Affine invariance revisited

Evgeni Begelfor Michael Werman
School of Engineering and Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
{aristo,werman}@cs.huji.ac.il

Abstract

This paper proposes a Riemannian geometric framework to compute averages and distributions of point configurations so that different configurations up to affine transformations are considered to be the same. The algorithms are fast and proven to be robust both theoretically and empirically. The utility of this framework is shown in a number of affine invariant clustering algorithms on image point data.

1. Introduction

Objects are often known up to some ambiguity, depending on the methods used to acquire them. The first-order approximation to any transformation is, by definition, affine, and the affine approximation to changes between images has been used often in computer vision [8]. Thus it is beneficial to deal with objects known only up to an affine transformation. For example, feature points on a planar transform projectively between different views, and the projective transformation can in many cases be approximated by an affine transformation. Likewise, color values of pixels vary close to affinely with change of illumination[10]. A number of popular clustering algorithms, such as k-means, mean shift and EM use averaging on the data and probability distributions. In order to use these algorithms we need methods to measure distances between such sets, to compute means and to put probability distributions on them.

Previous work treating affine invariance in computer vision can be generally divided into two approaches: invariants and normalization. The first method consists in computing different functions (invariants) of a set of points that are invariant to the relevant group of transformations [11]. The disadvantages of this approach is the difficulty of using invariants to define a meaningful distance between configurations of feature points, moreover, a full set of invariants is needed in order to distinguish between different sets. Also, averaging the invariants is not the way to average different configurations, as the averaged invariants don't necessarily correspond to a configuration.

One popular form of normalization, originating in statistics, is whitening, in which an affine transformation is applied to set of points such that they have zero average and the identity covariance matrix. The problem with this method is that there are remaining degrees of freedom, that is, two affinely equivalent sets can have different normal forms. Another normalization, obtained by bringing pivot points in the configuration to a standard location suffers from being arbitrary and thus highly sensitive to noise. (Note that the non-pivot points after the transformation are invariants).

Affine-invariant distance between sets of points in 2D has been suggested in [15], but, as above, having a distance does not allow us to compute means and probabilities.

The study of the space of ordered configurations of n points in $\mathbb{R}^k$ up to similarity transformations was pioneered by Kendall (see [9]), who coined the name *shape space*. For different groups of transformations (rigid, similarity, linear, affine, projective) one obtains different shape spaces. Shape spaces were considered in [3],[14], although no attempts to give a geometric structure on the shape space were made. Methods similar to ours were suggested [13] in order to morph between affine shapes, although mistakenly only the linear invariance was used (see Appendix).

Our approach is to define a canonic geometric structure on the affine shape space and use general geometric methods [12] to compute averages and distributions of affine-invariant point configurations. Thus the clustering algorithms mentioned above can be implemented on the affine shape space.

The paper is organized as follows: the next section provides the mathematical background in geometry and Section 3 explains how to use the geometrical methods on the affine shape space. We proceed with possible applications of the approach and results therein, finishing with a discussion and suggestions for future work.

2. Mathematical Background

We are going to map every n-point configuration to a single point in a certain manifold, on which we can compute distances and averages. To enable further discussion, we need to take a small detour into Riemannian geometry. Further information can be found in any textbook on the subject, such as [6].

An m-dimensional manifold is a space that locally looks like $\mathbb{R}^m$. A differential manifold M enables us to talk about derivatives of curves on the manifold, the derivative of a curve $\gamma(t)$ at a point $x \in M$ being a vector $\gamma'(t)$ lying in the vector space T_xM which is called the tangent space to M at x. A *Riemannian* manifold is a differential manifold with an inner product $<,>_x$ uniquely defined on each tangent space T_xM. With the inner product, any differentiable curve $\gamma : [a,b] \to M$ has length $L(\gamma)$ defined as

$$L(\gamma) = \int_a^b \|\gamma'(t)\|_{\gamma(t)} dt \tag{1}$$

where $\|v\|_x = \sqrt{<v,v>_x}$ is the norm on T_xM derived from the inner product. The distance $d(x,y)$ between two points on the manifold is the infimum of the lengths of paths between them. A path that minimizes the distance between two nearby points is called a geodesic. While it is not always true, in a generic situation for every point x in M there is a unique geodesic starting from x in every direction, giving us the *exponential* map (the usual exponential is a special case) $\exp_x : T_xM \to M$ such that $d(x, exp_x(v)) = \|v\|_x$ for every v in T_xM. The inverse map to $\exp_x$, the logarithm is defined only in a certain neighborhood of x and is denoted by $\log_x$.

