been used for the discontinuity preserving filtering of color images [2, 3]. This algorithm was extended to manifold-valued images in [18].

The image \mathbf{I} is considered to be a mapping on a d -dimensional lattice which assigns a value to each lattice point. Visual images typically have $d = 2$, although 3D images with $d = 3$ are also used. In this case $d = 2$ and at each location $\mathbf{z}_i = [x_i, y_i]^\top$, the data values $\mathbf{I}(\mathbf{z}_i)$ are assumed to lie on



Fig. 3 *Chromatic Noise Filtering. Mean Shift over $\mathbb{R}^2 \times \mathbf{G}_{3,1}$.* The *peppers* image corrupted with chromatic noise is shown on the left. The results of using standard mean shift filtering with EDISON are in the middle and the result of nonlinear mean shift filtering is on the right.

$\mathbf{G}_{3,1}$. A pixel $\mathbf{I}(\mathbf{z}_i)$ along with its location \mathbf{z}_i is considered as a single data point $\mathbf{x}_i = (\mathbf{z}_i, \mathbf{I}(\mathbf{z}_i))$, in the *joint domain* $\mathbb{R}^2 \times \mathbf{G}_{3,1}$.

The mean shift iterations are performed in this joint space to cluster the pixels. Consider an iteration starting at the point $\mathbf{x}_i = (\mathbf{z}_i, \mathbf{c}_i)$, where $\mathbf{c}_i = \mathbf{I}(\mathbf{z}_i)$, that converges to the mode $(\hat{\mathbf{z}}_i, \hat{\mathbf{c}}_i)$. In the filtered image $\hat{\mathbf{I}}$, all the pixel values converging to this mode are set to $\hat{\mathbf{c}}_i$. The profile in the joint domain is the product of a *spatial profile* defined on \mathbb{R}^2 and a *parameter profile* defined on the Grassmann manifold $\mathbf{G}_{3,1}$, as

$$\kappa(\mathbf{x}, \mathbf{x}_i) = \kappa_s \left(\frac{\|\mathbf{z} - \mathbf{z}_i\|^2}{h_s^2} \right) \kappa_p \left(\frac{d^2(\mathbf{c}, \mathbf{c}_i)}{h_p^2} \right). \quad (26)$$

A truncated normal kernel was used for both κ_s and κ_p . The bandwidth in the joint domain consists of a one-dimensional spatial bandwidth h_s and a one-dimensional parameter bandwidth h_p . The bandwidths h_s and h_p can be varied by the user to achieve the desired quality in the output.

Subbarao and Meer [19] showed that chromatic filtering using Grassmann manifolds leads to remarkable improvements over the original mean shift which smooths both intensity and color. The filtering results for the 512×512 *peppers* image are shown in Figure 3. The image is corrupted with chromatic noise by adding Gaussian noise with standard deviation $\sigma = 0.3$, along tangent directions followed by an exponential map on to $\mathbf{G}_{3,1}$. The original mean shift image filtering algorithm from EDISON, was executed with carefully selected bandwidth parameters (spatial $h_s = 11.0$ and color $h_p = 10.5$), to obtain the middle image. Using larger h_p led to oversmoothing and using smaller values did not denoise the original noisy image sufficiently. The nonlinear mean shift was executed with the same $h_s = 11.0$ but with $h_p = 0.5$ to obtain the image on the right. The nonlinear chromatic filtering is clearly better than EDISON due to the smoothing of the correct noise model.

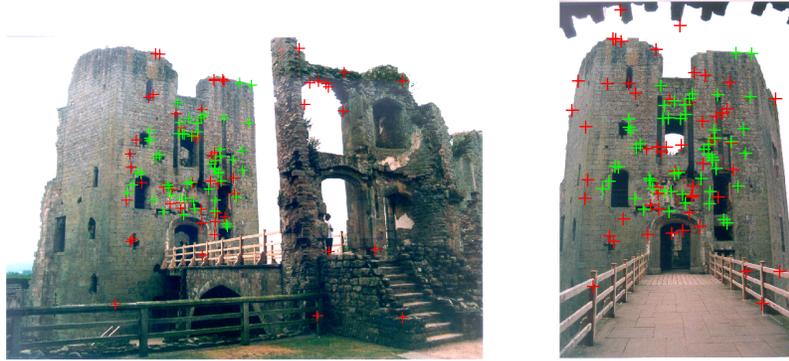


Fig. 4 Two images from the *Raglan Castle* sequence. The true inliers are marked with green markers while the outliers with red markers. The viewpoints of the two images are very different.

