Finding the Subspace Mean or Median to Fit Your Need

Tim Marrinan J. Ross Beveridge Bruce Draper Michael Kirby Chris Peterson
Colorado State University
1873 Campus Delivery
Fort Collins, CO 80523
{marrinan, kirby, peterson}@math.colostate.edu {ross, draper}@cs.colostate.edu

Abstract

Many computer vision algorithms employ subspace models to represent data. Many of these approaches benefit from the ability to create an average or prototype for a set of subspaces. The most popular method in these situations is the Karcher mean, also known as the Riemannian center of mass. The prevalence of the Karcher mean may lead some to assume that it provides the best average in all scenarios. However, other subspace averages that appear less frequently in the literature may be more appropriate for certain tasks. The extrinsic manifold mean, the L_2-median, and the flag mean are alternative averages that can be substituted directly for the Karcher mean in many applications.

This paper evaluates the characteristics and performance of these four averages on synthetic and real-world data. While the Karcher mean generalizes the Euclidean mean to the Grassman manifold, we show that the extrinsic manifold mean, the L_2-median, and the flag mean behave more like medians and are therefore more robust to the presence of outliers among the subspaces being averaged. We also show that while the Karcher mean and L_2-median are computed using iterative algorithms, the extrinsic manifold mean and flag mean can be found analytically and are thus orders of magnitude faster in practice. Finally, we show that the flag mean is a generalization of the extrinsic manifold mean that permits subspaces with different numbers of dimensions to be averaged. The result is a ”cookbook” that maps algorithm constraints and data properties to the most appropriate subspace mean for a given application.

1. Introduction

Many computer vision algorithms model collections of data samples as subspaces. For example, the set of images of a single object under different illuminations can be modeled as an illumination subspace, and the subspace can be estimated from a finite number of images. Alternatively, the frames of a video can be viewed as data samples, and the video as a whole modeled as the subspace that spans the observed frames. These are only two of many examples of subspaces being used to model sets of high-dimensional data; for more examples, see [1, 2, 8, 14, 16]. When collections of data are modeled as subspaces, a natural next step is to compute averages of subspaces. For example, when videos are modeled as subspaces, one task might be to cluster subspaces using K-means to determine which videos are similar; K-means, in turn, needs to compute averages of sets of subspaces.

The best known method for computing averages of subspaces is the Karcher mean [10], a.k.a. the Riemannian center of mass. However, just as there are many ways to select a prototype from a set of scalars, including the mean, median or mode values, there are many ways to select a prototype for a set of subspaces. This paper reviews four such methods for selecting prototypes. In addition to the well-known Karcher mean, it describes the extrinsic manifold mean [14], the L_2-median [8], and the flag mean [6]. The four averages are interchangeable in the sense that they can all compute an average subspace given a set of subspaces. Thus, current applications that employ the Karcher mean could substitute any of the other three methods in a straightforward manner. The four averages, however, have different mathematical properties which may make one perform better than another in any specific context.

The primary contribution of this paper is to analyze all four averages with regard to three properties: (1) how they behave in the presence of multiple underlying processes and/or outliers, (2) their efficiency as a function of the number and similarity of samples, and (3) their generality, i.e. whether or not all the subspaces must span the same number of dimensions (i.e. lie on a single Grassman manifold). The analysis is performed on both synthetic and real data, with K-means and prototype video selection as the guiding applications. By analyzing averages in terms of these three properties, we show that no single method is always best, and produce a “cookbook” indicating how to select an average for any particular application.

2. Related Work: Subspace Averages

As mentioned above, this paper compares four methods of computing an average or prototype for set of vector subspaces. Generally, these methods are described as averaging points on a Grassmann manifold, which is defined as the set of all q-dimensional subspaces of an n-dimensional vector space. In this paper, the vector space is $\mathbb{R}^n$, and the relevant Grassmann manifold is denoted $\text{Gr}(n, q)$. Points on $\text{Gr}(n, q)$ are equivalence classes of $n \times q$ matrices, where $X \sim Y$ if $X = YU$ for some $U \in O(q)$, the set of orthogonal $q \times q$ matrices. Computations on Grassmann manifolds are performed using orthonormal matrix representatives for the points, so measures of distance must be orthogonally invariant. In this paper, subspaces, or Grassmann data points will be denoted with square brackets like, $[X] \in \text{Gr}(n, q)$, while matrices and the orthonormal bases for Grassmann points will be denoted by capital letters like, $X \in \mathbb{R}^{n \times q}$.

