THESIS

THE FLAG OF BEST FIT AS A REPRESENTATIVE FOR A COLLECTION OF LINEAR SUBSPACES

Submitted by

Timothy P. Marrinan

Department of Mathematics

In partial fulfullment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Fall 2013

Master's Committee:

Advisor: Michael Kirby

Dan Bates
Bruce Draper
Chris Peterson

ABSTRACT

THE FLAG OF BEST FIT AS A REPRESENTATIVE FOR A COLLECTION OF LINEAR SUBSPACES

This thesis will develop a technique for representing a collection of subspaces with a flag of best fit, and apply it to practical problems within computer vision and pattern analysis. In particular, we will find a nested sequence of subspaces that are central, with respect to an optimization criterion based on the projection Frobenius norm, to a set of points in the disjoint union of a collection of Grassmann manifolds. Referred to as the flag mean, this sequence can be computed analytically. Three existing subspace means in the literature, the Karcher mean, the extrinsic manifold mean, and the L_2-median, will be discussed to determine the need and relevance of the flag mean. One significant point of separation between the flag mean and existing means is that the flag mean can be computed for points that lie on different Grassmann manifolds, under certain constraints. Advantages of this distinction will be discussed. Additionally, results of experiments based on data from DARPA's Mind's Eye Program will be compared between the flag mean and the Karcher mean. Finally, distance measures for comparing flags to other flags, and similarity scores for comparing flags to subspaces will be discussed and applied to the Carnegie Mellon University, 'Pose, Illumination, and Expression' database.

TABLE OF CONTENTS

ABSTRACT ii

Chapter 1. Introduction 1

Chapter 2. Overview of the Problem 4

2.1. The Mind's Eye project 4

2.2. Mathematical background 5

2.3. Existing subspace averages 11

Chapter 3. Presentation and Derivation 19

3.1. Motivation for an additional subspace average 19

3.2. Derivation 20

Chapter 4. Results and Analysis 25

4.1. Synthetic illustrations 25

4.2. Comparisons to the Karcher mean on practical experiments 30

4.3. Computations with flags 36

4.4. Analysis 48

Chapter 5. Concluding Remarks 58

5.1. Future work 58

BIBLIOGRAPHY 60

CHAPTER 1

Introduction

Pattern recognition is inherently concerned with finding the most likely classification of data with respect to some type of allowable variation. The techniques in pattern recognition can be reasonably partitioned into supervised classifiers and unsupervised classifiers. Supervised classifiers require a human to label the data points in the training set according to their class. After labeling, these models often fit a probability distribution to each class of data. Probabilistic classifiers are commonly used, partially because their labels can include a confidence value along with an ordered list of best labels. However, they often disregard the shape of the space the data live in. On the other hand, unsupervised classifiers often assume that data are restricted to some structured space. If the geometry of that space is known, any point in the space may be thought of as belonging to the data, even if such a sample has never been observed. In this case, classifications can be inferred by the location of the data points within the space.

Many researchers have had success representing data as points on a Grassmann manifold, or the set of all q-dimensional subspaces of $\mathbb{R}^n$, when their data can be seen as equivalent under the set of all rotations. For example, if a set of images of one person's face under different illumination conditions is treated as a subspace, it is possible to identify the person in the presence of any lighting condition the can be approximated by some linear combination of the original images. For this reason, linear subspace models have become increasingly popular. Admitting some type of variation in a recognition model helps avoid over-fitting to a training set and subsequently avoids exact pattern matching. Representations of videos and images as points on Grassmann manifolds are used for activity modeling and recognition [26],

shape analysis [20], appearance recognition [18], action classification [16], face recognition [22, 14, 26], person detection [28], subspace tracking [25], and general manifold clustering [2]. Within these applications is the more contained and general objective of sample-to-set matching. As the number of points that a classifier is trained on grows, pairwise comparisons between new samples and the training set quickly become computationally infeasible. Finding an average representation for each class of data is beneficial, because it allows new samples to be classified based on their similarity to these averages rather than the whole data set. This reduces the number of comparisons needed to classify a new sample.

Before sample-to-set matching can begin, most practical problems require a great deal of effort to achieve a subspace representation [26, 16]. For example, it is often non-trivial to standardize the dimensions of data points so that the subspaces live on a Grassmannian. Additionally, a data set may have multiple meaningful forms of variation, and it can be difficult to isolate a single type of variation for classification.

Once a linear subspace representation of a data set has been realized, there are additional challenges in finding an average for such a collection. Most commonly used techniques, such as the Karcher mean, are iterative in nature and thus expensive to compute. Many of the existing algorithms assume that the subspaces are points on a single Grassmann manifold. Due to the curvature of this manifold the algorithms can only find a unique average for subspaces that live within a convex ball, severely limiting the data sets that can be averaged [2]. The techniques can also be sensitive to noise, and since they assume the data live on a single Grassmannian, the subspaces to be averaged must all be of the same dimension.

In light of the difficulties in treating data points as subspaces we propose a novel average, the flag of best fit, as a representative for a collection of linear subspaces of $\mathbb{R}^n$. A thorough

treatment of this flag mean requires some mathematical context. Chapter 2 of this thesis discusses the setting in which this problem was formulated, defines the relevant properties the Grassmann manifold and flags of $\mathbb{R}^n$, and reviews other subspace averages in the literature. Chapter 3 motivates the optimization problem that the flag mean solves, outlines six ways of measuring the success of this representation, and details an analytic solution to the optimization problem. In Chapter 4 the flag mean is applied to three synthetic experiments that build intuition about the flag, and to two practical data sets that characterize the value of the representation. The six measures of success from Chapter 3 are then discussed with respect to these experiments. Finally in Chapter 5 we discuss the limitations of this subspace average, and suggest areas for further study.

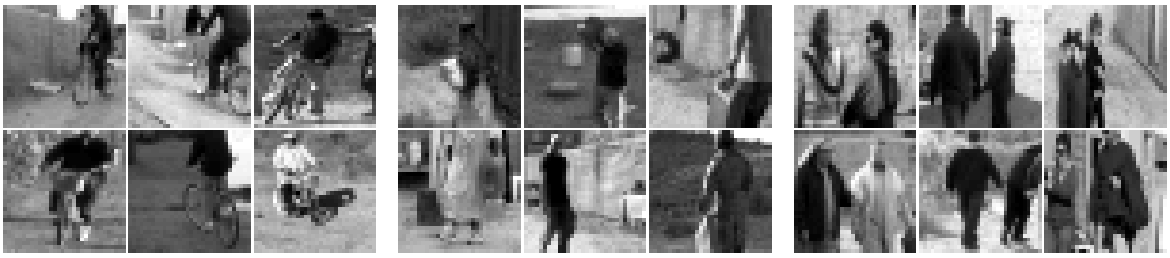
CHAPTER 2

Overview of the Problem

2.1. The Mind's Eye project

The Defense Advanced Research Projects Agency, or DARPA, commissions advanced research for the Department of Defense. One such research project is the Mind's Eye program. The goal of the Mind's Eye program is to develop machine-based visual intelligence for 'smart' cameras, that can recognize actions and events in real-time video data and describe them using simple text messages. A research team from the computer science department of Colorado State University, headed by professor Bruce Draper, has contributed heavily to this program with an approach described as, 'Visual Intelligence Through Latent Geometry and Visual Guidance.' Their machine learning system identifies short clips of video, cropped for length and size, that correspond to singular actions such as 'turn' or 'walk', and pieces of larger actions like 'arm-movement.' Using a representation of these video clips as 3rd order tensors from Lui *et al.* and a collection of randomized decision trees developed by O'Hara and Draper, the CSU team was able to cluster a huge number of training clips into semantically meaningful groups with only selective human guidance [16, 19]. Samples from three of these clusters can be seen in Figure 2.1.

Visual inspection suggested that many of the resulting clusters were dominated by a single action, and empirical results showed that they did a good job of classifying new samples. However, without human supervision it was difficult to identify the exceptional clusters that contained dissimilar actions, and thus contributed negatively to classification. In an attempt to quantify the purity of these clusters, there arose a need for a subspace average.



(A) Tracks labeled 'ride-bike'

(B) Tracks labeled 'carry'

(C) Tracks labeled 'walk-group'

FIGURE 2.1. Examples of three action classes from the Mind's Eye data.

2.2. MATHEMATICAL BACKGROUND

2.2.1. VIDEOS AND IMAGES AS SUBSPACES. The majority of the applications studied in this thesis are based around representations of video clips or collections of images as subspaces, so it is prudent to discuss how this representation is achieved. The idea was discussed by Belhumeur and Kriegman in 1997 when they considered the question, "What is the set of images of an object under all possible illumination conditions? [3]" Their inquiry was inspired by the appearance recognition techniques of Murase and Nayar in 1995, and the Eigenpicture methods of Sirovich and Kirby in 1987 and Turk and Pentland in 1991 [18, 22, 14, 27]. They discovered that the set of all n-pixel images under all possible lighting conditions is a convex cone in $\mathbb{R}^n$, and this cone could be determined out of the span of as little as three images under different illumination conditions. Their method for constructing the cone was to vectorize each digital image by concatenating the columns, creating an $n \times 3$ matrix from the vectorized images, and then finding a basis for the subspace spanned by these images using the well-known singular value decomposition. The resulting basis vectors allowed Belheumer and Kriegman to approximate images with previously unsampled lighting conditions by taking linear combinations of these basis vectors. More concisely, a collection of images that vary under pose, subject, or illumination can be seen as a linear subspace when

each of the images is vectorized and any rotation of an orthogonal basis for these vectors is viewed as equivalent. This gives a collection of images a very natural representation on a Grassmann manifold.

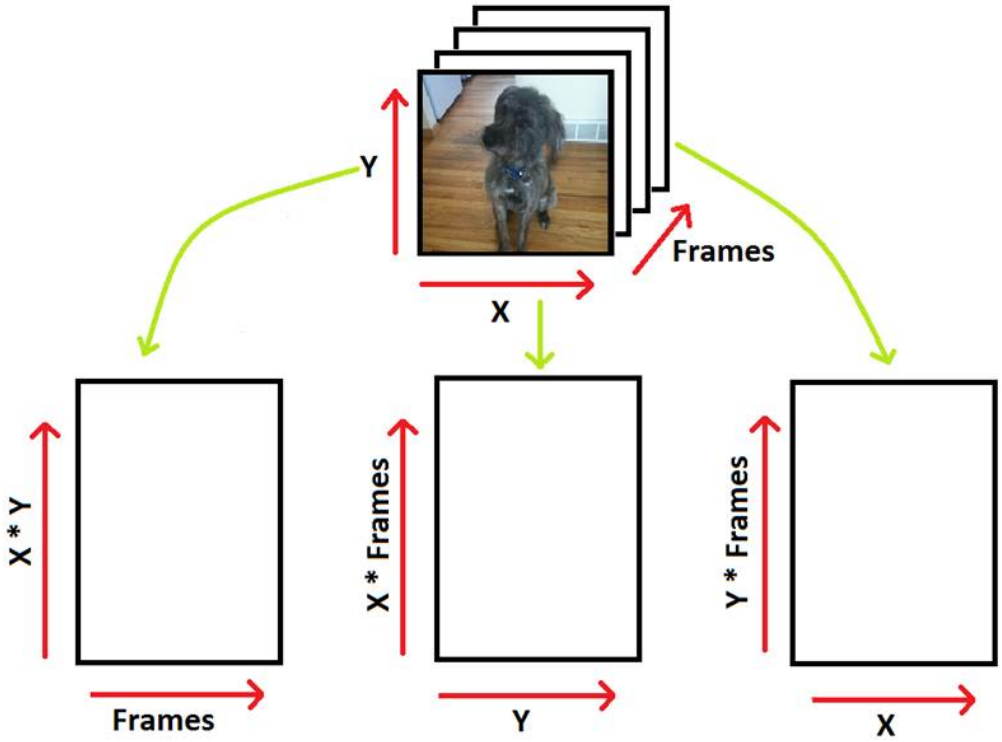
Definition A **Grassmann manifold** is the set of all q-dimensional subspaces of an n-dimensional vector space, V. For the purposes of this paper, $V = \mathbb{R}^n$, and thus a Grassmannian will be denoted $\mathrm{Gr}(q, n)$ with no mention of V. Points on $\mathrm{Gr}(q, n)$ are equivalence classes 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. Points on Grassmannians and linear subspaces will be denoted with square brackets,

$$\mathrm{span}\{X\} = [X] \in \mathrm{Gr}(q, n), \tag{1}$$

while the matrix representatives for these points will be denoted as capital letters with no brackets, like $X \in \mathbb{R}^{n \times q}$. Since computations are done on the matrix representatives, and not the actual subspaces, this thesis will change between the subspace notation and matrix notation as appropriate. For convenience, our choice of representative for an equivalence class will be an orthonormal matrix. For more information on Grassmann manifolds, refer to [15, 7].

In a similar fashion, Turaga *et al.* and Lui *et al.* each discussed the idea of action subspaces for video data [26, 16]. Treating a video clip as a collection of image matrices, they vectorized each frame, and found an orthogonal basis for the span of the frames. This representation has proved quite effective for action recognition. The ability to classify action was improved by treating each video as a 3rd order tensor, and representing it with the subspaces that spanned each of the three possible unfoldings of the tensor. In this scenario,

each video is thought of a 3-tuple on a disjoint union of Grassmann manifolds. Figure 2.2 shows how a video clip can have three different matrix representations, depending on which dimensions are concatenated. A basis can then be found for each of the resulting matrices. This technique will be employed in the applications to Mind's Eye data in Chapter 4.



(A)

FIGURE 2.2. An illustration of a video clip being unfolded into three matrix representations.

2.2.2. PROPERTIES OF GRASSMANN MANIFOLDS. In Euclidean space, distances are measured by the minimum length of a straight line between two points. In the case of a curved space like the Grassmann manifold, a straight line would not actually lie in the space. For manifolds a straight line is generalized as a geodesic, the shortest path on the manifold between two points, and we measure the distance between two points by the length

of the geodesic between them. We would like computations on a Grassmannian to be independent of our choice of representative for a point $[X] \in \mathrm{Gr}(q, n)$, so any measure of distance must be an orthogonally invariant function. It has long been known that the principal angles between linear subspaces are orthogonally invariant, and thus numerous distance metrics on Grassmannians have been developed based on principal angles [5].