4.2 Fundamental Matrix Estimation

Reliable estimation of the fundamental matrix is often crucial for multi-view vision systems. Typically, in robust estimation formulations, the 3×3 fundamental matrix is represented by $\boldsymbol{\theta} \in \mathbb{R}^8$ while $\alpha \in \mathbb{R}$. Each data point is a vector of variables $\mathbf{y} = [x \ y \ x' \ y']^\top$ and lies in \mathbb{R}^4 . Here, (x, y) and (x', y') 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) can be written as

$$[x' \ y' \ 1] \mathbf{F}_{3 \times 3} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0. \quad (27)$$

The carrier vector is written as $\mathbf{x} = [x \ y \ x' \ y' \ xx' \ xy' \ yx' \ yy']^\top$ 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

$$\mathbf{J}_{\mathbf{x}|\mathbf{y}} = \begin{bmatrix} 1 & 0 & 0 & 0 & x' & y' & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & x' & y' \\ 0 & 0 & 1 & 0 & x & 0 & y & 0 \\ 0 & 0 & 0 & 1 & 0 & x & 0 & y \end{bmatrix} = [\mathbf{I}_{4 \times 4} \ \mathbf{J}(\mathbf{y})] \quad (28)$$

$$\mathbf{C}_{\mathbf{x}} = \sigma^2 \mathbf{J}_{\mathbf{x}|\mathbf{y}}^\top \mathbf{I}_{4 \times 4} \mathbf{J}_{\mathbf{x}|\mathbf{y}} = \sigma^2 \begin{bmatrix} \mathbf{I}_{4 \times 4} & \mathbf{J}(\mathbf{y}) \\ \mathbf{J}(\mathbf{y})^\top & \mathbf{J}(\mathbf{y})^\top \mathbf{J}(\mathbf{y}) \end{bmatrix}. \quad (29)$$

Raglan Castle Images: The gpbM algorithm was used to estimate the fundamental matrix between the *Raglan Castle* image pair shown in Fig. 4. Notice the large viewpoint change between the left and the right images.



Fig. 5 Two images of the *Graffiti* dataset. The true inliers are shown with green markers while the outliers are in red.

Using the SIFT algorithm [11], 109 point matches were obtained, out of which 54 were true inliers. With the values of $M = 400$ and $N = 200$, the performance of the gpbM algorithm was compared over 50 runs with and without the optimization on Grassmann manifold. On average, the gpbM algorithm misclassified 7.1 (out of 109) points, while only 5.6 points were classified wrongly after using the conjugate gradient algorithm. The average absolute residual error for the 54 true inlier points using gpbM algorithm was 1.86 pixels, while it was 1.77 pixels when optimization using conjugate gradient algorithm was also performed.

4.3 Planar Homography Estimation

A planar homography is a general 2D mapping between corresponding points on two projective planes. A pair of inlier homogeneous point correspondences \mathbf{p} and \mathbf{p}' , represented in homogeneous coordinates, satisfy $\mathbf{p}' = \mathbf{H}\mathbf{p}$. Using the Direct Linear Transformation (DLT) [7, Alg. 7.1], the same equation can be re-written as

$$\mathbf{A}_i \mathbf{h} = \begin{bmatrix} -\mathbf{p}_i^\top & \mathbf{0}_3^\top & x'_i \mathbf{p}_i^\top \\ \mathbf{0}_3^\top & -\mathbf{p}_i^\top & y'_i \mathbf{p}_i^\top \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix} = \mathbf{0}_2 \quad i = 1, \dots, n_1 \quad (30)$$

where $\mathbf{p}_i = [x_i \ y_i \ 1]^\top$ and $\mathbf{p}'_i = [x'_i \ y'_i \ 1]^\top$ are obtained from the image point correspondences of the n_1 inliers. The parameter vector $\boldsymbol{\theta}$ is obtained by rearranging the 3×3 homography matrix $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3]^\top$. The variables are $\mathbf{y}_i = [x_i \ y_i \ x'_i \ y'_i]^\top$ and \mathbf{A}_i is the 2×9 carrier matrix. The rows of \mathbf{A}_i correspond to the *two* carrier vectors $\mathbf{x}_i^{[1]}, \mathbf{x}_i^{[2]} \in \mathbb{R}^9$ obtained from each point correspondence and are heteroscedastic due to multiplicative terms.