2.1. The Karcher mean

Distances on a Grassmannian are measured by the length of the shortest geodesic between two points. The canonical metric, the geodesic distance based on arc length, measures the distance between $[X], [Y] \in \text{Gr}(n, q)$ as $d([X], [Y]) = \|\Theta\|_2$, where Θ is the vector of q principal angles between the subspaces $[X]$ and $[Y]$ as discussed by Björck and Golub [4]. The Karcher mean, $[\mu_K]$, is the intrinsic mean on the Grassmann manifold because it is the point that minimizes the mean squared error using the canonical metric,

$$[\mu_K] = \underset{[\mu] \in \text{Gr}(n,q)}{\arg\min} \sum_{i=1}^{P} d([X_i], [\mu])^2. \tag{1}$$

The Karcher mean is most commonly found by using an iterative algorithm like Newton's method or first-order gradient descent [1, 2]. These algorithms exploit the matrix Exp and Log maps to move the data to and from the tangent space of a single point at each step. A unique optimal solution is guaranteed for data that lives within a convex ball on the Grassmann manifold, but in practice not all data sets satisfy this criterion [2, 10]. Using the geodesic distance based on arc length, the maximum distance between two points on $\text{Gr}(n, q)$ is $(\pi/2)\sqrt{q}$, but as Begelfor and Werman illustrated, the convexity radius is $\pi/4$ [2]. This means that if the point cloud being averaged has a radius greater than $\pi/4$ the Exp and Log maps are no longer bijective, and the Karcher mean is no longer unique.

The iterative nature of the Karcher mean algorithms make them quite costly. First-order gradient descent and Newton's method algorithms typically report linear and quadratic convergence, respectively [1, 2, 7, 15]. However, the number of iterations, $N_{d,\epsilon}$, needed to find the Karcher mean depends heavily on the diameter of the data set, d, and the error tolerance parameter, ϵ, so that even with efficient algorithms this calculation can be prohibitive for high-dimensional image and video data.

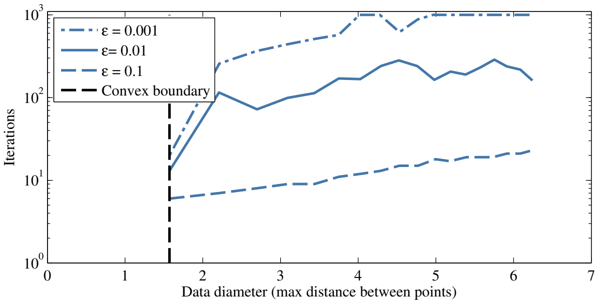


Figure 1: Iterations needed to find the Karcher mean, $[\mu_K]$, versus diameter of the data for different values of ϵ.

Figure 1 shows the number of iterations required to find the Karcher mean of 30 points from $\text{Gr}(1000, 20)$ to within the specified error tolerance as the diameter of the data being averaged grows. The diameter is measured as the furthest distance between any two points using the geodesic distance based on arc length. The maximum number of iterations was set to 1000 so that the algorithm would not run indefinitely. The complexity for our first-order gradient descent implementation of the Karcher mean is $\mathcal{O}\left(nPq^2 N_{d,\epsilon}\right)$, and Figure 1 shows that $N_{d,\epsilon}$ can be quite large if the data being averaged are far apart.

2.2. The L_2-median

The L_2-median, $[\mu_{L_2}]$, is one of many ways of generalizing the median for 1-dimensional data into higher dimensions. It is referred to by many names including the spatial median, the geometric median, the mediancentre, and confusingly the L_1-median [5, 9, 13]. By any name, the L_2-median is the point that minimizes the sum of the distances to the sample points, rather than the sum of the squares of the distances. For subspace data it solves

$$[\mu_{L_2}] = \underset{[\mu] \in \text{Gr}(n,q)}{\arg\min} \sum_{i=1}^{P} d([X_i], [\mu]), \tag{2}$$

where again $d([X_i], [\mu])$ is the geodesic distance based on arc length. As a direct generalization of the median for 1-dimensional data, the L_2-median is robust to outliers [5]. That is to say, if the data being averaged comes from multiple underlying processes, $[\mu_{L_2}]$ will better represent the dominant process rather than the entire set of data. This is in contrast to the behavior of the Karcher mean, which represents the center of mass.

Methods for finding $[\mu_{L_2}]$ also take advantage of the matrix Exp and Log maps, and thus fall prey to the same uniqueness condition as the Karcher mean. One such method comes from Fletcher *et al.*, and adapts the Weiszfeld algorithm to Riemannian manifolds [8]. This algorithm is also a gradient descent method, so while Figure 1 shows data only for the Karcher mean, it can be assumed that the L_2-median is similarly sensitive to data diameter and error tolerance, and the complexity is also $\mathcal{O}\left(nPq^2N_{d,\epsilon}\right)$.

2.3. The extrinsic manifold mean

Srivastava and Klassen proposed the extrinsic manifold mean, $[\mu_E]$, as an alternative to the Karcher mean in 2002 [14]. Given a set of points on $\mathrm{Gr}(n,q)$, the extrinsic mean is the point that minimizes the Frobenius norm squared of the difference in projections of the Grassmann points into the space of $n \times n$ matrices of rank q. That is,

$$[\mu_E] = \underset{[\mu]\in \mathrm{Gr}(n,q)}{\arg\min} \sum_{i=1}^{P} d_{pF}([X_i],[\mu])^2, \tag{3}$$

where $d_{pF}([X_i],[\mu]) = 2^{-\frac{1}{2}}\|X_iX_i^T - \mu\mu^T\|_F$ is the projection Frobenius norm, or projection F-norm, between the points. In contrast to $[\mu_K]$ and $[\mu_{L_2}]$, the extrinsic mean can be found analytically as the solution to an eigenvalue problem, and thus the complexity is $\mathcal{O}(n^3)$ [12]. The flag mean is a generalization of the extrinsic mean, so more of the details will be included in Subsection 2.4.