Definition If $[X]$ and $[Y]$ are subspaces of $\mathbb{R}^n$ such that $q = \min\{\dim([X]), \dim([Y])\}$, then the **principal angles** $\theta_k \in [0, \pi/2]$ between $[X]$ and $[Y]$ are recursively defined for $k = 1, 2, \ldots, q$ by

$$\cos\theta_k = \max_{v \in [X]} \max_{u \in [Y]} u^T v = u_k^T v_k, \tag{2}$$

subject to the constraints $\|u\| = \|v\| = 1$, and $u_j^T u_k = v_j^T v_k = 0$ for $j = 1, 2, \ldots k-1$. The vectors $\{u_1, u_2, \ldots u_q\}$ and $\{v_1, v_2, \ldots v_q\}$ are called the principal vectors of the pair of spaces.

It is worth mentioning that if Q_X and Q_Y are orthogonal bases for $[X]$ and $[Y]$, the principal angles and vectors between $[X]$ and $[Y]$ can be calculated by finding the thin singular value decomposition of $Q_X^T Q_Y$ [5]. If the decomposition is written as $Q_X^T Q_Y = U\Sigma V^T$, then the principal vectors of $[X]$ are the columns of the matrix $Q_X U$, the principal vectors of $[Y]$ are the columns of $Q_Y V$, and the cosines of the principal angles between the pair of spaces are the diagonal values of Σ, i.e. $\cos\theta_k = \sigma_k$.

What this characterization does not make entirely clear is that the first principal angle between $[X]$ and $[Y]$ is the arc length between vectors $u_1 \in [X]$ and $v_1 \in [Y]$ that are the closest together, and that each of the following principal angles is the arc length between the closest vectors of $[X]$ and $[Y]$ in the orthogonal complements to span of the principal vectors

that have already been found. Based on this interpretation of principal angles, two useful metrics on Grassmann manifolds from the literature will be reviewed, the geodesic distance based on arc length and the projection Frobenius norm. A more thorough treatment of the various metrics on Grassmann manifolds can be found in [7].

Definition If $[X], [Y] \in \mathrm{Gr}(q, n)$, then the **geodesic distance based on arc length** between the two is defined as

$$d([X], [Y]) = \|\Theta\|_2, \tag{3}$$

where Θ is the vector of principal angles between $[X]$ and $[Y]$.

It is shown in [7] that this measure of distance is the canonical metric on the Grassmann manifold in the sense that it is equivalent to the Euclidean metric in the tangent space of a single point on the Grassmannian. This metric arises from the representation of the Grassmann manifold as a quotient space, i.e. $\mathrm{Gr}(q, n) = O(n) \big/ (O(q) \times O(n-q))$ where $O(n)$ is the group of $n \times n$ orthogonal matrices.

Definition If $[X], [Y] \in \mathrm{Gr}(q, n)$, then the **projection Frobenius norm**, or projection F-norm, between the two is defined as

$$d_{pF}([X], [Y]) = 2^{-\frac{1}{2}} \|XX^T - YY^T\|_F = \|\sin\Theta\|_2, \tag{4}$$

where Θ is the vector of principal angles between $[X]$ and $[Y]$.

This metric is derived from embedding the Grassmann manifold in the space of $n \times n$ projection matrices of rank q and measuring the distance between them using the Frobenius

norm. It can be be shown that for $[X] \neq [Y]$, $d([X],[Y]) > d_{pF}([X],[Y])$, and that these metrics are asymptotically equivalent. Thus for points close together, $d([X],[Y]) \approx d_{pF}([X],[Y])$. Additionally, since both metrics are based on principal angles, distances on $\mathrm{Gr}(q,n)$ are bounded.

PROPOSITION 2.2.1. *For all* $[X],[Y] \in Gr(q,n)$,

$$d([X],[Y]) \leq (\pi/2)\sqrt{q}, \quad \textit{and} \tag{5}$$

$$d_{pF}([X],[Y]) \leq \sqrt{q}. \tag{6}$$

PROOF. Let $[X],[Y] \in \mathrm{Gr}(q,n)$. Then $q = \min\{\dim([X]), \dim([Y])\}$, and $\theta_k \in [0, \pi/2]$ for $k = 1, 2, \ldots q$. Thus we have

$$d([X],[Y]) = \|\Theta\|_2 = \sqrt{\sum_{k=1}^{q} \theta_k^2} \leq \sqrt{\sum_{k=1}^{q} (\pi/2)^2} = \sqrt{q(\pi/2)^2} = (\pi/2)\sqrt{q}, \tag{7}$$

and similarly

$$d_{pF}([X],[Y]) = \|\sin\Theta\|_2 = \sqrt{\sum_{k=1}^{q} \sin^2(\theta_k)} \leq \sqrt{\sum_{k=1}^{q} 1} = \sqrt{q} \tag{8}$$

as desired. □

The boundedness of the Grassmann manifold will become important when averaging point clouds.

2.2.3. FLAGS AND FLAG MANIFOLDS. The representation of data as points on Grassmannians is beneficial for classification as discussed in Chapter 1, but for reasons that will

be explained in Chapter 3 the result of averaging point clouds on Grassmann manifolds in this thesis will be characterized as a flag.

Definition Let $\mathbb{I}$ be a finite set of integers, $\mathbb{I} = \{q_1, q_2, \ldots, q_m\}$. A **flag** of $\mathbb{R}^n$ is a nested sequence of subspaces $[S_1] \subset [S_2] \subset \cdots \subset [S_m]$ such that $\dim([S_i]) = q_i$, for $i = 1, \ldots, m$. A **flag manifold**, denoted $\mathrm{FL}(n; q_1, q_2, \ldots, q_m)$, is the aggregate of all such flags.

If the set $\mathbb{I} = \{q_i\}$ is just one integer, then $\mathrm{FL}(n; q_i)$ is equivalent to $\mathrm{Gr}(q_i, n)$ and the points on either correspond to q_i-dimensional subspaces of $\mathbb{R}^n$. If $X = [x^{(1)}|x^{(2)}|\ldots|x^{(q)}]$ is an $n \times q$ matrix whose ith column is $x^{(i)}$, then the full flag living in $\mathrm{FL}(n; 1, 2, \ldots, q)$ created out of the vectors will be denoted with double square brackets as

$$[\![X]\!] = \mathrm{span}\{x^{(1)}\} \subset \mathrm{span}\{x^{(1)}, x^{(2)}\} \subset \ldots \subset \mathrm{span}\{x^{(1)}, \ldots, x^{(q)}\}. \tag{9}$$

For more explanation of flag manifolds and their geometry, refer to [17].

2.3. Existing subspace averages

2.3.1. Riemannian center of mass. There are a number of existing averages for subspace data, and more specifically, points on Grassmann manifolds. The most widely used representative for a point cloud on a Grassmannian is the Karcher mean, which is defined as the Riemannian center of mass for the cloud [13].

Definition Let $\mathbb{D} = \{[X_1], [X_2], \ldots, [X_p]\}$ be a finite set of points on $\mathrm{Gr}(q, n)$, such that the radius of $\mathbb{D} \leq \pi/4$. The sample **Karcher mean** of $\mathbb{D}$ is defined to be the solution to

$$[\mu_K] = \arg\min_{[\mu]} \frac{1}{P} \sum_{i=1}^{P} d([\mu], [X_i])^2 \tag{10}$$

for $[X_i] \in \mathbb{D}$. In this setting $d([\mu], [X_i]) = \|\Theta_i\|_2$.

It may be obvious that this average is the analogue of the Euclidean mean, since the geodesic distance based on arc length is the canonical metric on the Grassmann manifold as mentioned in Section 2.2, and we are finding the point that minimizes the sum of the squared of the distances between itself and the data. However, unlike the Euclidean mean, the Karcher mean does not have an analytic solution.

2.3.2. COMPUTING μ_K. The two most common methods for finding the sample Karcher mean are with a first-order gradient descent algorithm or with Newton's method. The algorithm employed in the applications of this thesis is the first-order gradient descent algorithm described in [2]. For completeness Algorithms 1 and 2, the Grassmannian Exp and Log maps, have been included as well. As mentioned, we choose to represent points on a $\mathrm{Gr}(q, n)$ with $n \times q$ orthonormal matrices.

Algorithm 1 $\mathrm{GrExp}([X], [Y])$

$U\Sigma V^T = \text{thin svd}(Y)$
$\mathrm{GrExp}([X], [Y]) = XV \cos\Sigma + U \sin\Sigma$

Algorithm 2 $\mathrm{GrLog}([X], [Y])$

$U\Sigma V^T = \text{thin svd}\left((I - XX^T)Y(X^TY)^{-1}\right)$
$\Theta = \tan^{-1}\Sigma$
$\mathrm{GrLog}([X], [Y]) = U\Theta V^T$

Let $\{[X_1], [X_2], \ldots, [X_P]\}$ be a finite set of points on $\mathrm{Gr}(q, n)$, with a radius $\leq \pi/4$. The sample Karcher mean is then computed by initializing the mean as a random data point, or for simplicity as the first data point, $[\mu_1] = [X_1]$. The data points are mapped into the tangent space at $[\mu_1]$ using Algorithm 1. The Euclidean mean of these tangent vectors points in the direction of steepest descent, δ. The estimate of the mean is then updated by moving on the Grassmannian in the direction of δ, i.e. $[\mu_2] = \mathrm{GrExp}(\mu_1, \delta)$. This process iterates

until there is no movement between $[\mu_i]$ and $[\mu_{i+1}]$. At this point, the Karcher mean of $\{[X_1], [X_2], \ldots, [X_P]\}$ is $[\mu_K] = [\mu_i]$. This process is detailed in Algorithm 3.

Algorithm 3 Karcher mean of $\{[X_1], [X_2], \ldots, [X_P]\}$

$\mu_1 = X_1$
while $d(\mu_i, \mu_{i+1}) > \epsilon$ **do**
  $\delta = \frac{1}{P} \sum_{j=1}^{P} \text{GrLog}(\mu_i, X_j)$
  $\mu_{i+1} = \text{GrExp}(\mu_i, \delta)$
end while
$[\mu_K] = \mu_{i+1}$

As a side note, if $[X_i]$ is not already represented by an orthogonal matrix, computing $U\Sigma V^T = \text{thin svd}(X_i)$ and using $X_i = UV^T$ gives us the closest point to X_i on $\text{Gr}(q, n)$. Additionally, the restriction of $\{[X_1], [X_2], \ldots, [X_P]\}$ to a ball of radius $\leq \pi/4$ guarantees that the GrExp and GrLog maps are bijective, and thus that the Karcher mean will find a unique optimal solution. As noted in [2, 13] this radius is the maximal size of a convex set on any Grassmannian, but as we proved in Proposition 2.2.1, the maximum distance between two points can be much larger than this.

2.3.3. Limitations of the Karcher mean. One of the key limitations of the Karcher mean is that it is found using an iterative algorithm. This leads to complications in larger machine-learning systems like CSU's contribution to the Mind's Eye project. Since there are multiple algorithms for finding the Karcher mean, it is hard to give a strict bound to the convergence. First-order gradient descent algorithms are typically reported as having linear convergence, while the Newton's method algorithms claim quadratic convergence [1, 2, 7, 26].

In either case, the number of iterations, $N_{d,\epsilon}$, depends heavily of the diameter of the data set, d, and the error tolerance, ϵ, so that even with efficient algorithms this calculation can be prohibitive for high-dimensional image and video data. To illustrate these dependencies, the convergence of the Karcher mean was tested using Algorithm 3 on $\text{Gr}(20, 1000)$. For

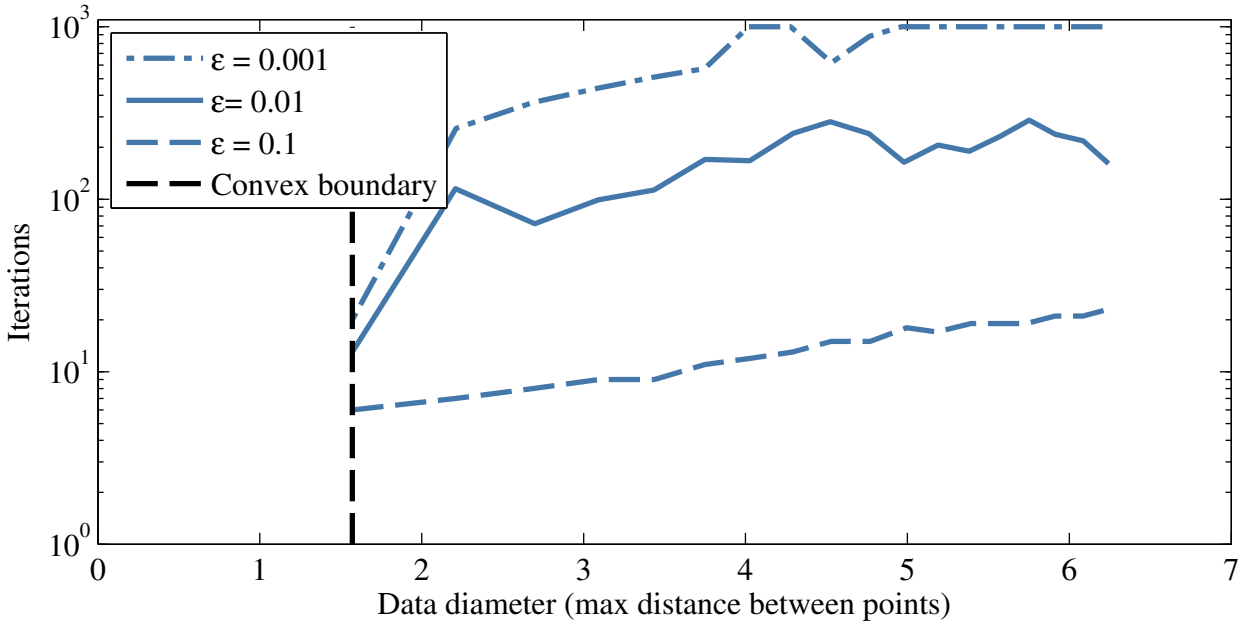


FIGURE 2.3. Iterations needed to find the Karcher mean for different values of ϵ.

each error tolerance, ϵ, 30 random points were created. The diameter, d, of this data set was measured as the furthest distance between any two points using the geodesic distance based on arc length, and μ_K was found. The maximum number of iterations allowed was set to 1000 so the algorithm would not run indefinitely. Figure 2.3 shows the number of iterations it took to find the Karcher mean of each point set when the diameter of the data is beyond the convex boundary. When $\epsilon = 0.1$, the algorithm takes around 40 iterations to find the Karcher mean for large values of d. This number continues to grow linearly as d grows. Similarly, for $\epsilon = 0.01$, the number of iterations appears to grow proportionately to d. At this tolerance the number of iterations required is around 200 when $d \to 6.5$. However for $\epsilon = 0.001$, and $d > 4$, it takes the maximum number of iterations to find the Karcher mean.