The notion of the mean of a set of points in M can be defined in different ways. One of them, called the Karcher mean [5](although it was originally defined and studied by Cartan) comes from noticing that the mean of a set of points in the euclidean space $\mathbb{R}^k$ minimizes the sum of the squared distances to the points in the set. This still makes sense in any Riemannian manifold, thus we define:

$$Mean(x_1, \ldots, x_n) = \arg\min_{y \in M} \sum_{i=1}^{n} d(y, x_i)^2 \tag{2}$$

The mean is not unique for a general set of points, consider two antipodal points on a sphere. Nevertheless, it is known [2] that for points lying close enough to each other the mean is unique and in addition the equation above has unique local minimum (See Appendix for further discussion). By differentiating we get that y is the mean if and only if

$$\sum_{i=1}^{n} \log_y x_i = 0 \tag{3}$$

This gives us a simple gradient descent algorithm for finding the mean.

Using the mean defined in the above paragraph we can estimate the expectation of an empirical distribution on a manifold. We can go one step further and compute the covariance matrix of the data. Let $x_1, \ldots, x_n \in M$ and $\mu = Mean(x_1, \ldots, x_n)$. Fix a basis $v_1, \ldots, v_n$ of $T_\mu M$. The covariance matrix Σ relative to the basis $v_1, \ldots, v_n$ is

$$\Sigma = \frac{1}{n} \sum_{i,j} \log_\mu(x_i) \log_\mu(x_j)^T \tag{4}$$

where $log_\mu(x_i)$ is written in the basis $\{v_k\}$. We then get a "normal" distribution $N(\mu, \Sigma)$ on M fitting our data with density

$$\phi(x) \propto e^{-\frac{1}{2} \log_\mu(x) \Sigma^{-1} \log_\mu(x)^T} \tag{5}$$

The exponent of the density is the Mahalanobis distance between x and the mean of the distribution $N(\mu, \Sigma)$

$$d(x) = \log_\mu(x) \Sigma^{-1} \log_\mu(x)^T \tag{6}$$

If M_1, M_2 are Riemannian manifolds then there is a natural Riemannian structure on the product manifold $M = M_1 \times M_2$: if (x_1, x_2) is a point in M and $v, w \in T_xM$ then $v = (v_1, v_2)$ and $w = (w_1, w_2)$. We define

$$< v, w >_x = < v_1, w_1 >_{x_1} + < v_2, w_2 >_{x_2} \tag{7}$$

One easily shows that geodesics in M are product of geodesics in M_1 and M_2: $\gamma(t)$ is a geodesic in M iff $\gamma(t) = (\gamma_1(t), \gamma_2(t))$ where γ_i is a geodesic in M_i. In the same way, means in M can be computed coordinatewise.

3. Geometry of the affine shape space

Our goal is to define a geometric structure on the set of n-point configurations in $\mathbb{R}^k$, where two configurations $(v_1, \ldots, v_n)$ and $(u_1, \ldots, u_n)$ are considered equivalent if there is an affine transformation from one to another:

$$u_i = Av_i + b \tag{8}$$

To achieve the goal, we need to assign a representative to every configuration such that equivalent configurations get the same representative. Given a configuration $v_1, \ldots, v_n$ we look at the subspace V spanned by the columns of the following matrix:

$$M(v_1, \ldots, v_n) = \begin{pmatrix} v_1^T & 1 \\ v_2^T & 1 \\ \ldots & 1 \\ v_n^T & 1 \end{pmatrix} \tag{9}$$

In other words, V is the image of the operator $M(v_1, \ldots, v_n)$. We show that V is invariant to any affine transformation applied to $v_1, \ldots, v_n$. If $u_i = Av_i + b$ then

$$M(u_1, \ldots, u_n) = \begin{pmatrix} v_1^T A^T + b^T & 1 \\ v_2^T A^T + b^T & 1 \\ \ldots & 1 \\ v_n^T A^T + b^T & 1 \end{pmatrix} = \tag{10}$$

$$= \begin{pmatrix} v_1^T & 1 \\ v_2^T & 1 \\ \dots & 1 \\ v_n^T & 1 \end{pmatrix} \begin{pmatrix} A^T & 0 \\ b^T & 1 \end{pmatrix} \tag{11}$$

As $\begin{pmatrix} A^T & 0 \\ b^T & 1 \end{pmatrix}$ is invertible, $M(u_1, \dots, u_n)$ and $M(v_1, \dots, v_n)$ have the same image. On the other hand, if the image of $M = M(v_1, \dots, v_n)$ is equal to the image of $M' = M(u_1, \dots, u_n)$ then there is a $k+1 \times k+1$ matrix B such that $M' = MB$ and this B has to be affine, so $\{v_i\}$ and $\{u_i\}$ are affinely equivalent.