2.4. The flag mean

As the most recent and least well-known of the subspace averages, the flag mean will be explained in more depth. We begin with some necessary definitions. Let $\widetilde{Q} = \{q_1, q_2, \ldots, q_M\}$ be an ordered set of integers such that $q_1 < q_2 < \ldots < q_M$. A flag in $\mathbb{R}^n$ of type $\widetilde{Q}$ is a nested sequence of subspaces $S_1 \subset S_2 \subset \cdots \subset S_M$ where $\dim(S_i) = q_i$. More background on flags and flag manifolds can be found in [11]. We describe a method for generating a flag that is central to a subspace point cloud. The flag is central in the sense that the kth subspace within the flag is the best k-dimensional representation of the data with respect to a cost function based on the projection Frobenius norm. We refer to the result as the flag mean, denoted $[\![\mu_{pF}]\!]$, where the double square brackets are meant to distinguish a flag from the single square brackets used for a subspace.

Let $\{[X_i]\}_{i=1}^P$ be a finite collection of subspaces of $\mathbb{R}^n$ such that $X_i^TX_i = I$. Let $\widetilde{Q} = \{q_1, \ldots, q_P\}$ be a collection of positive integers, and suppose that $\dim([X_i]) = q_i$ for $i = 1 \ldots P$. We can consider $\{[X_i]\}_{i=1}^P$ to be a point cloud in the disjoint union of a set of Grassmannians, $\coprod_{\widetilde{Q}} \mathrm{Gr}(n, q_i)$.

For this collection of subspaces we wish to find the one-dimensional subspace $[u^{(1)}] \in \mathrm{Gr}(n,1)$ that minimizes the sum of the squares of projection F-norms between itself and $[X_i]$ for $i = 1 \ldots P$. The projection F-norm loses its distinction as a metric when it is used to compare points that do not live on the same manifold, because it is possible to have $d_{pF}([u^{(1)}],[X_i]) = 0$ with $[u^{(1)}] \neq [X_i]$. However, there is still merit in using it to measure the similarity between the objects. Thus we aim to solve

$$\begin{aligned} &\underset{[u^{(1)}]}{\arg\min} && \sum_{i=1}^{P} d_{pF}([u^{(1)}],[X_i])^2 \\ &\text{subject to} && u^{(1)T}u^{(1)} = 1. \end{aligned} \tag{4}$$

This optimization problem is recognizable as the one solved by the extrinsic manifold mean, with the caveat that the data points and the solution are not restricted to live on a single Grassmannian. After finding the optimal $[u^{(1)}]$, the problem is extended to find a sequence of 1-dimensional subspaces that optimize Equation 4 with additional constraints. By solving

$$\begin{aligned} &\underset{[u^{(j)}]}{\arg\min} && \sum_{i=1}^{P} d_{pF}([u^{(j)}],[X_i])^2 \\ &\text{subject to} && u^{(j)T}u^{(j)} = 1 \\ & && u^{(j)T}u^{(k)} = 0 \qquad \text{for } k < j, \end{aligned} \tag{5}$$

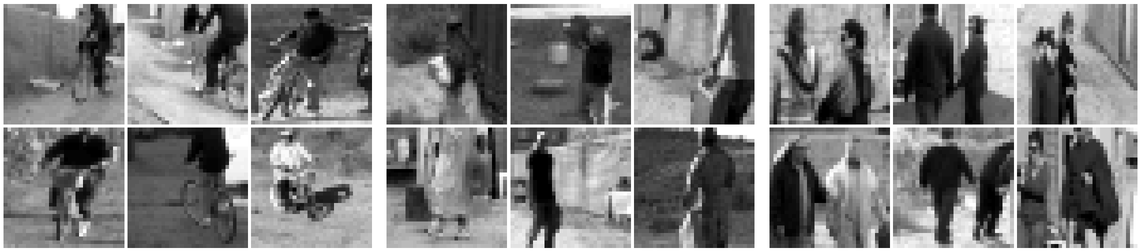
it is possible to find r ordered 1-dimensional subspaces, $\{[u^{(1)}],[u^{(2)}],\ldots,[u^{(r)}]\}$, where r is the dimension of the span of $\cup_{i=1}^P[X_i]$. These subspaces are then central to the collection of points $\{[X_i]\}_{i=1}^P$. From this sequence of mutually orthogonal vectors, the flag mean is defined explicitly as

$$\begin{aligned} [\![\mu_{pF}]\!] = \; & \mathrm{span}\{u^{(1)}\} \subset \mathrm{span}\{u^{(1)},u^{(2)}\} \subset \\ & \ldots \subset \mathrm{span}\{u^{(1)},\ldots,u^{(r)}\}. \end{aligned} \tag{6}$$