2.3.4. THE L_2-MEDIAN ON RIEMANNIAN MANIFOLDS. The median of a 1-dimensional data set in Euclidean space, $\{x_1, x_2, \ldots, x_p\}$ can be found as the minimizer of the sum of

Euclidean of distances. That is

$$\hat{\mu} = \arg\min_{\mu \in \mathbb{R}} \sum_{i=1}^{p} \|x_i - \mu\|_2. \tag{11}$$

While this definition has been agreed upon for the 1-dimensional case by mathematicians and statisticians alike, there has been no consensus on how to generalize the median into the multidimensional case. Several possible definitions which have the common property of producing the usual definition of the median in the uni-variate case, but give different kinds of medians in multivariate situations, are reviewed by Smalls in [23].

One incarnation that has appeared frequently in the literature as a multidimensional median is the point that minimizes the L_1-norm of the Euclidean distance between points. This version was first introduced by Gini and Galvani in 1929 [10], and by Eells a year later [8]. As the geodesic distance based on arc length is the Riemannian equivalent of the Euclidean distance, the definition for this point can be written in that context.

Definition For $\{[X_1], [X_2], \ldots, [X_p] | [X_i] \in \mathrm{Gr}(q, n)\}$, the point that minimizes the L_1-norm of the geodesic distance based on arc length between points,

$$[\mu_{L_2}] = \arg\min_{[\mu]} \sum_{i=1}^{P} d([\mu], [X_i]), \tag{12}$$

is the **$\mathbf{L_2}$-median** of the set.

The solution to Equation 12 is most commonly referred to as the spatial median, but has also been called the geometrical median [12], the mediancentre [11], the L_2-median [6], and confusingly, the L_1-median [23]. This thesis will refer to the point in Equation 12 as the L_2-median, and denote it as μ_{L_2} in Euclidean spaces and $[\mu_{L_2}]$ on Grassmannians, because

that terminology seems to best reflect the construction of the point as the minimizer of the L_1-norm of the vector of the L_2-norms between the data points and the median.

One method for finding μ_{L_2} in Euclidean spaces is called the Weiszfeld algorithm, although Weiszfeld did not create the algorithm or even intend to solve a location problem [29]. The algorithm is essentially a steepest descent method, and was generalized for Riemannian data, including Grassmannian data, by Fletcher *et al.* in 2009 [9]. Fletcher's modified algorithm, elucidated in Algorithm 4, employs the geodesic distance based on arc length, Algorithm 1, and Algorithm 2.

Algorithm 4 L_2-median of $\{[X_1], [X_2], \ldots, [X_P]\}$

$\mu_1 = X_1$
while $d(\mu_i, \mu_{i+1}) > \epsilon$ **do**
  $\delta = \sum_{j=1}^{P} \frac{\text{GrLog}(\mu_i, X_j)}{d(\mu_i, X_j)} * \left(\sum_{i=1}^{P} \frac{1}{d(\mu_i, X_j)} \right)^{-1}$
  $\mu_{i+1} = \text{GrExp}(\mu_i, \delta)$
end while
$[\mu_{L_2}] = \mu_{i+1}$

The L_2-median has convergence issues similar to those of the Karcher mean, because of the dependence on GrLog and GrExp. Thus a unique optimal solution to Equation 12 is only guaranteed when all points being averaged are within a ball of radius $\leq \pi/4$. Similarly, the number of iterations needed to find an optimal solution mimics those depicted for the Karcher mean in Figure 2.3, i.e. as the diameter of the data set grows the number of iterations needed grows proportionally.

2.3.5. The extrinsic manifold mean. A third average for points on a Grassmann manifold is the extrinsic manifold mean. First published by Srivastava and Klassen in 2002, the extrinsic manifold mean is an L_2-mean in that it looks to minimize the L_2-norm of the vector of distances [24]. The main difference between the Karcher mean and the extrinsic

manifold mean is that the Karcher mean uses the geodesic distance based on arc length, whereas the extrinsic mean relies on the projection F-norm.

Definition Let $\{[X_1], [X_2], \ldots, [X_P]\}$ be a finite collection of points on $\mathrm{Gr}(q, n)$. Srivastava and Klassen define the **extrinsic manifold mean**, $[\mu_E]$, as

$$[\mu_E] = \arg\min_{[\mu]} \frac{1}{P} \sum_{i=1}^{P} d_{pF}([\mu], [X_i])^2, \tag{13}$$

where $d_{pF}([\mu], [X_i])$ is again the projection Frobenius norm.

As mentioned in Section 2.2, the projection F-norm is realized by embedding the Grassmannian data points in the space of $n \times n$ projection matrices of rank q. By embedding $\mathrm{Gr}(q, n)$ in a higher dimensional space, this projection F-norm can be thought of as cutting corners in measuring distances between points. Algorithm 5 for finding $[\mu_E]$ comes from the method of Lagrange multipliers and depends on the second equality in Equation 4.

Algorithm 5 Extrinsic manifold mean of $\{[X_1], [X_2], \ldots, [X_P]\}$

$\mathbf{C} = \sum_{i=1}^{P} X_i X_i^T$
for j = 1 ... n **do**
  $\lambda^{(j)} u^{(j)} = \mathbf{C} u^{(j)}$
  Such that $\lambda^{(1)} > \lambda^{(2)} > \cdots > \lambda^{(n)}$
end for
$[\mu_E] = \left[\mathrm{span}\{u^{(1)}, u^{(2)}, \ldots, u^{(q)}\}\right]$

The result of Algorithm 5 is that $[\mu_E]$ is effectively the dominant q-dimensional eigenspace of the projection matrix representations of the original Grassmannian data points. The matrix $\mathbf{C}$ is an $n \times n$ real, symmetric, matrix, and $\mathrm{rank}(\mathbf{C}) \geq q$. Thus the eigenvectors form an orthonormal set, and the span of the q eigenvectors corresponding to the largest eigenvalues will indeed be a point on $\mathrm{Gr}(q, n)$. It will become evident in Section 3 that the

flag mean is a generalization in two directions of the extrinsic manifold mean, but interested readers may refer to [24] for the derivation of Algorithm 5.

2.3.6. Discussion of existing averages. After presenting the most common subspace averages in the literature, it is relevant to briefly discuss their similarities and differences. The most notable difference between $[\mu_{L_2}]$ and the others, is that it minimizes the L_1-norm of the vector of distances while $[\mu_K]$ and $[\mu_E]$ minimize the L_2-norm of their vectors of distances. This difference is significant, because it ensures that $[\mu_{L2}]$ is robust to outliers. The robustness of a location estimator is often measured using the finite sample breakdown point. Although it will not be defined explicitly, the finite sample breakdown point essentially measures how many samples can be corrupted before the location estimator is also corrupt. The breakdown point is trivially defined for bounded manifolds because points cannot be corrupted without bound. However, on more general Riemannian manifolds it can be shown that at least half the data needs to be corrupted to infinity for $[\mu_{L_2}]$ to be pulled to infinity. In contrast, if only one sample is pulled to infinity, both $[\mu_K]$ and $[\mu_E]$ get skewed to infinity as well. The proofs of these statements and more details about finite sample breakdown points can be found in [9].

In terms of computation, the previous sections illustrated how $[\mu_K]$ and $[\mu_{L_2}]$ are found through iterative algorithms. For P points in $\mathrm{Gr}(q, n)$, the time to compute either $[\mu_K]$ or $[\mu_{L_2}]$ takes $O(Pnq^2N_{d,\epsilon})$ flops. On the other hand, Algorithm 5 shows that $[\mu_E]$ is the analytic solution to Equation 13, and can be found by computing eigenvalue decomposition of a real, symmetric $n \times n$ matrix. Thus the complexity of finding $[\mu_E]$ is $O(n^3)$. Additionally, the computation of the extrinsic mean is independent of the distance between data points or any error tolerance.

CHAPTER 3

Presentation and Derivation

3.1. Motivation for an additional subspace average

After the previous section, it may seem that a subspace average exists to suit every scenario or need. However, there are a number of challenges that remain to be addressed. The first is the iterative nature of the algorithms for finding $[\mu_K]$ and $[\mu_{L_2}]$. When dealing with high dimensional image and video data, the time required to compute either of these averages becomes prohibitive quickly. The computation time for $[\mu_E]$ can be significantly less, depending on the make-up of the data set. More importantly though, the solution is analytic, and thus a good approximation of the computation time can be calculated.

It is also possible in practical experiments that data points do not live within a convex ball on their Grassmann manifold. In this scenario, the algorithms for the Karcher mean and the L_2-median will return a solution, but the solution may not be a unique optimum. As an analogy, imagine a data set consisting of two antipodal points on a sphere. To find a point averaging those two locations while staying on the sphere, any point on the equator could be reasonably chosen. However, if that equatorial point was then used in a measure of distance to other points on the sphere, the distance computed would be a very poor measure of how close those points were to the set of antipodal points. Again, the extrinsic manifold mean does not fall prey to this limitation. The points to be averaged need not be within a convex set, because of the extrinsic mean's nature as an embedding. However, in the rare case where the qth-eigenvalue of the matrix $\mathbf{C} = \sum_{i=1}^{P} X_i X_i^T$ has multiplicity greater than 1, the optimal solution may not be unique.

A third hurdle was realized as a result of the CSU team's approach to the Mind's Eye Project. When cutting video clips to create points on a Grassmannian, it was difficult to determine how many frames would be needed to create a subspace of the desired dimension. Video clips were cut with the poor assumption that all frames would be linearly independent. When this failed, the closest point on the manifold was found. However, this closest point is not unique. Thus a need arose for a way to average points that live on a collection of Grassmann manifolds, as none of the existing averages could accommodate such data.

3.2. Derivation

The places listed previously where existing subspace means leave something to be desired are the areas where the flag mean will hopefully add value. The derivation will begin by finding a set of vectors central to a collection of subspaces, which will then be used to create a flag. Let $\{[X_i]\}_{i=1}^{P}$ be a collection of subspaces of $\mathbb{R}^n$ such that $\dim([X_i]) = q_i$ and $X_i^T X_i = I$. If $\mathbb{I} = \{q_1, \ldots, q_P\}$, with the q_i's not necessarily distinct, then $[X_i] \in \coprod_{\mathbb{I}} \mathrm{Gr}(q_i, n)$ for $i = 1 \ldots P$. For this collection of subspaces we wish to find the one-dimensional subspace $[u^{(1)}] \in \mathrm{Gr}(1, n)$ that minimizes the sum of the squares of projection F-norms between $[u^{(1)}]$ and $[X_i]$ for $i = 1 \ldots P$. The projection F-norm loses its distinction as a metric when it 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 measuring similarity between the two objects. Thus we aim to solve

$$\begin{aligned} &\underset{[u^{(1)}]}{\arg\min} && \sum_{i=1}^{P} d_{pF}([u^{(1)}], [X_i])^2 \\ &\text{subject to} && u^{(1)T} u^{(1)} = 1. \end{aligned} \tag{14}$$

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 $[u^{(1)}]$, the problem is extended to find a sequence of optimizers to Equation 14 with additional constraints. By solving

$$\begin{aligned} \underset{[u^{(j)}]}{\arg\min} \quad & \sum_{i=1}^{P} d_{pF}([u^{(j)}],[X_i])^2 \\ \text{subject to} \quad & u^{(j)T}u^{(j)} = 1 \\ & u^{(j)T}u^{(k)} = 0 \qquad \text{for } k < j, \end{aligned} \tag{15}$$

it is possible to find $\{[u^{(1)}],[u^{(2)}],\ldots,[u^{(r)}]\}$ where r is the dimension of the span of the $\{[X_i]\}_{i=1}^P$. These subspaces are then central to the collection of points $\{[X_i]\}_{i=1}^P$, and will be used in the construction of the flag mean.

Definition The **flag mean** of $\{[X_i]\}_{i=1}^P$, denoted $[\![\mu_{pF}]\!]$, is a point on the flag manifold $\mathrm{FL}(n;1,2,\ldots,r)$ where $r = \dim(\mathrm{span}(\{[X_i]\}_{i=1}^P))$. Each subspace in $[\![\mu_{pF}]\!]$ provides an average for $\{[X_i]\}_{i=1}^P$ that lives on a Grassmann manifold of the appropriate dimension. The flag mean is built from the ordered 1-dimensional subspaces that optimize Equation 15, $\{[u^{(1)}],...,[u^{(r)}]\}$. From this sequence of mutually orthogonal vectors, the flag mean is defined explicitly as

$$[\![\mu_{pF}]\!] = \mathrm{span}\{u^{(1)}\} \subset \mathrm{span}\{u^{(1)},u^{(2)}\} \subset \ldots \subset \mathrm{span}\{u^{(1)},\ldots,u^{(r)}\}. \tag{16}$$

3.2.1. Simplifying the optimization problem. As presented, the solutions to the optimization problem in Equation 15 give an intuitive average for the collection of subspaces $\{[X_i]\}_{i=1}^P$, because they optimize a similar cost function to that of the widely used Karcher

mean and the same cost function as the lesser known extrinsic mean. The methods employed in Karcher mean algorithms are unappealing for this problem because they would require solving r separate steepest descent or Newton's method algorithms. Instead we apply the equality $d_{pF}([X],[Y]) = \|\sin\Theta\|_2$ from Equation 4 to find a fast, analytic solution to the problem similar to how Srivastava and Klassen solved for $[\mu_E]$ in [24].