Thus every n-point configuration in $\mathbb{R}^k$ gives rise to a $k+1$ dimensional subspace of $\mathbb{R}^n$, with equivalent configurations giving the same subspace. Of course, the configuration can be reconstructed from the subspace only up to affine equivalence. Note, however, that not every $k+1$ dimensional subspace of $\mathbb{R}^n$ is the representation of an n point configuration, only those containing the vector $\vec{1} = (1, 1, \dots, 1)^T$. Thus our representative will be the orthogonal complement of $\vec{1}$ in V. Summarizing: the space of affine shapes is the space of all k-dimensional subspaces in $\vec{1}^{\perp}$.

The space of k-dimensional subspaces of $\mathbb{R}^n$ is called the Grassman manifold and denoted by $G(k, n)$. It is a generalization of the notion of projective space, which is the space of all 1-dimensional subspaces of $\mathbb{R}^n$. The are different ways to define distances in $G(k, n)$, and in particular, Riemannian structure. Nevertheless, there is a unique (up to scale) Riemannian structure that is invariant to the action of the orthogonal group on the left (see Appendix), thus becoming invariant to permutations (of the points), so that $d((v_1 \dots v_n), (w_1 \dots w_n)) = d((v_{\pi(1)} \dots v_{\pi(n)}), (w_{\pi(1)} \dots w_{\pi(n)})$ for any permutation π. The geometry of $G(k, n)$ with this metric has been studied [16], [17] and algorithmic methods for solving problems on the Grassman manifolds have been suggested in [4] [7],[1]. For completeness we give the algorithms for computing the distance, exponent and logarithm on Grassman manifolds equipped with this metric.

We represent a k-dimensional subspace W of $\mathbb{R}^n$ by any $n \times k$ matrix A whose columns span W. Clearly, for any nonsingular $k \times k$ matrix P the subspace spanned by AP is identical to the one spanned by A. Thus, the dimension of $G(k, n)$ is $k \cdot n - k \cdot k = k(n-k)$. Notice that we are not using any canonical coordinates for the Grassman manifold, such as Plucker coordinates, but work with any matrix spanning the subspace.

Let X,Y be orthogonal $n \times k$ matrices representing subspaces W and W'. Recall that $U\Sigma V^T = A$ is a *thin* SVD decomposition of A if U is $n \times k$ orthogonal, Σ is $k \times k$ diagonal and V is $k \times k$ orthogonal. Now the distance, exponent and logarithm on the Grassman manifold can be computed using standard mathematical tools:

Algorithm 1. Distance=d(X,Y)

$U\Sigma V^T = $ thin [1]SVD(X^TY)
$\Theta = \cos^{-1}\Sigma$
$d(X, Y) = \sqrt{\sum_i \theta_i^2}$

Algorithm 2. $Gexp(X, H)$

$U\Sigma V^T = $ thin SVD(H)
$Gexp(X, H) = XV\cos\Sigma + U\sin\Sigma$

Algorithm 3. $Glog(X, Y)$

$U\Sigma V^T = $ thin SVD$((I - XX^T)Y(X^TY)^{-1})$
$\Theta = \tan^{-1}\Sigma$
$Glog(X, Y) = U\Theta V^T$

Using the functions above, we can write down the algorithm for computing the mean of a set of configurations $M_1, \dots, M_N$ where each M_i is the matrix built from the points of $i-$th configuration as described in the beginning of the section.

Algorithm 4. Mean $(A_1, \dots, A_N)$

Choose any orthogonal basis $w_1, \dots, w_{n-1}$ for $1^{\perp}$.
$P = \begin{pmatrix} w_1 \\ \dots \\ w_{n-1} \end{pmatrix}$
for $j = 1$ to N **do**
  $R_j = PA_j$
  $U_jD_jV_j^T = $ thin SVD(R_j)
$\mu = U_1$
repeat
  $\delta = \frac{1}{N}\sum_{j=1}^{N} Glog(\mu, U_j)$
  $\mu = Gexp(\mu, \delta)$
until $\|\delta\| < \epsilon$
Mean$= P^T\mu$

The convergence issues are handled in the Appendix, where we show that the algorithm works in a very general setting. In [1] a different definition of a mean on the Grassman manifold is given together with a faster algorithm for computing it. Unfortunately, nothing is known about the algorithm's convergence.