While the subspaces $\{[u^{(1)}],[u^{(2)}],\ldots,[u^{(r)}]\}$ are derived iteratively, they can actually be computed analytically. Edelman *et al.* provide the identity $d_{pF}([X],[Y]) = \|\sin\Theta\|_2$ as another way of computing the projection F-norm between two points [7]. This equality and the method of Lagrange multipliers lead to the computation of $\{[u^{(1)}],[u^{(2)}],\ldots,[u^{(r)}]\}$ as the left singular vectors of the matrix $\mathbf{X} = [X_1|X_2|\ldots|X_P]$, where X_i is an orthonormal basis for $[X_i]$. This algorithm for finding $[\![\mu_{pF}]\!]$ is presented in Algorithm 1. The complexity of this algorithm is $\mathcal{O}(n(\sum_{i=1}^P q_i)^2)$. More details on the derivation and mathematical background of the flag mean can be found in [6].

2.5. Flag mean as generalized extrinsic manifold mean

The cost function of the flag mean is the same as the cost function for the extrinsic manifold mean. One separation between the two comes from the relaxation of the



(a) Tracks labeled ‘ride-bike’

(b) Tracks labeled ‘carry’

(c) Tracks labeled ‘walk-group’

Figure 2: Still frames of tracks in three action classes from the Mind’s Eye data.

```
Algorithm 1 $[\![\mu_{pF}]\!]([X_1],\dots[X_P])$
Ensure: $X_i^T X_i = I$ for $i = 1,\dots,P$
  Let $\mathbf{X} = [X_1|X_2|\dots|X_P]$
  Let $r = \dim(\mathrm{span}(\cup_{i=1}^{P}[X_i]))$
  $U\Sigma V^T$ = thin SVD($\mathbf{X}$),
          such that $U = [u^{(1)}|u^{(2)}|\dots|u^{(r)}]$
  $[\![\mu_{pF}]\!] = \{\left[u^{(1)}\right], \left[u^{(1)}|u^{(2)}\right], \dots, \left[u^{(1)}|\dots|u^{(r)}\right]\}$
```

requirement that all subspaces be of the same dimension. Requiring data to live on a single Grassmannian can be undesirably restrictive. For example, suppose subspaces are being used to represent objects under a variety of illumination conditions. Belhumeur and Kriegman discovered that the illumination space of an object is a convex cone that lies near a low dimensional linear subspace [3]. However, the dimension of that subspace depends on the number of unique surface normals. Thus different objects may require subspaces of different dimensions to fully capture variations in lighting, and these subspaces cannot be directly averaged by $[\mu_E]$, $[\mu_K]$, or $[\mu_{L_2}]$.

A typical workaround for this problem is to find the subspace in the data set with the largest dimension, and up-project the rank-deficient data to its Grassmannian. For an $n \times q$ matrix X with $\dim(X) < q$, let $U\Sigma V^T$ be the thin singular value decomposition of X. Then $\dim(UV^T) = q$ and $[UV^T]$ is the closest point to $[X]$ on $\mathrm{Gr}(n, q)$. Unfortunately this projection is not unique, and can create artifacts if $[UV^T]$ is a point to be averaged. The flag mean can be computed for subspaces of different dimensions, because $[\![\mu_{pF}]\!]$ is built one dimension at a time. Thus it avoids this non-unique projection. For all three of the averages other than the flag mean, this method for finding the closest orthonormal matrix is used to preprocess the video data in the experiments of Subsection 3.3 & Subsection 3.4. We will see in Figure 4a that this projection has a significant negative effect on the ability of the extrinsic mean to represent the data.

3. Empirical evaluation

This section describes three experiments that were performed to illustrate the characteristics of the various subspace averages. The first experiment finds the averages of a synthetic 2-dimensional data set in an effort to visualize how the different methods behave when a point cloud is not tightly clustered or contains data generated by multiple processes. The second experiment uses $[\mu_K]$, $[\mu_{L_2}]$, $[\mu_E]$, and $[\mu_{pF}]$ to identify exemplars from clusters of similar data that have been grouped using a method that does not require averaging. Each choice of prototype is then evaluated according to whether or not it matches the dominant class of data in its cluster. The third and final experiment uses the averages to perform K-means clustering, and the results are evaluated for cluster purity. The second and third experiments are performed on noisy, real-world data.

3.1. Data

Three data sets are used to evaluate the subspaces averages. The first, used in Subsection 3.2, is a collection of 1-dimensional subspaces in $\mathbb{R}^2$, or points on $\mathrm{Gr}(2, 1)$. These lines were generated synthetically by two processes. They come from normal distributions about two orthogonal means with standard deviations of $\sigma = 0.2$.