Let $\theta_i^{(j)}$ be the lone principal angle between $[u^{(j)}]$ and $[X_i]$. For $j = 1,\ldots,r$ we can rewrite the cost function of Equation 15 as,

$$\arg\min_{[u^{(j)}]} \sum_{i=1}^{P} d_{pF}([u^{(j)}],[X_i])^2 = \arg\min_{[u^{(j)}]} \sum_{i=1}^{P} \|\sin\theta_i^{(j)}\|_2^2 \tag{17}$$

$$= \arg\max_{[u^{(j)}]} u^{(j)T}\left(\sum_{i=1}^{N} X_i X_i^T\right) u^{(j)}. \tag{18}$$

The equality between Equation 17 and Equation 18 follows from the singular value decomposition of $u^{(j)T}X_i$.

Substituting Equation 18 in as the new cost function transforms the optimization problem from Equation 15 into

$$\begin{aligned} \arg\max_{[u^{(j)}]} \quad & u^{(j)T}\left(\sum_{i=1}^{N} X_i X_i^T\right) u^{(j)} \\ \text{subject to} \quad & u^{(j)T}u^{(j)} = 1 \\ & u^{(j)T}u^{(k)} = 0 \qquad \text{for } k < j. \end{aligned} \tag{19}$$

3.2.2. Solving for the flag mean. To find $\{[u^{(1)}],[u^{(2)}],\ldots,[u^{(r)}]\}$, first define $\mathbf{A} = \sum_{i=1}^{N} X_i X_i^T$. For $u^{(1)}$, the problem can be turned into the Lagrangian

$$L(u^{(1)},\lambda^{(1)}) = u^{(1)T}\mathbf{A}u^{(1)} - \lambda^{(1)}(u^{(1)T}u^{(1)} - 1). \tag{20}$$

The partial derivatives of $L(u^{(1)}, \lambda^{(1)})$ are then

$$\begin{aligned} \frac{\partial L}{\partial u^{(1)}} &= 2\mathbf{A}u^{(1)} - 2\lambda^{(1)}u^{(1)} \\ \text{and} \quad \frac{\partial L}{\partial \lambda^{(1)}} &= u^{(1)T}u^{(1)} - 1, \end{aligned} \tag{21}$$

so that the first order necessary conditions for optimality are satisfied when

$$\begin{aligned} \mathbf{A}u^{(1)} &= \lambda^{(1)}u^{(1)} \\ \text{and} \quad u^{(1)T}u^{(1)} &= 1. \end{aligned} \tag{22}$$

Thus the solution to the eigenvector problem $\mathbf{A}u^{(1)} = \lambda^{(1)}u^{(1)}$ will maximize the cost function when $u^{(1)}$ is the eigenvector associated with the largest eigenvalue. Once $u^{(1)}$ has been found, each consecutive $u^{(j)}$ can be computed by solving a Lagrangian with additional constraints to guarantee that $u^{(j)T}u^{(k)} = 0$ for each $k < j$. Since $\mathbf{A}$ is a real, symmetric matrix, there are $r = \dim(\mathbf{A})$ mutually orthogonal eigenvectors associated with non-zero eigenvalues. If these eigenvectors are ordered by their associated eigenvalues in descending order the resulting sequence is $\{[u^{(1)}], ..., [u^{(r)}]\}$, or the sequential optimizers of Equation 19. Returning to the definition of the flag mean presented in Equation 16, these dominant r eigenvectors of $\mathbf{A}$ are then used to construct $[\![\mu_{pF}]\!]$.

3.2.3. Alternative computation via the SVD. Finding the r mutually orthogonal eigenvectors of $\mathbf{A}$ can be done in $O(n^3)$ flops with standard eigenvector solvers. If we exploit the properties of the singular value decomposition that computation can be sped up in many cases.

Let $Q = \sum_{i=1}^{P} q_i$ and define $\mathbf{X} = [X_1|X_2|\dots|X_P]$, so that $\mathbf{X} \in \mathbb{R}^{n\times Q}$. The thin SVD of the bases matrix $\mathbf{X}$ is $\hat{U}\hat{\Sigma}\hat{V}^T$, where $\hat{U} = [\hat{u}^{(1)}|\dots|\hat{u}^{(r)}]$ is the $n \times r$ orthonormal matrix

of left singular vectors, $\hat{\Sigma}$ is an $r \times r$ diagonal matrix whose entries, $\hat{\sigma}^{(1)}, \ldots, \hat{\sigma}^{(r)}$, are the singular values, and $\hat{V} = [\hat{v}^{(1)}| \ldots |\hat{v}^{(r)}]$ is the $n \times r$ matrix of right singular vectors.

Note that $\mathbf{X}\mathbf{X}^T = \sum_{i=1}^{P} X_i X_i^T = \mathbf{A}$, and by using the thin SVD representation of $\mathbf{X}$, we can rewrite

$$\mathbf{X}\mathbf{X}^T = \hat{U}\hat{\Sigma}^2\hat{U}^T. \tag{23}$$

Thus up to a change in sign, the r left singular vectors of $\mathbf{X}$ are equal to the first r eigenvectors of $\mathbf{X}\mathbf{X}^T$, and are exactly the solutions to the optimization problem in Equation 19, $\{[u^{(1)}], ..., [u^{(r)}]\}$. By solving for the singular value decomposition of $\mathbf{X}$ instead of an eigenvalue decomposition of $\mathbf{A}$ the complexity changes to $O(nQ^2)$ flops, which is typically less than $O(n^3)$ for image and video data.

CHAPTER 4

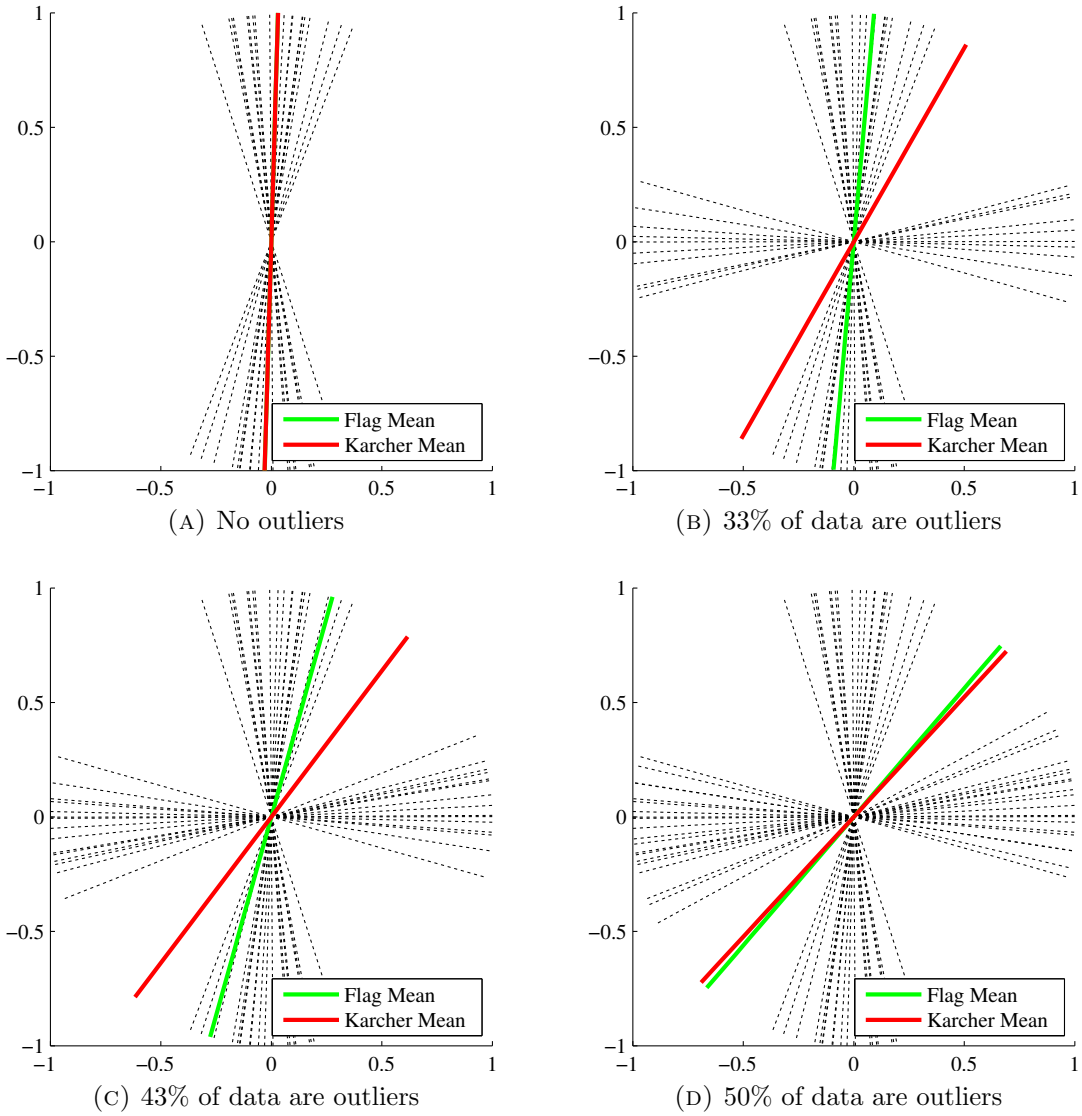
Results and Analysis

The original task that led to the creation of the flag mean was to purify clusters of videos for the Mind's Eye Project. Thus early experiments with the flag mean are characterized in that context. This chapter of the thesis will explore applications of the flag mean. The first section contains synthetic experiments that explore that attempt to build intuition about what individual subspaces contained in $[\![\mu_{pF}]\!]$ represent geometrically. The second section contains direct comparisons to the Karcher mean on practical experiments, because the Karcher mean is the Computer Vision industry standard for averaging subspaces. The third section contains applications with the Carnegie Mellon University, 'Pose, Illumination, and Expression' database.

4.1. Synthetic illustrations

This section illustrates how the flag mean differs from the Karcher mean and extrinsic manifold mean when input data is restricted to a single Grassmann manifold. In these examples, the full flag from $[\![\mu_{pF}]\!]$ is not used. Instead, individual subspaces contained within $[\![\mu_{pF}]\!]$ are examined, and the one that best classifies the data is used. The first data set shows that the flag mean may be less sensitive to outliers than the Karcher mean, despite the inability to compute the finite sample breakdown point as explained in Subsection 2.3.6. The second one shows how the flag mean is more flexible than the extrinsic manifold mean in the presence of structured data.

4.1.1. Data fitting (robustness). The first data set shows how the flag and Karcher means represent data on $\mathrm{Gr}(1,2)$, i.e. lines in 2-space, that contain outliers. The data



(A) No outliers

(B) 33% of data are outliers

(C) 43% of data are outliers

(D) 50% of data are outliers

FIGURE 4.1. Behavior of the flag mean (green) and Karcher mean (red) when outliers are present.

consists of two vectors chosen to be true means, $\hat{\mu}_1 = [0, 1]^T$ and $\hat{\mu_2} = [1, 0]^T$, along with points sampled from a normal distribution around each true mean with a standard deviation of 0.2. We begin by choosing 30 points around $\hat{\mu}_1$ and no points around $\hat{\mu}_2$. We then generate the Karcher mean, $[\mu_K]$, and the 1-dimensional subspace contained in $[\![\mu_{pF}]\!]$, of the set of points. Since the angles between the points are fairly small the two means are almost equivalent, as shown in Figure 4.1a.

Next, we introduce outliers by adding 15 points around $\hat{\mu}_2$ to the 30 we already have around $\hat{\mu}_1$ and calculate the means for the resulting 45 points. The term 'outlier' may be slightly abused in this experiment. The goal is to have $[\mu_K]$ and $[\![\mu_{pF}]\!]$ represent $\hat{\mu}_1$ as closely as possible. Thus the term outlier is used here to mean that the data from the distribution about $\hat{\mu}_2$ that is pulling $[\mu_K]$ and $[\![\mu_{pF}]\!]$ away from $\hat{\mu}_1$. Figure 4.1b shows that the Karcher mean is pulled towards the second distribution, while the flag mean is closer to the mean of the larger distribution. The same is true for 23 outlier points, as shown in Figure 4.1c. In Figure 4.1d the number of points about $\hat{\mu}_2$ is increased to 30, meaning that the two distributions are now equally represented. In this case there are no real outliers and $[\mu_K] \approx [\![\mu_{pF}]\!]$. Overall, when the data can be described as samples from a dominant class plus outliers, the flag mean is less influenced by the outliers than the Karcher mean and is therefore more robust. When the samples are drawn from a single distribution or an equal number of samples are drawn from two distributions, the two means are approximately the same.