A tangent vector v in the tangent space T_W is represented by matrix H such that $A^TH = 0$. Notice that this representation of T_W doesn't depend on the choice of A, as

$$A^TH = 0 \iff (AP)^TH = P^TA^TH = 0 \tag{12}$$

for any nonsingular P. To estimate the covariance matrix of an empirical distribution $x_1, \dots, x_n$ with mean μ we need to pick a basis $v_1, \dots, v_{n(n-k)}$ for T_μ. If Y represents μ and $Y = U\Sigma V^T$ is the full SVD decomposition, then the first k columns of U span μ, while the last $n-k$ columns

are an orthogonal basis to T_μ. For any matrix H such that $Y^T H = 0$ (that is, representing a vector in T_μ) the matrix $U^T H$ will have zeros in the top k rows. The rest of $U^T H$, rewritten in a vector form $c_U(H)$ is the representation of H in our basis of T_μ. Summarizing we obtain the algorithm for computing covariance:

Algorithm 5. Covariance= $\Sigma(A_1, \ldots, A_N)$

$\mu = \text{Mean}(A_1, \ldots, A_N)$
$UDV^T = \text{SVD}(\mu)$
$\forall i \; v_i = c_U(Glog(\mu, A_i))$
$\Sigma = \frac{1}{N} \sum_i v_i v_i^T$

Now the computation of the Mahalanobis distance and the density of the normal distribution is as in Section 2.

In some cases, as in some models for illumination invariance, we want to consider configurations up to linear transformations, instead of affine. This case is even simpler and is treated in the Appendix.

As remarked in the end of Section 2, a product of Riemannian spaces is a Riemannian space. This can be used in the case of colored points, where the points are transformed affinely in the space, and the colors transform affinely in the RGB space.

4. Applications and Results

Being able to compute means, we can now employ clustering algorithms that use averaging, such as k-means and mean-shift. If the data is sufficient, Mixture of Gaussians with EM can be used as well, applying the methods above to estimate the covariance of a cluster. A number of synthetic and real examples follow.

We give a short description of the well-known algorithms used for clustering: k-means, mixture of Gaussians and mean shift.

Algorithm 6. k-Means($x_1, \ldots, x_N$)

Pick k points $c_1, \ldots, c_k$ uniformly from $x_1, \ldots, x_N$
repeat
  $\forall j \quad C_j = \emptyset$
  for $i = 1$ to N **do**
    Find j minimizing $d(x_i, c_j)$
    $C_j \Leftarrow C_j \cup \{x_i\}$
  for $j = 1$ to k **do**
    $c_j = Mean(C_j)$
until there are no updates

Notice that these algorithms can be implemented in any data space as long as we can compute distance, mean and covariance (for EM) in that space. As we have seen above, the affine shape space, which is effectively a Grassman manifold, has efficient algorithms for computing these quantities. We shall denote by k-means, EM and Mean-shift the algorithms above using euclidean space after normalization of the configurations. On the other hand, G-k-means, G-EM, G-Meanshift use the Grassman geometry. To show the need for mean computing clustering algorithms, we compare the results with the average link clustering of the distance graph. This algorithm will be denoted by *link* or *G-link* when using the euclidean distance after normalization or Grassman distance, accordingly.

Algorithm 7. Mean shift($x_1, \ldots, x_N$)

for $iter = 1$ to IterNum **do**
  for $i = 1$ to N **do**
    Let A be r closest neighbors of x_i
    $x_i = Mean(A)$
Run average-link clustering

Algorithm 8. Mixture of Gaussians($x_1, \ldots, x_N$)

Choose random k points c_j from $x_1, \ldots, x_N$
$\forall j \quad C_j \Leftarrow \emptyset$
$\forall j \Sigma_j \Leftarrow I$
repeat
  for $i = 1$ to N **do**
    Find j minimizing $d_{c_j, \Sigma_j}(x_i)$ {w}ith $d_{c_j, \Sigma_j}(x_i)$ the Mahalanobis distance from x_i to c_j with covariance Σ_j
    $C_j \Leftarrow C_j \cup \{x_i\}$
  for $j = 1$ to k **do**
    $c_j = Mean(C_j)$
    $\Sigma_j = Covariance(C_j)$
until

As the ground truth clustering $\tilde{F} : \{1, \ldots, N\} \to \{1, \ldots, k\}$ in the experiments is known, we use the misclassification ratio of clustering F to evaluate the performance. As the order of the labels is meaningless, we define the misclassification ratio as the minimum of error percentage for all possible orderings of the labels.

$$\alpha(F) = \frac{1}{N} \min_{\pi \in S_k} \sum_c |F^{-1}(c) \setminus \tilde{F}^{-1}(\pi(c))| \qquad (13)$$

We begin with a synthetic example demonstrating the usefulness of the proposed approach. Two schematic drawings of a man defined by a set of points were perturbed by random affine transformations and random noise, resulting in a set of drawings in Figure 1. The goal was to partition the set of drawings into two clusters. The comparison of the misclassification relative to ground truth is shown in table 1.

Algorithm	linkage	k-means	meansift
Normalization	7	6	4
Grassman	7	2	1

Table 1. Classification error for clustering of the dancing stick figures. The smaller the number, the better the result.