Given four point correspondences across two planes, the 8×9 data matrix is formed by stacking the corresponding four carrier matrices. For a point

Sequence	Ext-MS [19]	Int-MS [1]	gpbM [12]	gpbM+CG [13]
arm(2)	30.65%	27.73%	7.99%	7.79%
articulated(3)	30.17%	24.50%	6.90%	6.70%
cars1(2)	20.07%	23.00%	6.51%	5.96%
cars2(2)	11.90%	9.08%	3.58%	3.55%
cars4(2)	21.60%	11.94%	7.55%	7.31%
cars5(3)	19.94%	19.41%	8.93%	8.05%
cars6(2)	5.68%	7.09%	1.86%	1.85%
cars8(2)	42.71%	35.29%	7.31%	6.97%
truck1(2)	28.56%	13.24%	6.27%	6.09%
2RT3RC(3)	12.52%	7.40%	10.92%	10.06%
Overall	17.91%	14.64%	6.58%	6.18%

Table 1 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. The results of gpbM and gpbM+CG were averaged over 50 runs while those of Ext-MS and Int-MS were averaged over 20 runs.

manifolds (gpbM+CG) [13]. The input data are the point matches across the F frames and the performance is compared using percentage of misclassified points.

The algorithms are tested on ten video sequences containing multiple motions from *Hopkins155* dataset. This dataset does not contain outliers. For each sequence, $F = 5$ frames were used by picking every sixth or seventh frame from the sequence. For gpbM with and without using conjugate gradient on Grassmann manifolds, the values of $M = 1000$ and $N = 200$ were used. The Ext-MS algorithm used 1000 randomly generated elemental subset based hypotheses. The corresponding parameter matrices were clustered on the Grassmann manifold $\mathbf{G}_{10,3}$ using the algorithm described in Section 3.1 with the bandwidth parameter h set to 0.1. The algorithm Int-MS used 1000 hypotheses generated as in the Ext-MS case.

Table 1 shows the comparative performance based on the percentage misclassification error. The results of the gpbM and gpbM+CG algorithms were averaged over 50 independent runs, while those of Ext-MS and the Int-MS were averaged over 20 independent runs. Neither of the methods assumed the knowledge of the true number of motions, but the nonlinear mean shift algorithms need the bandwidths as input from the user. It is clear that the optimization over Grassmann manifolds improves the estimates for every sequence.

5 Discussion

The image formation process imposes constraints on the imaged objects, e.g. the trajectory of points on a moving rigid body in a video sequence, hu-

man joint angle trajectories, shape of planar regions across multiple views, etc. Many computer vision techniques assume the pinhole camera model for image formation, which allows us to interpret the physical constraints as data points lying in an unknown linear subspace. The subspace parameters characterize physical world phenomena like camera rotation, object motion, human activity, etc. An orthonormal matrix $\mathbf{X}_{n \times k}$, lying on the Grassmann manifold can uniquely represent a linear subspace. Therefore, the geometry of Grassmann manifolds provides a natural mathematical framework to design subspace estimation or tracking algorithms for several computer vision applications.

The nonlinear mean shift on Grassmann manifolds takes a clustering based approach to solve the robust subspace estimation problem. For the applications that we presented here, both the Ext-MS [19] and Int-MS [1] produced comparable results. However, these methods can only be used when the user provides a correct bandwidth parameter. A new research direction could be to make these algorithms fully automatic by employing purely data-driven techniques for estimating the bandwidth. In [19], many other Riemannian manifolds were analyzed - Lie groups, the space of symmetric positive definite matrices and the essential manifold, which is a composite Lie group formed by the product space of two special orthogonal ($\mathbf{SO}(3)$) groups. Clustering-based approaches on these manifolds were used to solve a variety of computer vision problems.

We also discussed the generalized projection based M-estimator (gpbM), which benefited from optimization over Grassmann manifolds. The first step of gpbM estimates an initial set of inliers and the scale of inlier noise using a discrete set of elemental subset-based hypotheses. The model estimation step optimizes a robust objective function over a smaller set of hypotheses generated only from the initial inlier set. To further refine the estimate, the optimal hypotheses are used to initialize the conjugate gradient algorithm over the Grassmann manifold, which is a continuous space of subspace parameters. It remains an open question if, after the scale estimation step, the Grassmann manifold can directly be used to estimate the model and the correct inliers. This would require a reformulation of the problem definition.