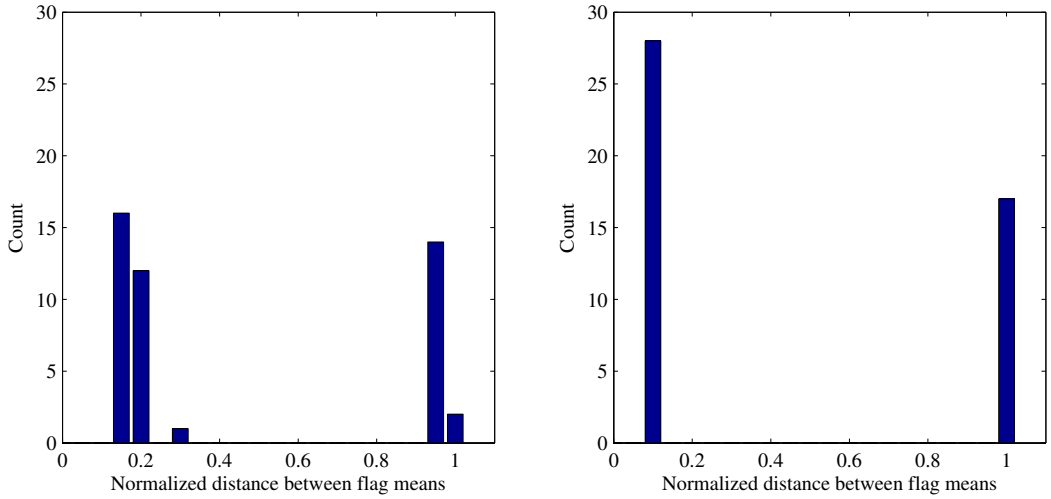
4.1.2. True data dimension (flexibility). The extrinsic manifold mean is limited by two requirements: the input data must live on a single Grassmann manifold and the resulting mean must live on the same manifold. The inability of the extrinsic manifold mean to handle subspaces of different dimensions is discussed in the real-world experiments of Section 4.2. This section explores the limitation that $[\mu_E]$ must live on the same Grassmann manifold as the original data. In particular, this lack of flexibility is a problem when the data set has many dimensions of noise in addition to the signal dimensions, because the extrinsic manifold mean must generate a subspace of the same dimension as the data and therefore include the noise.

For this illustration we fix a 2-dimensional subspace of $\mathbb{R}^{100}$, $[w]$. We then generate points on $\mathrm{Gr}(10, 100)$ by letting $[Y_{(i,j)}] = \mathrm{span}\{w, w_{(i,j)}\}$, where $w_{(i,j)}$ is an orthonormal matrix representative for a random 8-dimensional subspace of $\mathbb{R}^{100}$ for $i = 1 \dots 9$ and $j = 1 \dots 20$. We then set $\mathbf{D_i} = \{[Y_{(i,1)}], \dots, [Y_{(i,20)}]\}$ in order to form $\mathbf{D_1}, \mathbf{D_2}, \dots, \mathbf{D_9}$. We let $\mathbf{D_{10}}$ be 20 random 10-dimensional subspaces of $\mathbb{R}^{100}$. The resulting data set consists of nine sets whose elements share a common 2-dimensional subspace, and one set that is pure noise.

We now create a flag mean for each of these point sets, $[\![\mu_{pF}]\!](\mathbf{D_1}), \dots, [\![\mu_{pF}]\!](\mathbf{D_{10}})$. Thus each mean is a flag, and consists of a nested sequence of vector spaces. The histograms in Figure 4.2 show the normalized pairwise distances between the k-dimensional subspaces in each flag of $\mathbf{D_1}, \dots, \mathbf{D_{10}}$ for $k = 1, 2, 5, 10$. These values are calculated using the projection F-norm, and in each histogram the distances have been divided by the maximum distance so we can compare across dimensions.

We see that when the dimension of the flag mean is low, there are two distinct groups of distances. The group of small distances are the pairwise ones between $\mathbf{D_1}, \dots, \mathbf{D_9}$. Since the points in these sets all contain the 2-dimensional subspace $[w]$ in their span, their low dimensional means are very close. The set of large distances are those between $\mathbf{D_{10}}$ and everything else. $\mathbf{D_{10}}$ was created to be an anomaly in our set so we could observe which mean best represented our full data set. We can see in Figure 4.2d that $[\mu_E]$, the extrinsic manifold mean (which is also the 10-dimensional $[\![\mu_{pF}]\!]$) does not recognize the mean of $\mathbf{D_{10}}$ as an anomaly.

4.1.3. Fisher's Discriminant Analysis. This experiment continues down the same line of thought as the previous subsection. The data is generated in similar way, except that this time we fix two 2-dimensional subspaces of $\mathbb{R}^{100}$, $[w]$ and $[v]$. From these we generate



(A) 1-dim. subspace of $[\![\mu_{pF}]\!]$

(B) 2-dim. subspace of $[\![\mu_{pF}]\!]$

(C) 5-dim. subspace of $[\![\mu_{pF}]\!]$

(D) 10-dim. subspace of $[\![\mu_{pF}]\!] = [\mu_E]$

FIGURE 4.2. Histograms of the pairwise distances between the k-dimensional subspaces in the flag means.

two classes of data, $\hat{\mathbf{D}} = \mathbf{D_1}, \mathbf{D_2}, \ldots, \mathbf{D_{10}}$ and $\hat{\mathbf{E}} = \mathbf{E_1}, \mathbf{E_2}, \ldots, \mathbf{E_{10}}$. For each $\mathbf{D_i}$ we have 20 points created as $\text{span}\{w, w_{(i,j)}\}$ where $w_{(i,j)}$ is a orthonormal matrix representative for a random 8-dimensional subspace of $\mathbb{R}^{100}$, and for each $\mathbf{E_i}$ we have 20 points of the form $\text{span}\{v, v_{(i,j)}\}$ where $v_{(i,j)}$ is a orthonormal matrix representative for a random 8-dimensional subspace of $\mathbb{R}^{100}$.

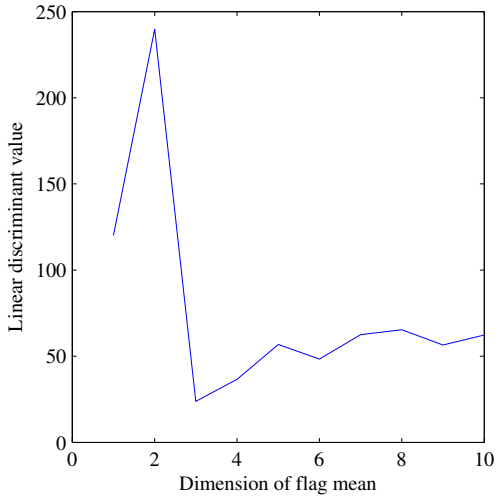


FIGURE 4.3. Fisher's discriminant calculated for k-dimensional flag means of $\mathbb{R}^{100}$.

From the sets of points in these two classes we again generate flag means. This time we use Fisher's discriminant to determine how tightly the classes of means are clustered. From the previous experiment one would assume that when we use 2-dimensional flag mean for each set of points, the means from class $\hat{\mathbf{E}}$ and class $\hat{\mathbf{D}}$ would be tightly clustered with the means from their own class and Fisher's discriminant would be high. This is in fact the case, as is illustrated in Figure 4.3. Once again the 10-dimensional flag mean, or the extrinsic manifold mean, does a very poor job of separating the two classes of data because it is forced to represent the 8-dimensions of noise.

4.2. Comparisons to the Karcher mean on practical experiments

While the preceding section used synthetic data to illustrate the differences between some of the means, this section tests the flag mean and the Karcher mean on complex real-world tasks. In particular, two tasks that can benefit by computing means of vector subspaces are exemplar selection and clustering. Exemplar selection is the task of selecting the most prototypical sample in a set, where prototypical is defined as closest to the mean. The k-means

clustering algorithm computes cluster means as a prelude to measuring distances between samples and clusters. There are other clustering algorithms that do not require computing sample means, but they are not addressed in this thesis. We compare the flag and Karcher means on exemplar selection and k-means clustering applied to subspaces representing video clips. On both tasks, the flag mean provides comparable or more accurate results than the alternative in a fraction of the time.

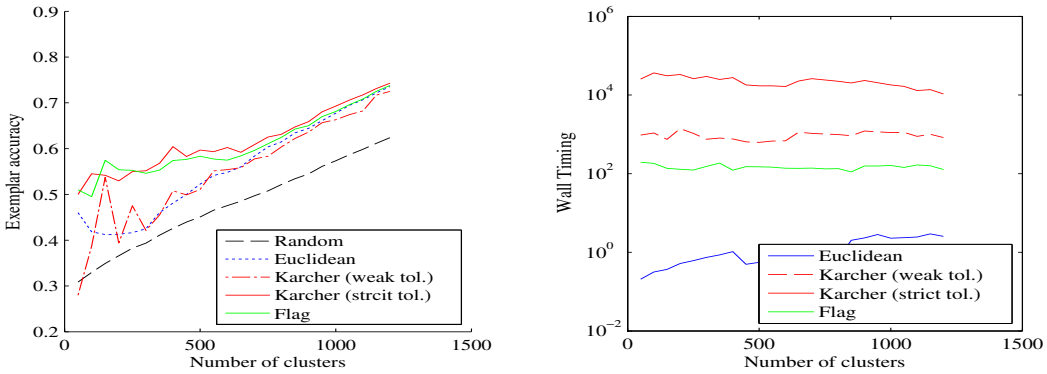
In these tasks, we have again selected a single subspace from within the flag, $[\![\mu_{pF}]\!]$. In Subsection 4.1.2 and Subsection 4.1.3 it was known ahead of time how many dimensions of the data were meaningful and how many were noise. In these practical experiments the dimension of the subspace from $[\![\mu_{pF}]\!]$ selected was not known a priori to be be the hidden dimension of the data. It was chosen because it provided the best results. One advantage of the flag mean was that information about all the constituent subspaces within $[\![\mu_{pF}]\!]$ is known without additional computations, so the best subspace was simply selected as a showcase.

4.2.1. Mind's Eye data. The data set contains $2,345$ short videos 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 selected through background subtraction to be centered on a person, although the background subtraction is imperfect and sometimes only part of the person 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 2.1 in Section 2.1 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.).

For the purposes of these experiments, tracks are treated as 3-dimensional $32 \times 32 \times 48$ tensors with axes corresponding to width, height, and frame number. The tensors are 'unfolded' along each of their axes to create 3 different matrix representations of the data as depicted in Figure 2.2. The traditional unfolding along the temporal axis turns each tensor into a matrix of size 1024×48; the other two unfoldings create 1536×32 matrices. In these experiments, the flag and Karcher means are used to compute track means in each of the three possible unfoldings. Following [16], the distance between two tracks is the product manifold distance. This means that the distance between tracks T_1 and T_2 is the sum of the squares of the geodesic distances based on arc length in the three unfoldings.

One might expect that each unfolding would produce data points on a single Grassmann manifold; for example, that the temporal unfolding would produce points on $\mathrm{Gr}(48, 1024)$. Often, however, the matrices are not full rank. Physically, this means that some of the frames in the track are linearly dependent. This is not a problem when computing the flag mean, because the flag mean does not require that the subspaces being averaged span the same number of dimensions. It is a problem for the Karcher mean, however. Therefore, when computing the Karcher mean we replace rank-deficient samples with the nearest point on the appropriate Grassman manifold.

4.2.2. Exemplar selection. The first task is to choose exemplars from sets of tracks. On each trial, the system is given a set of similar tracks, and computes the mean of the set using the flag and Karcher means. For each mean it then selects the closest sample to the mean as an exemplar. Since the goal is to find exemplars that represent the set well, an automatically selected exemplar is considered 'correct' if the action label associated with the



(A) Exemplar selection accuracy (B) Exemplar selection timing

FIGURE 4.4. Rate at which exemplars chosen match the dominant class of a cluster for various meant types.

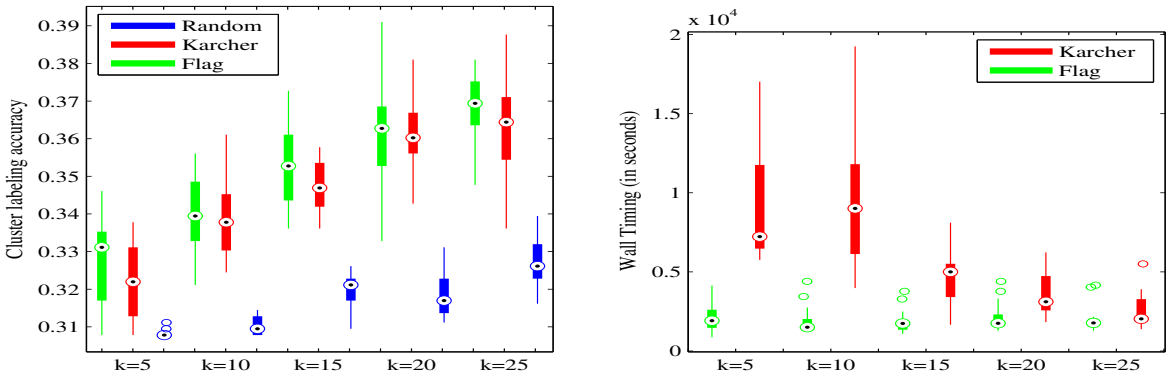
exemplar is the most common action label in the set, and 'incorrect' otherwise. The quality of a mean is measured by how often it predicts a correct exemplar.

Some methodological details. First, there are different algorithms for computing the Karcher mean. We implemented Algorithm 3. Second, the Karcher mean in particular is sensitive to how similar the samples being averaged are to each other. We therefore formed similar sets of tracks by clustering. To avoid interactions between the exemplar selection method and the clustering algorithm, we used an algorithm that does not require computing means, namely agglomerative clustering with Ward's linkage. Third, the Karcher mean is computed by an iterative algorithm that requires a convergence threshold. We tested the Karcher mean with a strict convergence tolerance of $\epsilon = 0.01$ and a weak tolerance of $\epsilon = 1$. Fourth, the flag mean contains a variable number of nested subspaces depending on the dimension of the data being averaged; we empirically chose the 10-dimensional subspace in $[\![\mu_{pF}]\!]$ as the best subspace for classification. Finally, as baselines for comparison we also selected exemplars randomly, and according to their distance from the Euclidean mean.

Figure 4.4a illustrates how often the exemplar label matches the label of the dominant action in a cluster. We see that the flag mean (green) is competitive with the strict tolerance Karcher mean (solid red). It is well known the the Euclidean mean (blue) is poorly suited to averaging data on Grassmann manifolds, and yet the Karcher mean with a weak convergence tolerance (dashed red) does almost as poorly. We also notice that the accuracy for each mean increases as the number of clusters increases. This happens when the average number of tracks in a cluster approaches 2, making either track an equally valid exemplar.