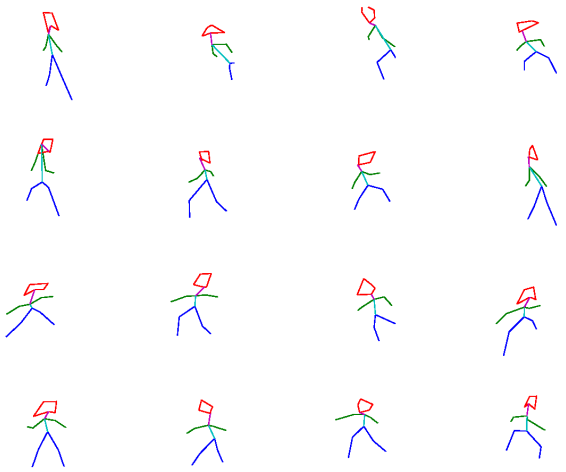

Figure 1. Dancing men: Two "stick" drawings of a person were randomly perturbed. The task is to cluster the images into two groups. The ground truth: the first and the third column are perturbations of one figure and second and fourth of the other.

We now proceed to a systematic evaluation of the approach. 4 sets of 10 points in the plane were drawn from $N(0,1)$. Each one of the sets was cloned 25 times. A different random affine transformation was applied to every one of 100 sets, and normally distributed noise with standard deviation σ was added. Our goal is to cluster the resulting 100 sets. Figure 5 shows the average performance of different algorithms.

Real-life example. We took a set of images of f15 and f18 airplanes from the web (Figure 5) and manually labeled 11 points on each image (the nose, corners of the wings and tail, etc.) as in Figure 2. The misclassification error of different algorithms is shown in Table 2.

Table 2. Classification error for clustering f15/f18 images

Algorithm	link	k-means	meansift
Normalization	6	6	6
Grassman	4	3	0

5. Discussion and future work

We showed that affine invariance can be treated robustly and efficiently using a Riemannian framework. Using this framework we showed how classic clustering algorithms can be adapted to the affine invariant case. We believe that these methods will have many uses in computer vision and image processing systems.

We plan to map other computer vision problems, such as projective invariance and continuous shape deformations to their relevant Riemannian manifolds in order to be able to carry out the analysis on the correct spaces.

Figure 2. A set 11 points were labeled on each airplane image

Appendix

A. Linear shape space

The structure of the linear shape space is similar the affine case, only simpler. Given a configuration $v_1, \ldots, v_n$ we look at the subspace V spanned by the columns of the following matrix:

$$M(v_1, \ldots, v_n) = \begin{pmatrix} v_1 \\ v_2 \\ \ldots \\ v_n \end{pmatrix} \tag{14}$$

In other words, V is the image of the operator $M(v_1, \ldots, v_n)$. We show that V is invariant to any linear transformation applied to $v_1, \ldots, v_n$. If $u_i = Av_i$ then

$$M(u_1, \ldots, u_n) = \begin{pmatrix} Av_1 \\ Av_2 \\ \ldots \\ Av_n \end{pmatrix} = \tag{15}$$

$$= \begin{pmatrix} v_1 \\ v_2 \\ \ldots \\ v_n \end{pmatrix} A^T \tag{16}$$

As A^T is invertible, $M(u_1, \ldots, u_n)$ and $M(v_1, \ldots, v_n)$ have the same image. On the other hand, if the image of $M = M(v_1, \ldots, v_n))$ is equal to the image of $M' = M(u_1, \ldots, u_n)$ then there is a $k \times k$ matrix A such that $M' = MA$, so $\{v_i\}$ and $\{u_i\}$ are linearly equivalent.

In this way every configuration gives rise to a k-dimensional subspace of $\mathbb{R}^n$. As opposed to the affine case, there are no technical complications and the shape space is just the Grassman manifold $G(k, n)$.

B. Convergence of the mean

A set S in a Riemannian manifold X is called convex if for any two points $x_1, x_2 \in S$ there is a unique shortest geodesic between x_1 and x_2 lying in S. Following [2], if the points on a manifold lie in a convex ball then $\sum_{i=1}^{n} d(y, x_i)^2$ has only one local (thus global) minimum ensuring the convergence of the mean finding algorithm. As the space is homogeneous, the convexity of a ball depends only on its radius. The convexity radius ConRad (the maximal radius of convex balls) obeys

$$ConRad \geq \min\{\frac{1}{2}InjRad, \frac{1}{2}K\} \tag{17}$$

where $InjRad$ is the injectivity radius (the radius of the biggest ball on which $\exp_x$ is injective) and K is a upper bound on the sectional curvature. According to [16] any geodesic in $G(k,n)$ with $\min\{k, n-k\} \geq 2$ that intersects itself is closed, and the minimal length of a closed geodesic is π. Thus the injectivity radius is $\frac{\pi}{2}$. By [17] the curvature of $G(n,k)$ is bounded by 4, thus the convexity radius is $\frac{\pi}{4}$. The diameter, that is, the maximal distance between two points, of $G(k,n)$ is equal to $\min\{\sqrt{k}, \sqrt{n-k}\}\frac{\pi}{2}$ for $k, n-k \neq 1$ [16]. In our applications $k = 2$ or 3, so we see that the algorithm for computing the mean will converge even for relatively spread out sets.