The gpbM algorithm has been used in many applications to recover multiple structures by iteratively removing inlier points associated with each structure [12]. Without any prior knowledge about the scale of inlier noise, it is difficult for the robust estimation algorithm to recover more than a few structures automatically, which seems to be an easy task for the human vision system. We believe that the capabilities of automatic estimation algorithms can be significantly improved if top-down information is effectively incorporated into existing data-driven techniques.

Data analysis on manifolds involves higher computation to account for the non-Euclidean geometry. These problems have been somewhat mitigated with availability of parallel processing hardware and development of efficient new algorithms. For example, the Grassmannian Robust Adaptive Sub-

space Tracking Algorithm (GRASTA) [9] uses stochastic gradient descent on Grassmann manifolds to move along the geodesic with efficient rank one updates. GRASTA was applied to foreground/background segmentation in videos achieving a frame rate of about 46 fps. Such algorithmic advances have bridged the gap between the theoretical research and practical applications to large-scale computer vision problems that are of interest to researchers and practitioners alike.

The computational complexity of solutions may be higher for problems that incorporate the underlying Riemannian geometry, but it is important to avoid heuristics and ad-hoc approximations incurred due to solutions designed in Euclidean space. There is only a small amount of research work that has used Riemannian geometry to make significant impact in computer vision applications. Computer vision researchers could benefit from a course on Riemannian geometry with a special focus on its applications to computer vision problems. We hope that this book will serve as a catalyst to push forward computer vision research in this direction in order to understand and exploit the vast mathematical infrastructure of Riemannian geometry.

References

- [1] H. Cetingul and R. Vidal. Intrinsic mean shift for clustering on Stiefel and Grassmann manifolds. In *CVPR*, pages 1896–1902, 2009.
- [2] C. M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *ICPR*, pages 150–155, 2002.
- [3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.
- [4] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, apr 1999.
- [5] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [6] R. Hartley, J. Trunpf, Y. Dai, and H. Li. Rotation averaging. *Int. J. Comput. Vision*, 103(3):267–305, 2013.
- [7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
- [8] S. Hauberg, A. Feragen, and M. J. Black. Grassmann averages for scalable robust PCA. In *CVPR*, pages 3810–3817, 2014.
- [9] J. He, L. Balzano, and A. Szelam. Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video. In *CVPR*, pages 1568–1575, 2012.
- [10] S. Jain and V. Govindu. Efficient higher-order clustering on the Grassmann manifold. In *ICCV*, pages 3511–3518, 2013.

- [11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.
- [12] S. Mittal, S. Anand, and P. Meer. Generalized projection-based M-estimator. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(12):2351–2364, 2012.
- [13] S. Mittal and P. Meer. Conjugate gradient on Grassmann manifolds for robust subspace estimation. *Image Vision Comput.*, 30(6-7):417–427, 2012.
- [14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C. The art of scientific computing*. Cambridge: University Press, second edition, 1992.
- [15] R. Raguram, J.-M. Frahm, and M. Pollefeys. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In *ECCV*, pages 500–513. 2008.
- [16] A. Shrivastava, S. Shekhar, and V. M. Patel. Unsupervised domain adaptation using parallel transport on Grassmann manifold. In *WACV*, pages 277–284, 2014.
- [17] R. Subbarao and P. Meer. Beyond RANSAC: User independent robust regression. *IEEE CVPR Workshop on 25 years of RANSAC*, page 101, 2006.
- [18] R. Subbarao and P. Meer. Discontinuity preserving filtering over analytic manifolds. In *CVPR*, pages 1–6, 2007.
- [19] R. Subbarao and P. Meer. Nonlinear mean shift over Riemannian manifolds. *Int. J. Comput. Vision*, 84(1):1–20, 2009.
- [20] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(11):2273–2286, 2011.
- [21] J. Xu, V. Ithapu, L. Mukherjee, J. Rehg, and V. Singh. GOSUS: Grassmannian online subspace updates with structured-sparsity. In *ICCV*, pages 3376–3383, 2013.
- [22] L. Yang. Riemannian median and its estimation. *LMS J. of Computation and Math.*, 13:461–479, 12 2010.