Figure 4.4b shows the time needed to compute each set of means. Cluster sizes ranged from 50 to 1200 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 147 seconds to compute the flag mean for a set of clusters, and 3.53×10^3 seconds or about an hour in total. In contrast, it took 2.222×10^4 seconds on average to create the Karcher means with the strict tolerance, which is the line in Figure 4.4a that is comparable to the flag mean in accuracy. In total, calculating the strict Karcher means took 5.333×10^5 seconds, or over 148 hours. Even for the Karcher mean with the weak tolerance, computation time was more than that of the flag mean. The weak tolerance Karcher took 941.72 seconds on average and 2.26×10^4 seconds or 6.278 hours total.

4.2.3. Naive k-means clustering. The second task is k-means clustering. This algorithm is well-known and 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. In this experiment, we first select 601 videos to cluster from the original set of $2,345$. There are two reasons for this. First, the original data set has one label ("walk", with 637 instances) that dominates all



(A) k-means accuracy (B) k-means timing

FIGURE 4.5. Comparison of k-means clustering on the Mind's Eye data set using different means.

others. Second, it was simply not feasible to evaluate k-means clustering with the Karcher mean even with a loose tolerance threshold on $2,345$ samples. The 601 videos that were chosen for this experiment have 17 unique labels. The largest class has 187 members and the smallest has 3.

In the first step of the k-means clustering algorithm we initialize k centers randomly chosen from our data set. Distances are then calculated between all points and each center, and points are assigned to the closest center. In the second step, means are calculated from the data points assigned to each cluster, distances are calculated between the means and all points, and each point is re-assigned to the mean nearest it. We allow Step 2 to iterate until the clusters have stabilized, that is, 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 4.5a, we see the cluster purity for the k-means clusters made using the 10-dimesional flag mean, the Karcher mean with weak convergence tolerance ($\epsilon = 1$), and random samples used as means. The clustering was performed 20 times for each value of k. The cluster purity is not high for either the flag or Karcher mean, indicating that the data set is very challenging, but both means beat the random baseline by a significant amount. As we see in Figure 4.5b, computation time was an order of magnitude bigger for the Karcher mean when the number of clusters was small. The difference in time decreases as the number of clusters grows. As the average number of tracks in a cluster shrinks, so does the diameter of the point set on the Grassmann manifold. This in turn improves the convergence of the Karcher mean, and can account for this decrease in time. The computation time for the Karcher mean with a strict convergence tolerance was infeasible on a data set of this size.

4.3. Computations with flags

The aim of this section is to compare flags to flags and subspaces to flags in a way that respects the structure of a flag as a nested sequence. The order of the subspaces within a flag is important. The earlier work, which presented in Section 4.2, disregarded much of the structure of the flag because it was desirable at that time to remain in the familiar territory of comparing subspaces to subspaces. This section contains more recent material that works directly with the flags, and attempts to exploit the order of the nested sequence. To compare flags, we need to define metrics between them. Grassmannian metrics can be intuitively generalized into flag metrics if the flags live on a single flag manifold. As defined in Subsection 2.2.3, two flags live on the same manifold if the subspaces in their nested sequence have compatible dimensions.

Definition Let $\mathbb{I} = \{q_1, q_2, \ldots, q_m\}$ be a collection of integers and let $\mathrm{FL}(n; q_1, q_2, \ldots, q_m)$ be the associated flag manifold. Suppose that $[\![X]\!], [\![Y]\!] \in \mathrm{FL}(n; q_1, q_2, \ldots, q_m)$. The **flag distance based on arc length** between $[\![X]\!]$ and $[\![Y]\!]$, denoted $d([\![X]\!], [\![Y]\!])$, is the sum of the Grassmannian distances between each of the constituent subspaces in the two flags. That is,

$$d([\![X]\!], [\![Y]\!]) = \sum_{k \in \mathbb{I}} d([X]^{(k)}, [Y]^{(k)}), \tag{24}$$

where $[X]^{(k)}$ means the k-dimensional subspace contained within $[\![X]\!]$. The format of this definition can generalize all of the Grassmannian distance measures by substituting the appropriate metric into Equation 24. For example, the **flag projection Frobenius norm** is defined similarly as

$$d_{pF}([\![X]\!], [\![Y]\!]) = \sum_{k \in \mathbb{I}} d_{pF}([X]^{(k)}, [Y]^{(k)}). \tag{25}$$

In practice the flags generated from PIE images are full flags, that is, their index set $\mathbb{I}$ contains all of the integers between 1 and some number q. However, that number q is not fixed. An extreme example is that a flag could be created out the span of a single image vector, i.e., a point on $\mathrm{Gr}(1, n)$. The flag generated by this single point then lives on $\mathrm{FL}(n; 1)$. This flag cannot be compared to a flag on $\mathrm{FL}(n; 1, \ldots, q)$ for any $q > 1$ with the metric in Equation 24. Thus, we will define similarity scores that have many of the properties of metrics but allows us to compare flags from different flag manifolds.

Definition Suppose $[X] \in \mathrm{Gr}(q, n)$, and $[\![Y]\!] \in \mathrm{FL}(n; p_1, \ldots, p_m)$. For $j = 1, \ldots, m$, let θ_j be the smallest principal angle between $[X]$ and the p_j-dimensional subspace in $[\![Y]\!]$, $[Y]^{(p_j)}$.

The **first-angle similarity score between subspaces and flags** is defined as

$$\tilde{d}([X], [\![Y]\!]) = \sqrt{\sum_{j=1}^{m} \theta_j^2}. \tag{26}$$

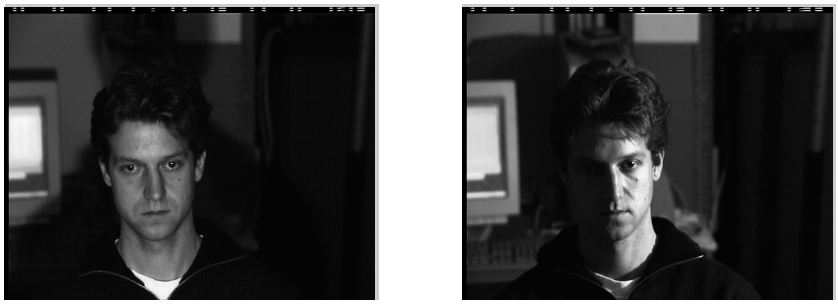
Note that $\theta_1 \geq \theta_2 \geq \ldots \geq \theta_m$, because the subspaces contained in $[\![Y]\!]$ are nested.

Now suppose $[\![X]\!] \in \mathrm{FL}(n; 1, \ldots, q)$ and $[\![Y]\!] \in \mathrm{FL}(n; 1, \ldots, p)$ with $q \leq p$. For $j = 1, \ldots, q$, let θ_j be the smallest principal angle between the $[X]^{(j)}$ and $[Y]^{(j)}$. The **first-angle similarity score between full flags** is defined as

$$\tilde{d}([\![X]\!], [\![Y]\!]) = \sqrt{\sum_{j=1}^{q} \theta_j^2}, \tag{27}$$

and again the sequence of θ_j's is non-increasing. These two similarity measures are not equivalent in the case where $[\![X]\!] = [X]$, but they will both prove to have interesting discriminatory properties. In this thesis, the similarity measure being used will only be denoted by the notation on the input data. That is $\tilde{d}([X], [\![Y]\!])$ or $\tilde{d}([\![X]\!], [\![Y]\!])$.

4.3.1. PIE DATA. To take advantage of the structure of the flag mean, this section will consider a data set that contains multiple meaningful forms of variation; the Carnegie Mellon University, 'Pose, Illumination, and Expression' database, or PIE for short [21]. This database consists of color images of the head and shoulders of 68 different subjects. The subjects were photographed under 42 different lighting conditions (illumination), with cameras in 13 different locations about their head (pose). Each set of photos was then duplicated for 4 different facial expressions. With this small amount of expressions, there is not really enough to do interesting classification so they were excluded from experiments. Thus all expressions are 'neutral' for this thesis. The native resolution of each image is



(A) Pose 04, Illumination 10

(B) Pose 04, Illumination 21

(C) Pose 11, Illumination 10

(D) Pose 11, Illumination 21

FIGURE 4.6. Uncropped images of Subject 07 from the PIE data set.

486×640 pixels. The images were converted to gray-scale for ease of computation. The same techniques that are employed in this section could be used on color images, but it would takes three times as long. Samples of two different illumination conditions and poses for a single subject are shown in Figure 4.6.

4.3.2. PRE-PROCESSING. The PIE data set is well-known, and considered 'cracked' in the Computer Vision community. That is, state-of-the-art methods for classification can accurately classify the samples with perfect accuracy. For example, Beveridge *et al.* are able to perfectly identify all subjects in the PIE dataset using the minimum principal angle between illumination spaces [4]. This thesis will not claim to be the state-of-the-art in classifying images from PIE. That is not the intention of these applications. Instead, we aim

to show that even naive classification with the flag mean is good, and that there is room to use these techniques in areas that other algorithms cannot operate.

To improve error rates, this thesis, like most techniques in the literature, performs some pre-processing on the PIE images. The images will all be converted to gray-scale. Additionally they will be cropped and registered by hand so that the eyes of every image are at the same height. To perform this registration without cutting all images down to *just* the eyes, 12 subjects were removed from the data set leaving 56 unique subjects. The removed subjects typically had the front of their face obscured by the left edge of the image in some samples. The cropped images are 277 pixels by 299 pixels, and when vectorized they become vectors in 82823-dimensional space. Additionally, the 42 illumination conditions were broken up into two sets of 21. In one set, the ambient lighting is turn on and the 21 conditions come from a variety of flashes. In the other set, the ambient lighting is turned off, and the flashes are the same. For these experiments, only the samples with the ambient lighting turned off were used, giving us 21 illumination conditions. The other 21 were discarded because they included some subjects wearing glasses. This is not an insurmountable obstacle, but does present some issues that are beyond the scope of this thesis. Samples of the cropped and registered images that are used for classification can be seen in Figure 4.7.

Once the images have been trimmed and aligned properly, they need to be used to create subspaces in order for the flag mean to be applicable. Each image is vectorized as described in Section 2.2. Some number of similar image vectors are then concatenated into a matrix, and an orthonormal basis for the span of the columns of that matrix is found. This orthonormal matrix then becomes the representative for the Grassmannian point of that set of images.

(A) Pose 04,
Illumination 10

(B) Pose 04,
Illumination 21

(C) Pose 11,
Illumination 10

(D) Pose 11,
Illumination 21

FIGURE 4.7. Cropped and registered images of Subject 07 from the PIE data set.

Since there are three meaningful forms of variation in our modified PIE database, there remains a choice about how to group the image vectors.

4.3.3. ORGANIZATION MATTERS. One natural way to organize these PIE images into subspaces is to fix two of the variables and allow the third to run free. For example, take the image vectors of one single subject, a single pose, and all possible illumination conditions and create a point on a Grassmannian. This process could be repeated for the same single subject and all illumination conditions, but with a different fixed pose each time. This creates Grassmannian points of one subject's illumination space, each with a different pose. Points with this organization will be referred to as Subject-Pose-Illumination points or SPI-points.

Consider a subset of the images that consists of 10 illumination conditions, 10 poses, and 1 expression for a single subject. Thus the subset contains 100 unique images of the subject.

FIGURE 4.8. One set of images of Subject 07 with a fixed pose and a variety of illumination conditions used to create a subspace. This organization creates an SPI-point.

FIGURE 4.9. A second set of images of Subject 07 with a fixed pose and a variety of illumination conditions used to create an SPI-point..

Find an orthonormal basis for 10 images of the subject that share a pose and each have a distinct illumination condition from the subset. This defines an SPI-point. It is possible to create 10 distinct SPI-points from the subset of 100 images. The images used to create two of these SPI-points for Subject 07 are displayed in Figure 4.8 and Figure 4.9.

From the subset of 100 images, create 10 SPI-points, one for each pose. If all ten images are linearly independent, the SPI-points will be 10-dimensional subspaces of the pixel space,

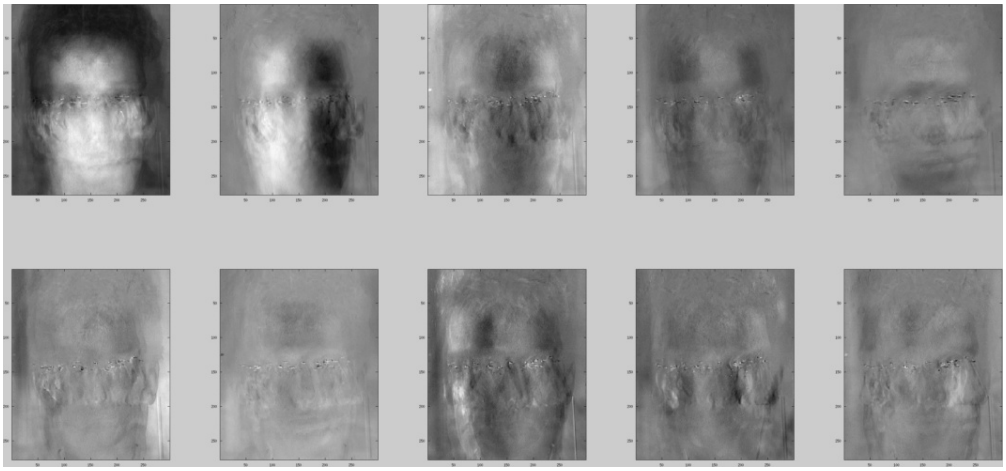


FIGURE 4.10. The first ten images in the flag mean of SPI-points of Subject 07. The points each have a fixed pose and a variety of illumination conditions.

and thus they will live on $\mathrm{Gr}(82823, 10)$. It is difficult to visualize the flag mean in an illustrative way, but there is intuition to be gained from visualizing the vectors, $\{u^{(1)}, \ldots, u^{(r)}\}$, that are used to create it. The first 10 vectors in the flag mean of 10 SPI-points of Subject 07 are displayed in Figure 4.10. With the exception of the first vector, Figure 4.10 makes it appear as though each vector in the flag approximates a single illumination condition of Subject 07, while individual poses are not discernible. This seems reasonable, because the commonality between the SPI-points used to create the flag mean of Subject 07 is that they each contain all 10 illumination conditions present in the subset.