The data for prototype selection in Subsection 3.3 consists of $2,345$ short video clips extracted from larger and longer outdoor videos collected as part of DARPA’s Mind’s Eye project. The video clips – which we call tracks – were automatically centered on moving objects, mostly people, through background subtraction. However, the background subtraction process is imperfect and sometimes only part of the person or object is visible. All tracks are 48 frames long (about 1.5 seconds) and are rescaled to a size of 32×32 pixels. The tracks were manually assigned labels based on the action they depict. There are a total of 77 unique labels. Figure 2 shows examples of frames from tracks labeled “ride-bike”, “carry”, and “walk-group”. The largest number of tracks associated with a label is 637 (“walk”) and the smallest is 1 (“climb,” “shove,” etc.).

The third data set is a subset of the second, and is used for K-means clustering in Subsection 3.4. Some classes are in the second data set are singletons, making it poorly suited to K-means clustering. It was pruned to 601 tracks with 17 unique labels. In this subset, the largest class has 187 members and the smallest has 3. To represent the videos as subspaces, the frames of each track are vectorized and concatenated into a matrix of size 1024×48. The Grassmann point associated with each track is the span of the column space of its matrix. One might expect that the resulting data points would live on a single Grassmann manifold; i.e. $\mathrm{Gr}(1024, 48)$. Often, however, the matrices are not full rank. Physically, this means that some of the frames in the track are linearly dependent. Therefore, apart from the flag mean which can be computed for subspaces of variable dimensions, we replace rank-deficient samples with the nearest point on $\mathrm{Gr}(1024, 48)$ as described in Subsection 2.5.

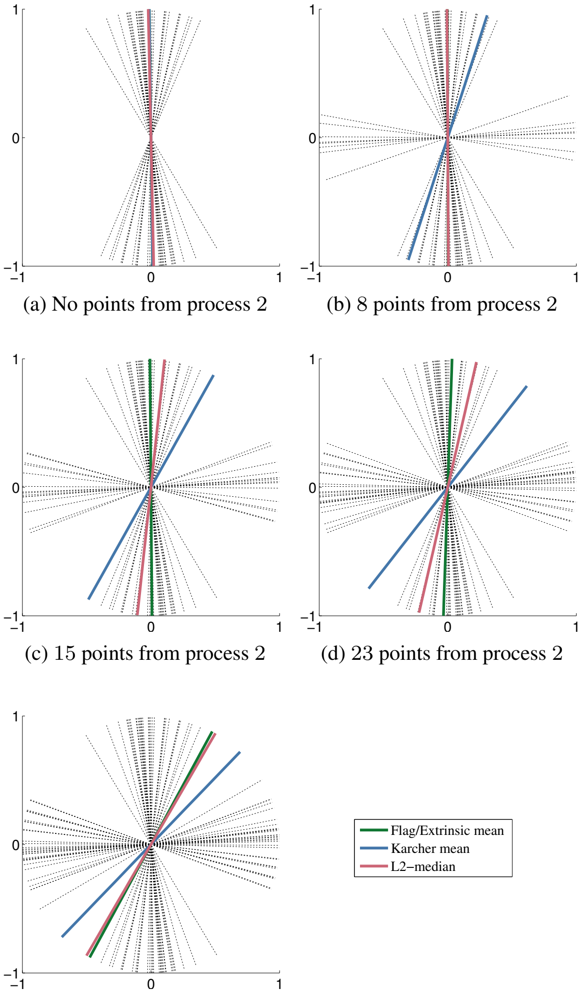
Additionally, these experiments use a single subspace, $[\mu_{pF}]$, from within the flag $[\![\mu_{pF}]\!]$ as the subspace average, because using the full flag would require different measures of distance and would make the results incomparable. The subspace chosen as $[\mu_{pF}]$ in each experiment is the one that lives on the same manifold as the other averages, i.e. $\mathrm{Gr}(2, 1)$ for the first experiment, and $\mathrm{Gr}(1024, 48)$ for the second two. Note that a subspace of the appropriate size will always be contained within $[\![\mu_{pF}]\!]$, and that it can serve as a direct replacement for the others in practice.

3.2. Experiment 1: Data fitting

The purpose of this experiment is to gain intuition about the behavior of the averages. The data set contains points generated from two processes, which are normal distributions centered around the axes. The first process generates the points about $[0, 1]^T$. There are 30 points from this distribution with $\sigma = 0.2$. The second process creates points about $[1, 0]^T$ with $\sigma = 0.2$. Figure 3 shows the behavior of the subspace averages as points from the second process are added to the data set. For this synthetic data, the points are all subspaces of the same dimension. Thus the extrinsic mean and the flag mean produce the same solution. In Figure 3 they are both represented by the green line.

In agreement with our intuition, $[\mu_K]$ behaves like the center of mass. The introduction of even a small number of points from the second process pulls $[\mu_K]$ to the fringe of the data generated by the first process. It appears to best represent the entire set of data, not just the points from one process. If all the data is valid for the task at hand, this point is the one that truly minimizes the mean squared error.