C. Noise sensitivity

A substantial advantage of our affine shape representation over normalization is its insensitivity to noise. Normalization suffers badly from noise, as a small change in one of the pivot points can lead to a large error in the position of the other points. The difference in the stability between the two methods is shown in Figure 3.

We analyze the robustness of the Grassman metric under Gaussian noise on the coordinates of the points. Let $u_1, ...u_n$ be a configuration of n points in the plane with zero mean and unit covariance. We shall assume additive independent Gaussian noise y_{ij} of zero mean and variance ϵ. Our goal is to compute the average distance between the configuration $\{u_i\}$ and $\{v_i = u_i + y_i\}$. Let A be the matrix $\begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{pmatrix}$. As the $\{u_i\}$ have zero mean and unit covariance, there is a orthogonal matrix O such that

$$OA = \begin{pmatrix} \sqrt{n} & 0 \\ 0 & \sqrt{n} \\ 0 & 0 \\ \dots & \\ 0 & 0 \end{pmatrix} \tag{18}$$

As argued before, the metric is invariant under $O_n(\mathbb{R})$, thus $d(A, A+Y) = d(OA, OA+OY)$. The coefficients of OY have the same distribution as the coefficients of Y, as O is orthonormal, so we can assume that A is $\begin{pmatrix} \sqrt{n}I \\ 0 \end{pmatrix}$.

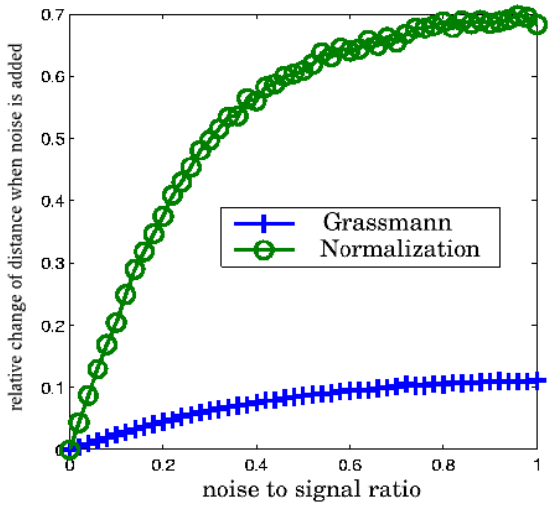

Figure 3. A and B are two sets of 6 points on the plane drawn from N(0,1). ΔA and ΔB represent noise drawn from $N(0, \sigma)$. The graph shows the average value of $D = \frac{|d(A+\Delta A, B+\Delta B) - d(A,B)|}{d(A,b)}$ for both the distance on the Grassman manifold and the euclidean distance after normalizing.

Let C and D be the matrices whose columns are the orthogonal bases of A and $A+Y$, accordingly. Write D as $\begin{pmatrix} F \\ G \end{pmatrix}$ when F is 2×2 matrix. It is easily seen that

$$d(A, A+y) \leq \sqrt{2}\cos^{-1}\|C^T D\| \tag{19}$$

As C is just $\begin{pmatrix} I \\ 0 \end{pmatrix}$ in our case, $C^T D = F$. To show that $d(A, A+Y)$ is small we need to show that the norm $\|F\|$ is close to 1. We know that D is orthogonal, thus maps every unit vector to a unit vector.

$$1 = \|D\binom{1}{0}\| = \sqrt{\|F\binom{1}{0}\|^2 + \|G\binom{1}{0}\|^2} \tag{20}$$

As the first column of D is obtained from the first column of $A+Y$ divided by $\sqrt{n}$ and the y_{ij} are independent, on average $\|G\begin{pmatrix} 1 \\ 0 \end{pmatrix}\| < \frac{\epsilon\sqrt{n}}{\sqrt{n}} = \epsilon$. We have

$$\|F\| \geq \|F\binom{1}{0}\| = \sqrt{1 - \|G\binom{1}{0}\|^2} \simeq 1 - \frac{\epsilon^2}{2} \tag{21}$$

and finally

$$d(A, A+Y) \leq \sqrt{2}\cos^{-1}\sqrt{1 - \|G\binom{1}{0}\|^2} \simeq \sqrt{2}\epsilon \tag{22}$$