Similarly, with the exact same collection of images we could create points organized as Subject-Illumination-Pose points or SIP-points. These points have a single subject, a single illumination condition, and a range of poses. The SPI-points and the SIP-points would likely not live on the same Grassmann manifold, and there would certainly not be the same number of points of each type. Two of the 10 possible SIP-points of Subject 07 are shown in Figure 4.11 and Figure 4.12.

FIGURE 4.11. One set of images of Subject 07 used to create an SIP-point. The images have a fixed illumination condition and a variety of poses.

FIGURE 4.12. A second set of images of Subject 07 used to create an SIP-point. The images have a fixed illumination condition and a variety of poses.

As before, 10 SIP-points are created from the subset, one for each pose. Since an orthonormal basis for each point must be found to perform computations, the two representations of the same collection of images are quite different, and we hope that the flag mean of the SPI-points and the flag mean of the SIP-points are markedly different as well. The first 10 vectors in the flag mean of the 10 SIP-points of Subject 07 are displayed in Figure 4.13. This time each vector in the flag appears to approximate a single pose of Subject 07.

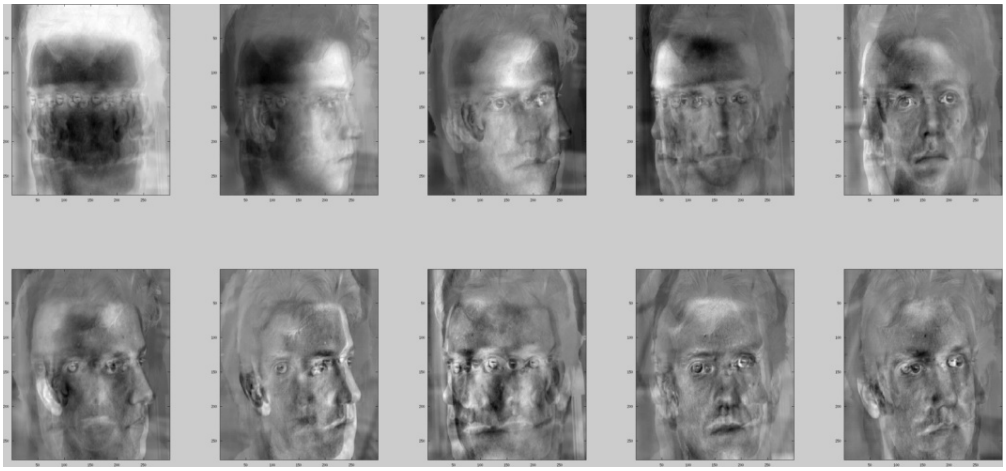


FIGURE 4.13. The first ten images in the flag mean of 10 SIP-points of Subject 07.

Regardless of interpretation, the two flag means are created from the same 100 images, but clearly represent those images in distinct ways.

The flag vectors in Figure 4.10 and Figure 4.13 appear potentially meaningful, but can the two flags be distinguished quantitatively? At a minimum, two flags created out of samples organized randomly should be more similar to each other than a flag created out of SIP-points and a flag created out of SPI-points, if the flags all use the exact same image vectors. Figure 4.14 show the results of 100 trials of this procedure. For each trial, one subject, ten poses, and ten illumination conditions were randomly selected. The images containing all possible combinations of those poses and illuminations were used, creating a set of 100 images. From these 100 images, ten points on $\mathrm{Gr}(10, 82823)$ were created for each flag. The flags were compared using the first-angle similarity score for full flags described in Equation 27. The top, blue line shows the similarity score of the SPI-flag with the SIP-flag of these images, and the bottom, blue line shows the similarity score between the two random flags. The dotted line shows the average distance between the structured flags and the dash-dot line shows the average distance between the random flags. As can be seen in the figure, there

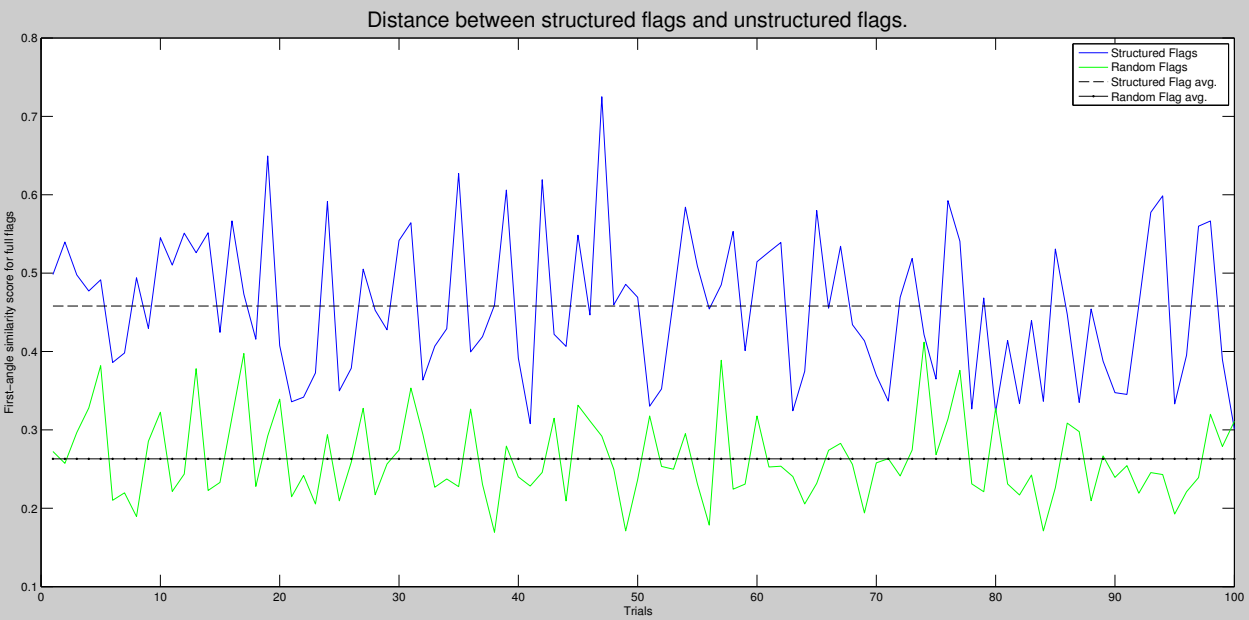


FIGURE 4.14. Distance between an SIP-flag and an SPI-flag compared to the distance between two random flags using the same collection of PIE images, along with their average distances.

were only two trials out of 100 where the distance between the random flags was greater than the distance between the structured flags.

4.3.4. PIE CLASSIFICATION. The results in Figure 4.14 show that organizing the PIE images in meaningful ways has the potential to separate the different forms of variation in the set, or at least separate the true variation from the noise. To push this hypothesis further we attempt supervised classification of the images in the data set based on their subject. Note that these same computations could be performed to classify images based on pose or illumination if they were used as our top level variables. The supervision in this experiment is that we know the labels associated with each sample ahead of time. A portion of the images will be set aside as test samples. The goal is to match each of the test samples to the cluster, or mean, associated with their subject.

The training set consists of the samples corresponding to all combinations of 67% of the poses, and 67% of the illumination conditions. That means there are $9 \times 14 = 126$ images associated with each of the 56 subjects. For each subject we create two flags, a flag made from the 14 possible SPI-points, and a flag created from the 9 possible SIP-points. The remaining 33% of the poses and illumination conditions are held out, leaving $4 \times 7 = 28$ images of each subject where neither the pose or the illumination condition is contained in the training set. There are also $9 \times 7 = 63$ images of each subject where the exact image is not contained in the training set, but the pose has been seen, and $4 \times 14 = 56$ images of each subject where the illumination condition has been seen. Combining these three types of images, the test set has 147 samples of each subject that were not included in the training set. Once the flags have been constructed, each individual test image is compared to all 56 SIP-flags and all 56 SPI-flags using the first-angle similarity score between subspaces and flags. Each image is then given the subject label of the flag that is closest to it. The labels from the classification are then compared to the known labels for each test sample. The confusion matrix for classification based on the SPI-flags can be seen in Figure 4.15 and for the SIP-flags in Figure 4.16. These matrices show the percentage of samples of Subject i, that were correctly classified as Subject i.

The matrices themselves can be quite dense, but a few quantities gleaned from them can provide additional insight. For the SPI-flags, there was an overall error rate of 48.3%. For the SIP-flags, that error rate was 43.6%. If the numbers are broken down by the type of test image used, it appears that images where the pose has already been seen in the training set are easy to classify using the flag means. As shown in Figure 4.17, the error rate for images that overlap pose using SPI-flags is 1.81%. For the SIP-flags, the error rate is 0.00% as in

Figure 4.18. In other words, the SIP-flags perfectly classified the test images where the pose had been seen in the training set.

The high classification rates seen in these test images is not mimicked by the ones where the illumination condition had previously been seen. It makes sense that classification would not improve by using a previously seen lighting condition, because as mentioned in Subsection 2.2.1, the illumination subspaces of an object can be spanned with as little as three images under varying illumination conditions. The error rate for classifying test images of previously seen illumination condtions using SPI-flags is 83.6%, and using SIP-flags it is 76.3%. These confusion matrices can be seen in Figure 4.19 and Figure 4.20. Comparably, the error rate for classifying the test images where neither the pose or illumination condition have been previously seen is 82.0% using SPI-flags and 76.0% using SIP-flags. In all scenarios the error rate is lower for SIP-flags than it is for SPI-flags. The confusion matrices for the test images where neither the pose or illumination have previously been seen are shown in Figure 4.21 and Figure 4.22.

4.4. Analysis

This chapter began with three pieces of motivation for creating a new way to average subspaces. The first was that the Karcher mean and the L_2-median are computed via iterative algorithms. The derivation in Subsection 3.2.1 shows that we can find an analytic solution to the flag mean optimization problem. The exemplar selection experiment in Subsection 4.2.2 and the k-means clustering problem in Subsection 4.2.3, show that the computation time for the flag mean is significantly less than for the Karcher mean. These computation times are even faster than the method for computing the extrinsic mean in some cases, due to the SVD formulation presented in Subsection 3.2.3.

The second factor to be addressed by the flag mean is that the Karcher mean and L_2-median can only provide a unique optimal solution if the points being averaged live close enough together on the Grassmann manifold. Distance between points is not an issue for the flag mean. The rare situation in which the flag mean could find a non-unique optimal solution is if one or more of the eigenvalues of the matrix $\mathbf{A} = \sum_{i=1}^{P} X_i X_i^T$ had a multiplicity greater than 1. This scenario never occurred in the applications contained in this thesis, but if it did there are reasonable ways to create a unique optimum. For example, suppose that the rank$(\mathbf{A}) = 3$, so that $\lambda^{(1)}, \lambda^{(2)}, \lambda^{(3)}$ were the only nonzero eigenvalues, with associated eigenvectors $u^{(1)}, u^{(2)}, u^{(3)}$. Additionally, suppose that $\lambda^{(1)} = \lambda^{(2)} \neq \lambda^{(3)}$. A natural solution would be to use the flag, $[\![\mu_{pF}]\!] = \text{span}\{u^{(1)}, u^{(2)}\} \subset \text{span}\{u^{(1)}, u^{(2)}, u^{(3)}\}$, as the unique optimal solution. The result is still a flag, it just lives on $\text{FL}(n; 2, 3)$ rather than $\text{FL}(n; 1, 2, 3)$.

The third and final motivation for creating the flag mean was to average data that lives on multiple Grassmann manifolds, and be able to compare each data point to the resulting average. This has been achieved by finding the flag mean as the sequential optimizers to Equation 19 and through the similarity scores defined in Section 4.3. The examples shown in Subsection 4.3.3 and Subsection 4.3.4 show that these similarity scores can be applied to real problems for comparing flags to flags and subspaces to flags.