The other three averages are more resistant to the introduction of points around the horizontal axis. They appear to better represent the larger cloud of points, which were generated by the first process. This agrees with what one would expect from a robust average like the L_2-median. The flag mean and the extrinsic mean are mathematically not generalizations of a median. However, the use of the projection F-norm allows them to imitate that behavior. Using $\|\sin\Theta\|_2$ in the cost function gives large angles less weight than small ones. For the resulting averages, this translates into points that approximate medians by paying more attention to the data that is tightly clustered than the points that are far away. If the data from the second process in Figure 3 is interpreted as noise, $[\mu_{pF}]$, $[\mu_E]$, and $[\mu_{L_2}]$ do a better job of representing the relevant data.



(a) No points from process 2

(b) 8 points from process 2

(c) 15 points from process 2

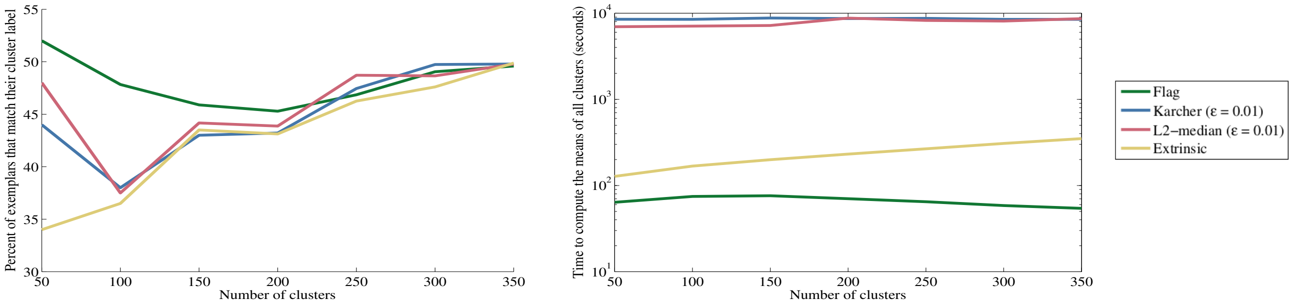
(d) 23 points from process 2

(e) 30 points from process 2

Figure 3: Behavior of the flag and extrinsic means (green), the Karcher mean (red), and the L_2-median (blue), as points from the second process are added to the data set.

3.3. Experiment 2: Prototype selection

Figure 3 shows that the L_2-median, extrinsic mean and flag mean are more robust to outliers than the Karcher, at



(a) Percentage of prototypes correctly chosen in a trial vs. the number of clusters in a set of trials.

(b) Total time required to compute the average of every cluster in a trial vs. the number of clusters in a trial.

Figure 4: Results of the Experiment 2: Prototype selection.

least in theory. But real data is never as clean as a theoretical model. The second experiment tests all four means on the task of selecting prototypes from noisy sets of real tracks, while also allowing us to measure the cost of computation and the effects of rank deficient data. On each trial, the system is given a set of similar tracks, and computes the means of the set. It then selects the closest sample to the mean as a prototype. Since the goal is to find prototypes that represent the set well, an automatically selected prototype is considered ‘correct’ if the action label associated with the prototype is the most common action label in the set, and ‘incorrect’ otherwise. The quality of a mean is measured by how often it predicts a correct prototype.

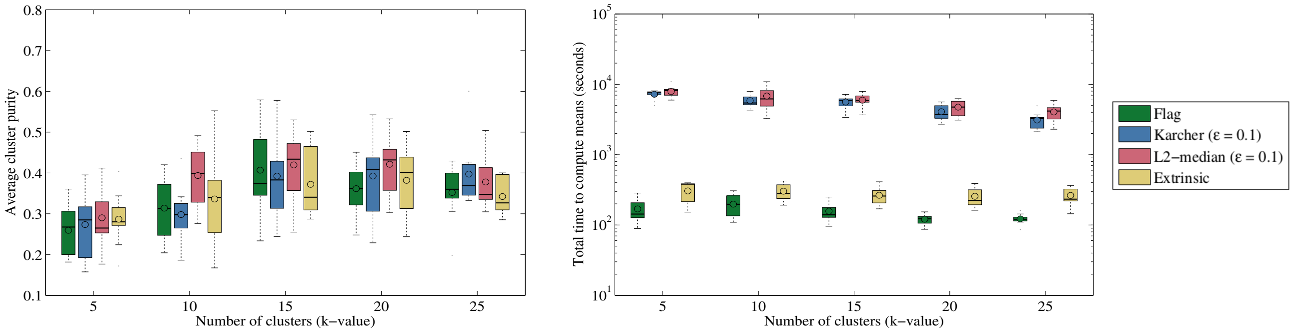
Some methodological details. First, the Karcher mean and L_2-median are sensitive to how similar the samples being averaged are to each other. We therefore formed sets of similar tracks by clustering. To avoid interactions between the prototype selection method and the clustering algorithm, we clustered with a method that does not require computing means, namely agglomerative clustering with Ward’s linkage. Second, the Karcher mean and the L_2-median require an error tolerance threshold to determine convergence. We tested them with $\epsilon = 0.01$ radians to achieve our results.