D. Uniqueness of metric

Denote $G = SO_n(\mathbb{R})$ and $H = SO_k(\mathbb{R}) \times SO_{n-k}(\mathbb{R})$. We wish to show that there is a unique (up to scale) G-

invariant Riemannian metric on $G(k,n) = G/H$. Any invariant metric on G/H is uniquely determined by an inner product $<,>$ on the tangent space T_x, when $x = eH \in G/H$. In our case, x is the subspace spanned by $e_1, \ldots, e_k$ and T_x can be represented as the set of all matrices of size $(n-k) \times k$. Necessary and sufficient condition for the metric to be invariant is:

$$\forall h \in H \quad \forall W, Z \in T_x \;\; < w, z >=< hW, hZ > \tag{23}$$

If $h = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix}$ then

$$hW = \frac{d}{dt} h \begin{pmatrix} I \\ tW \end{pmatrix} = \frac{d}{dt} \begin{pmatrix} U \\ tVW \end{pmatrix} = \frac{d}{dt} \begin{pmatrix} I \\ tVWU^{-1} \end{pmatrix} = VWU^{-1} \tag{24}$$

Thus $<,>$ must be invariant to the action of $SO_{n-k}(\mathbb{R})$ on the right and $SO_k(\mathbb{R})$ in the left. Denote by e_{ij} the natural basis for T_x, the set of all $(n-k) \times k$ matrices. Assume $< e_{11}, e_{11} >= 1$. To show that $<,>$ is the standard inner product and thus unique we need:

$$\begin{aligned} \forall i,j \quad & < e_{ij}, e_{ij} >= 1 \\ \forall (i,j) \neq (k,l) \quad & < e_{ij}, e_{kl} >= 0 \end{aligned} \tag{25}$$

For every i, j there exist U, V such that $Ue_{ij}V = e_{11}$, thus $< e_{ij}, e_{ij} >= 1$. For every $i \neq k$ there is a U (rotation by $\pi/4$) such that for every j

$$Ue_{ij} = \frac{\sqrt{2}}{2}(e_{ij} + e_{kj}) \quad \text{and} \quad Ue_{kj} = \frac{\sqrt{2}}{2}(e_{ij} - e_{kj}) \tag{26}$$

In the same way, for every $j \neq l$ there is a V such that for every i

$$e_{ij}V = \frac{\sqrt{2}}{2}(e_{ij} + e_{il}) \quad \text{and} \quad e_{il}V = \frac{\sqrt{2}}{2}(e_{ij} - e_{il}) \tag{27}$$

Finally, for $i \neq k$ and $j \neq l$

$$< e_{ij}, e_{il} >= \frac{1}{2} < e_{ij} + e_{il}, e_{ij} - e_{il} >= 0 \tag{28}$$

$$< e_{ij}, e_{kj} >= \frac{1}{2} < e_{ij} + e_{kj}, e_{ij} - e_{kj} >= 0 \tag{29}$$

and

$$\begin{aligned} < e_{ij}, e_{kl} >= \frac{1}{2} < e_{ij} + e_{il}, e_{kj} - e_{kl} >= \\ \frac{1}{2}(< e_{ij}, e_{kj} > + < e_{il}, e_{kj} > - < e_{ij}, e_{kl} > - \\ < e_{il}, e_{kl} >) = \frac{1}{2}(< e_{il}, e_{kj} - < e_{ij}, e_{kl} >) = \\ \frac{1}{2}(< e_{ij}, e_{kl} > - < e_{ij}, e_{kl} >) = 0 \end{aligned} \tag{30}$$

References

[1] P. Absil, R.Mahony, and R. Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80:199–220, 2004. 3

[2] M. Berger. *A Panoramic View of Riemannian Geometry*. Springer, Berlin, 2003. 2, 6

[3] R. Berthilsson. A statistical theory of shape. In *SSPR/SPR*, pages 677–686, 1998. 1

[4] A. Bjork and G. Golub. Numerical methods for computing angles between linear subspaces. *Math. Comp*, 27:579–594, 1973. 3

[5] P. Buser and H. Karcher. Gromov's almost flat manifolds, Asterisque 1981. 2

[6] M. D. Carmo. *Riemannian Geometry*. Birkuser, Boston, 1992. 2

[7] A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality constrains. *SIAM J. Matrix Anal. Appl*, 20(2):303–353, 1998. 3

[8] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002. 1

[9] D. G. Kendall, D. Barden, T. K. Carne, and H. Le. *Shape and Shape Theory*. Wiley Series in Probability and Statistics, 1999. 1