Figure 4a illustrates how often the prototype’s label matches the label of the dominant action in a cluster. We see that when the number of clusters is small, and thus the number of samples per cluster is high, the flag mean (green) outperforms the others, followed by the L_2-median (red). We believe this is because large clusters contain more outliers. As the number of clusters grows, the accuracy of each mean increases and they all converge. This is consistent with there being fewer outliers. When the number of clusters approaches 350, there are on average only 7 tracks per cluster, making the choice of prototype less difficult. Near this point all four methods converge in accuracy. One conclusion that these results suggest is that the flag mean and the L_2-median do a better job of finding the dominant action in a cluster when there are more tracks per cluster. When the clusters have fewer actions and thus presumably more pure, all four means perform comparably.

Figure 4a also shows that the flag mean significantly outperforms the extrinsic manifold mean when the number of clusters is small. This might seem counter-intuitive, since both means minimize the sum of the squared sines of the principal angles. Some of the video tracks generate rank deficient matrices, however. In the case of the extrinsic manifold mean, these data samples have to be up-projected, as described in Section 2.5. This introduces error and makes the extrinsic manifold mean less accurate. The flag mean does not require this step, leading to a more accurate mean.

Figure 4b shows the total time needed to compute the mean of all clusters versus the number of clusters used. The number of clusters in a trial ranged from 50 to 350 in increments of 50. The experiment was run with Matlab code timed by the computer’s wall clock, but even with that caveat the differences are meaningful. On average, it took 0.50 seconds to compute the flag mean for a single cluster, 1.40 seconds for the extrinsic manifold mean, 55.16 seconds for the L_2-median, and 63.37 seconds for the Karcher mean. To compute the means for all the clusters in a trial took on average 66.07 seconds for $[\mu_{pF}]$, 235.76 seconds for $[\mu_E]$, 7.85×10^3 seconds or 2.18 hours for $[\mu_{L_2}]$, and 8.58×10^3 seconds, or about 2.38 hours for $[\mu_K]$.

One point of interest in Figure 4b is how the time needed to compute the extrinsic mean for all clusters increases as the number of clusters grows, in contrast to the behavior of the flag mean. The complexity of the eigenvector decomposition used to find $[\mu_E]$ is $\mathcal{O}(n^3)$, and is independent of the number of samples. However, since the ambient space here is $\mathbb{R}^{1024}$, repeating that computation 350 times is costly. The complexity of the flag mean, $\mathcal{O}(n(\sum_{i=1}^{P} q_i)^2)$, depends on the ambient dimension, the number of samples, and the dimension of the sample subspaces. If the number of samples per cluster was constant as the number clus-



(a) Purity of the clusters formed using the K-means algorithm.

(b) Total time to run the K-means algorithm to convergence.

Figure 5: Results of Experiment 3: K-means clustering.

ters increased, the total time for the flag mean would be much closer to that of the extrinsic mean. Furthermore, if $\sum_{i=1}^{P} q_i > n$, the SVD in Algorithm 1 can be replaced with an eigenvector decomposition to reduce the cost.

3.4. Experiment 3: K-means clustering

The second task on real data is K-means clustering. K-means is a well-known algorithm that iteratively clusters data by computing the means of sets of samples and then re-assigning every sample to the nearest mean. As a result, it matters both how accurate the computed mean is and how quickly it can be computed. The first step of K-means initializes cluster centers to randomly chosen samples from the data set. Distances are then calculated between every vector space and each cluster center, and subspaces are assigned to the closest center. In the second step, means are re-calculated from the vector spaces (i.e. data points) assigned to each cluster, distances are calculated between the means and the spaces, and each point is re-assigned to the nearest cluster center mean. We allow Step 2 to iterate until the calculation of a new mean does not change cluster membership. We measure the quality of a cluster in terms of its label purity. For example, if all the samples in a cluster share the same label, its purity is 100%; if half the samples share a label, its purity is 50%. In general, if there are N samples in a cluster, the lowest possible purity is $\frac{1}{N}$.

In Figure 5a, we see the cluster purity for the K-means clusters made using $[\mu_{pF}]$, $[\mu_K]$, $[\mu_{L_2}]$, and $[\mu_E]$. For this task, the error tolerance was set at $\epsilon = 0.1$ radians, because the tolerance used in Experiment 2 was computationally infeasible. The clustering was run 10 times for each value of K to get the data displayed. The cluster purity is low for all of the subspace means, indicating that the data set is challenging. The highest purity for a single cluster was 60%, achieved by the Karcher mean when K was 20. The highest average purity for a single value of K was 43.4%, reached by the L_2-median with K= 15. It makes sense that the best results were achieved for K values of 15 and 20, because there are 17 unique labels in the data set.

Figure 5b shows the total time required to compute the means of each cluster until the K-means algorithm has converged. The time is almost two orders of magnitude bigger for the Karcher mean and the L_2-median than the extrinsic mean and the flag mean. On average, creating all K means for one iteration took 7.68 seconds for $[\mu_{pF}]$, 13.92 seconds for $[\mu_E]$, 258.62 seconds for $[\mu_K]$, and 294.48 seconds for $[\mu_{L_2}]$. The difference in time between the iterative methods and the closed form ones decreases as the number of clusters grows. One interpretation of this trend is that as the average number of tracks in a cluster shrinks, the diameter of the point set on the Grassmann manifold likely does as well. This in turn speeds up the convergence of the Karcher mean and L_2-median algorithms as we saw in Figure 1.

Overall the L_2-median appears to have a slight edge in terms of accuracy, as one might expect from a robust average on a messy data set. However, the average purity of each method for each value of K is within the error bars of the others. The greatest difference comes from the speed with which we obtain these comparable results. The extrinsic mean and the flag mean far out-stripped the iterative methods. In fact, the total time for the K-means algorithm to converge was on the order of one day per trial for the Karcher mean and L_2-median, whereas the analytical methods could complete one trial in tens of minutes.

4. Conclusion

This paper explores the utility of four subspace averages. Each method has advantages in the right context, but the L_2-median, the extrinsic mean, and the flag mean have been overshadowed by the Karcher mean in the literature. Building an intuition about the properties of each mean allows us to choose the appropriate representation for an application.

	Model All Data	Model Dominant Process
Speed is less important	Karcher mean	L_2-median
Speed is more important	?	flag mean

Table 1: Mean selection cookbook.

When the vector subspaces being averaged span the same number of dimensions, are close together, and are equally reliable, all of the subspace averages provide a similar solution, as in Figure 3a. However, without these idealized properties, a choice of representation must be made. In a scenario where the data are subspaces of variable dimensions, the flag mean is the only method that can directly average the subspaces; the other averages require some form of non-unique projection as a pre-process.

If we assume that the dimension of the subspaces is the same, the choice becomes one of application. For tasks where data is generated by a single process and the subspace dimension is low or time is not an issue, the Karcher mean is the appropriate choice. It is the sole average that minimizes the mean squared error using the intrinsic metric of the Grassmann manifold. When data contains outliers or is generated by a mixture of processes, on the other hand, the other averages do a better job of modeling the dominant process. In particular, the L_2-median is a true generalization of a median. It is robust to outliers, and like the Karcher mean employs the intrinsic distance of the Grassmann manifold.

If time is a factor, the extrinsic mean and the flag mean can approximate the L_2-median at a fraction of the cost. The flag mean can provide the same result as the extrinsic mean whenever the extrinsic mean is applicable. However, flag mean's ability to accommodate data without additional projection can lead to a more accurate representation as we saw in Figure 4a and the flag mean can be computed more quickly as shown in Figure 4b, thus it is a better choice.

Table 1 summarizes the scenarios in which each subspace average is most appropriate. There is no fast approximation of the Karcher mean currently in the literature, so the bottom left box remains empty. It is important to note that the choices made in Table 1 are based on the cost functions being minimized by the averages. Our experiments support these claims, but the claims themselves are based on mathematical properties. Our experiments simply confirm what the mathematics say. All Matlab code and experimental data from this paper is publicly available at www.cs.colostate.edu/~vision/summet.

Acknowledgments. This research was partially supported by the NSF: CDS&E-MSS-1228308, DMS-1322508, DOD-USAF: FA9550-12-1-0408 P00001 and by DARPA: N66001-11-1-4184. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80:199–220, 2004. 1, 2

[2] E. Begelfor and M. Werman. Affine invariance revisited. *CVPR*, 2:2087 – 2094, 2006. 1, 2

[3] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible illumination conditions? *IJCV*, 28(3):245–260, 1998. 4

[4] A. Björck and G. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123):579–594, 1973. 2

[5] Y. Dodge and V. Rousson. Multivariate L1 mean. *Metrika*, 49(2):127–134, 1999. 2

[6] B. Draper, M. Kirby, J. Marks, T. Marrinan, and C. Peterson. A flag representation for finite collections of subspaces of mixed dimensions. *Linear Algebra and its Applications*, 451:15–32, 2014. 1, 3

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

[8] P. T. Fletcher, S. Venkatasubramanian, and S. Joshi. The geometric median on Riemannian manifolds with application to robust atlas estimation. *NeuroImage*, 45(1 Suppl):S143, 2009. 1, 3

[9] J. Haldane. Note on the median of a multivariate distribution. *Biometrika*, 35(3-4):414–417, 1948. 2

[10] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977. 1, 2

[11] D. Monk. The geometry of flag manifolds. *Proceedings of the London Mathematical Society*, 3(2):253–286, 1959. 3

[12] Q. Rentmeesters, P. Absil, P. Van Dooren, K. Gallivan, and A. Srivastava. An efficient particle filtering technique on the Grassmann manifold. In *ICASSP*, pages 3838–3841. IEEE, 2010. 3

[13] C. G. Small. A survey of multidimensional medians. *International statistical review*, 58(3):263–277, 1990. 2

[14] A. Srivastava and E. Klassen. Monte Carlo extrinsic estimators of manifold-valued parameters. *IEEE Transactions on Signal Processing*, 50(2):299–308, 2002. 1, 3

[15] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *PAMI*, 33(11):2273–2286, 2011. 2

[16] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on Riemannian manifolds. In *CVPR*, pages 1–8, 2007. 1