[10] F. Mindru, L. J. V. Gool, and T. Moons. Model estimation for photometric changes of outdoor planar color surfaces caused by changes in illumination and viewpoint. In *ICPR*, volume 2, pages 620–623, 2002. 1

[11] J. L. Mundy and A. P. Zisserman, editors. *Geometric Invariance in Computer Vision*. MIT Press, Cambridge, MA, 1992. 1

[12] X. Pennec. Probabilities and statistics on riemannian manifolds: Basic tools for geometric measurements. In *Proc. of Nonlinear Signal and Image Processing*, pages 194–198, 1999. 1

[13] D. A. Sepiashvili, J. M. F. Moura, and V. H. S. Ha. Affine-permutation symmetry: Invariance and shape space. In *Proceedings of the IEEE Workshop on Statistical Signal Processing. Special Session on Statistical Inferences on Manifolds with Apllications in Image Analysis*, 2003. 1

[14] G. Spar. Structure and motion from kinetic depth. In *Proc. of the Sophus Lie International Workshop on Computer Vision and Applied Geometry, Mordfjordeid, Norway*, 1995. 1

[15] M. Werman and D.Weinshall. Similarity and affine invariant distances between 2d point sets. *IEEE Trans. Pattern Anal. Mach. Intell*, 17(8):810–814, 1995. 1

[16] Y. Wong. Differential geometry of grassmann manifolds. *Proc. Nat. Acad. Sci. U.S.A*, 57:589–594, 1967. 3, 6

[17] Y. C. Wong. Sectional curvatures of grassmann manifolds. *Proc. Natl. Acad. Sci. U.S.A.*, 60:75–79, 1968. 3, 6

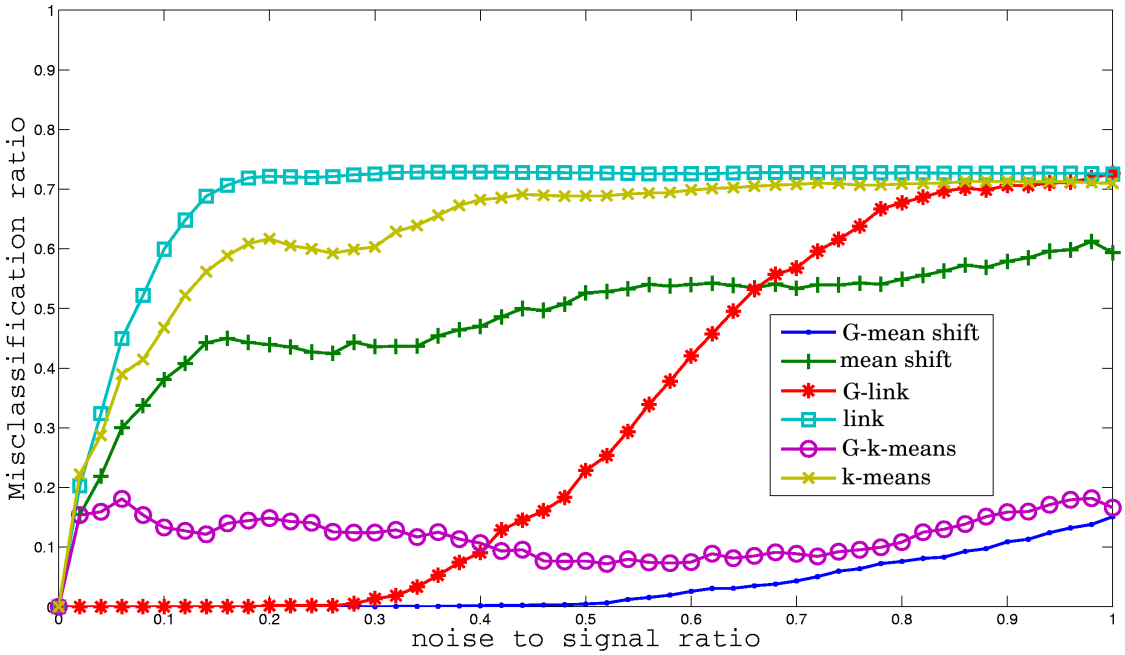

Figure 4. Comparison of the average classification error of six algorithms: link, G-link, k-means, G-k-means, meanshift and G-meanshift as a function of noise amplitude. As there were 4 clusters in the experiment, a misclassification ratio of 0.75 corresponds to the worst result possible. The plot clearly shows the advantage of Grassman based algorithms Notice that for small enough noise the average link algorithm using the Grassman distance performs well, although as the noise grows it gives way to the algorithms that use averaging.

Figure 5. A set of images of airplanes downloaded from the web. After labeling feature points, the images were successfully partitioned by the G-meanshift algorithm in 2 clusters, corresponding to f15 and f18. Algorithms using normalization failed to accomplish the task.