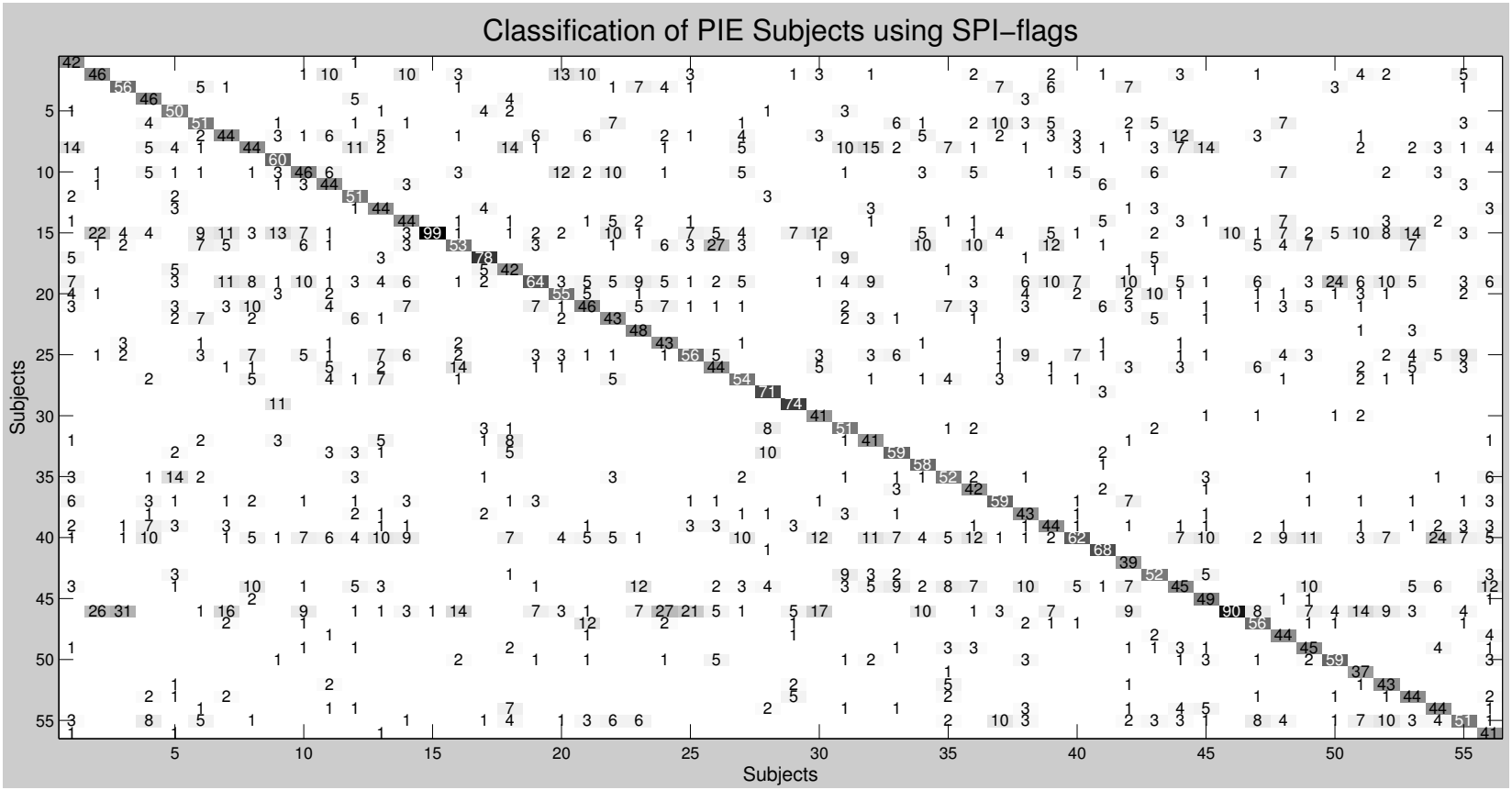


FIGURE 4.15. Confusion matrix for Subject classification using SPI-flags with all test images.

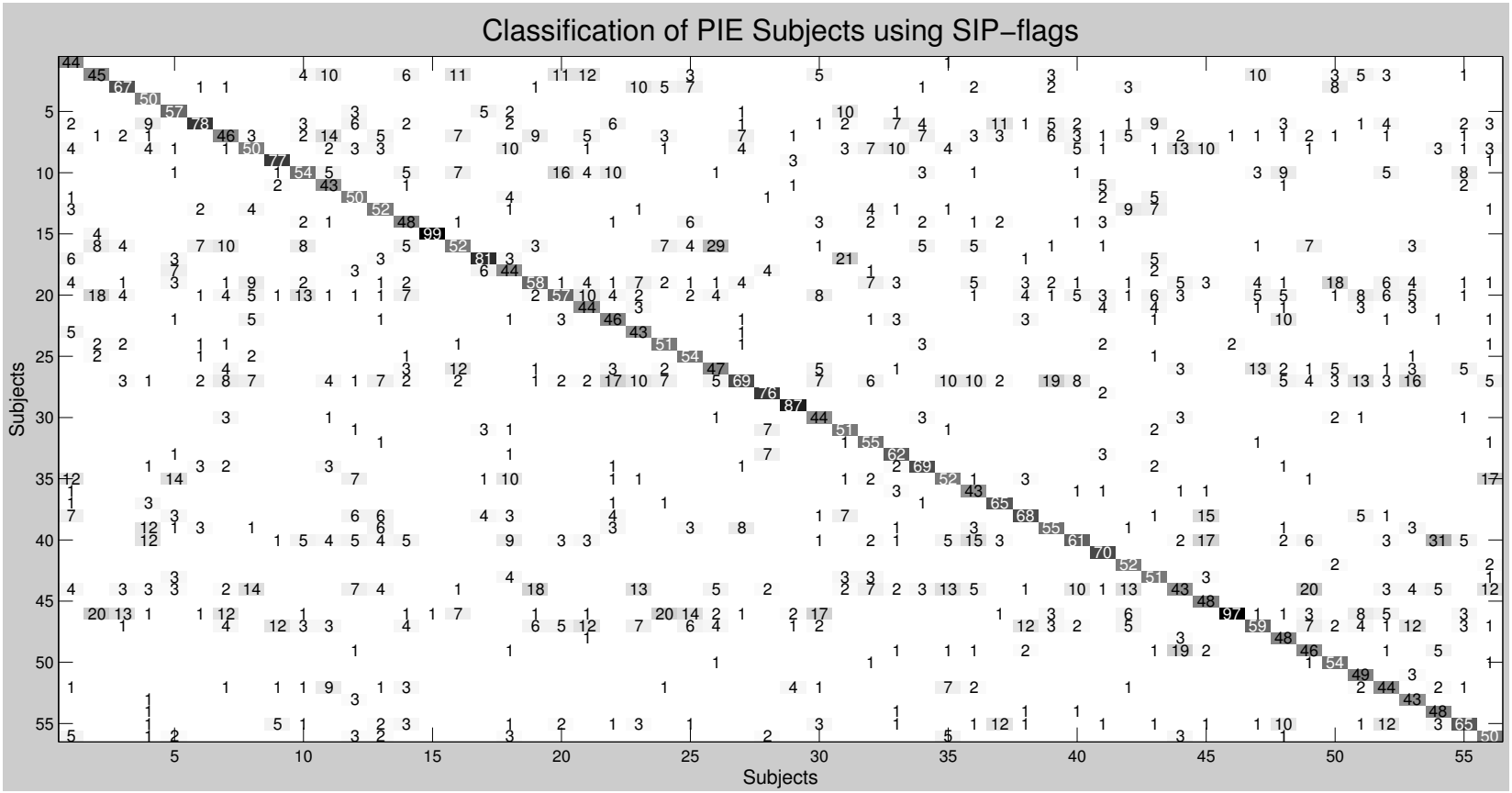


FIGURE 4.16. Confusion matrix for Subject classification using SIP-flags with all test images.

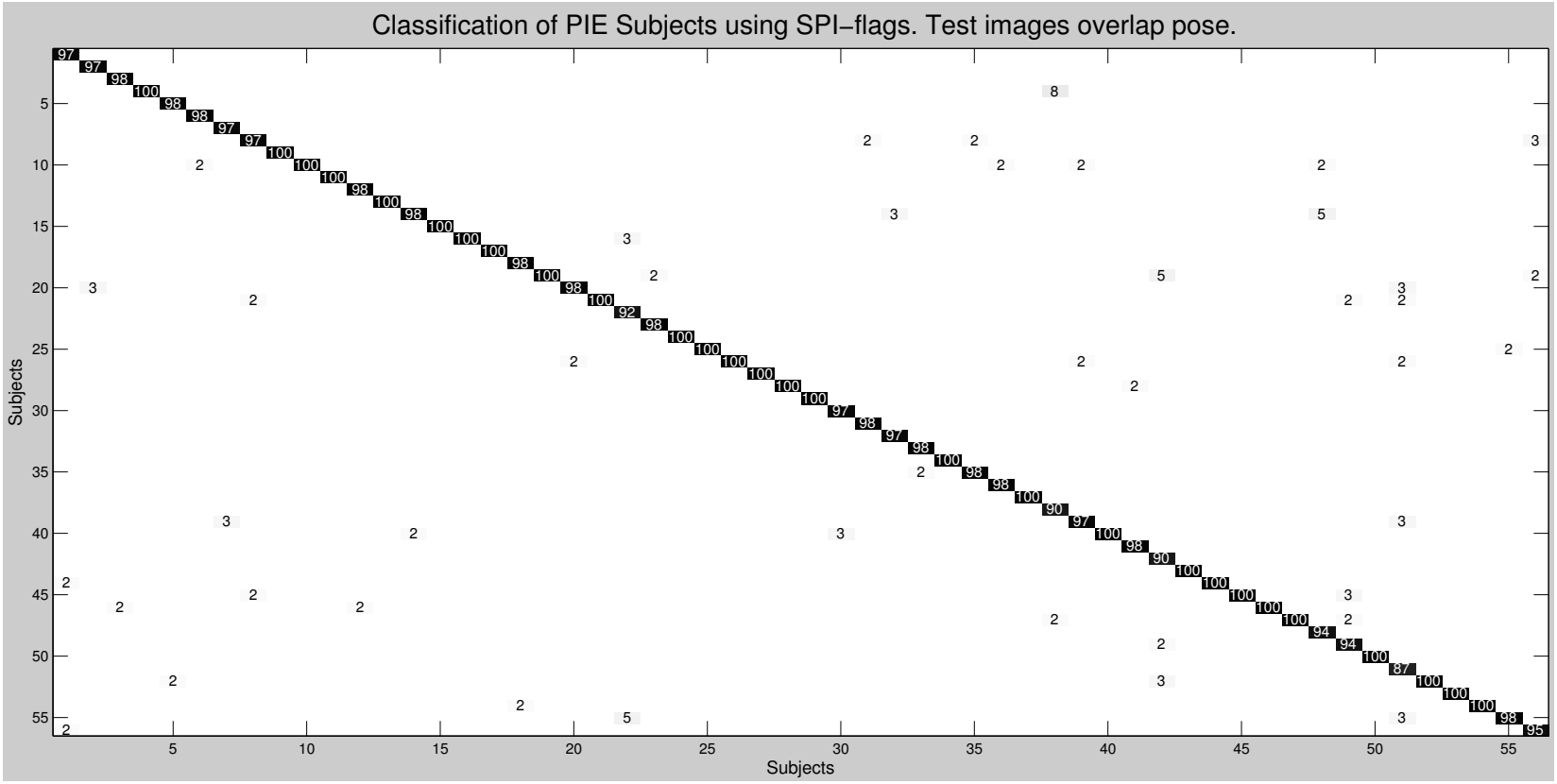


FIGURE 4.17. Confusion matrix for Subject classification using SPI-flags with test images whose poses overlap training samples.

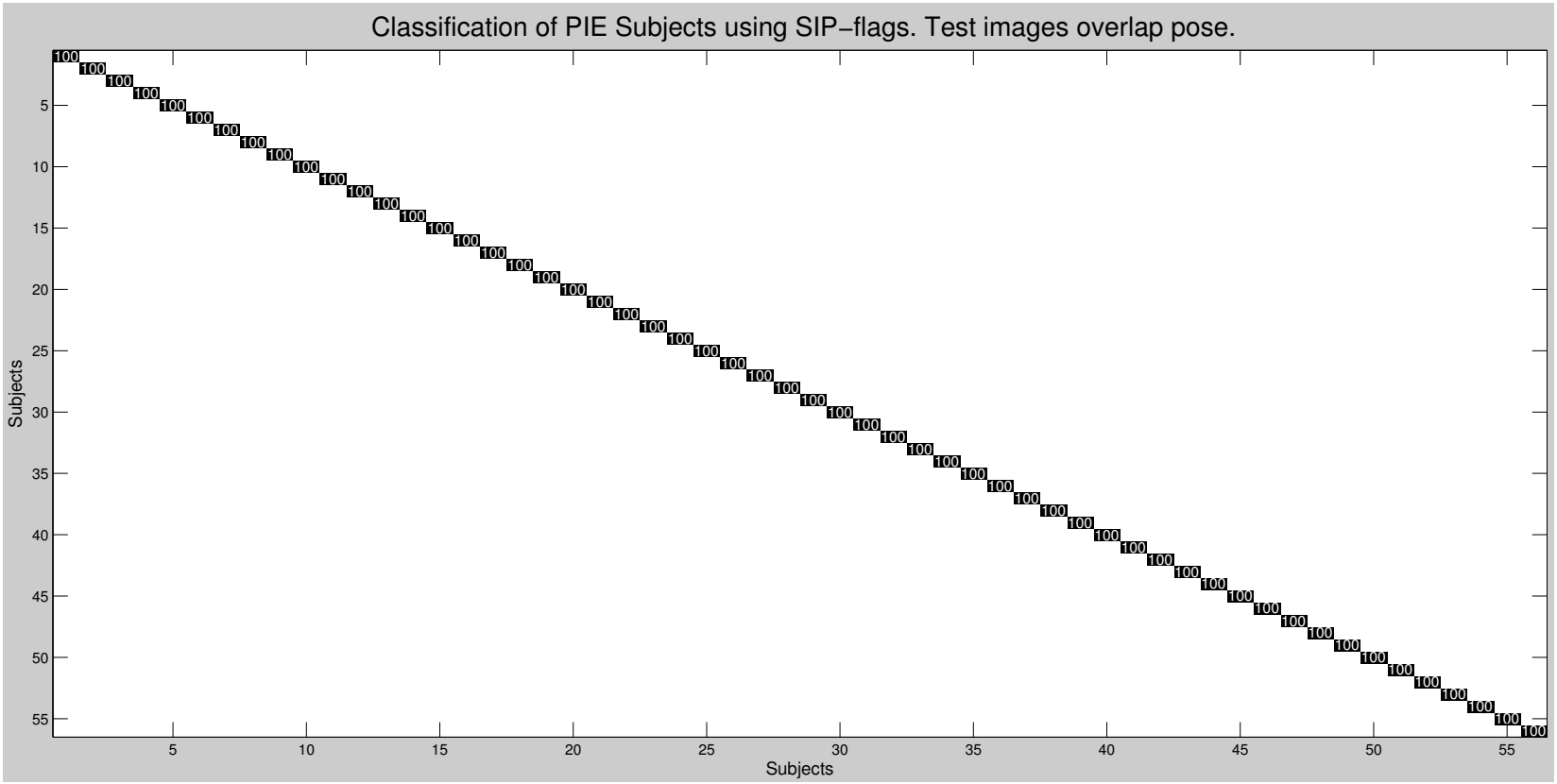


FIGURE 4.18. Confusion matrix for Subject classification using SIP-flags with test images whose poses overlap training samples.

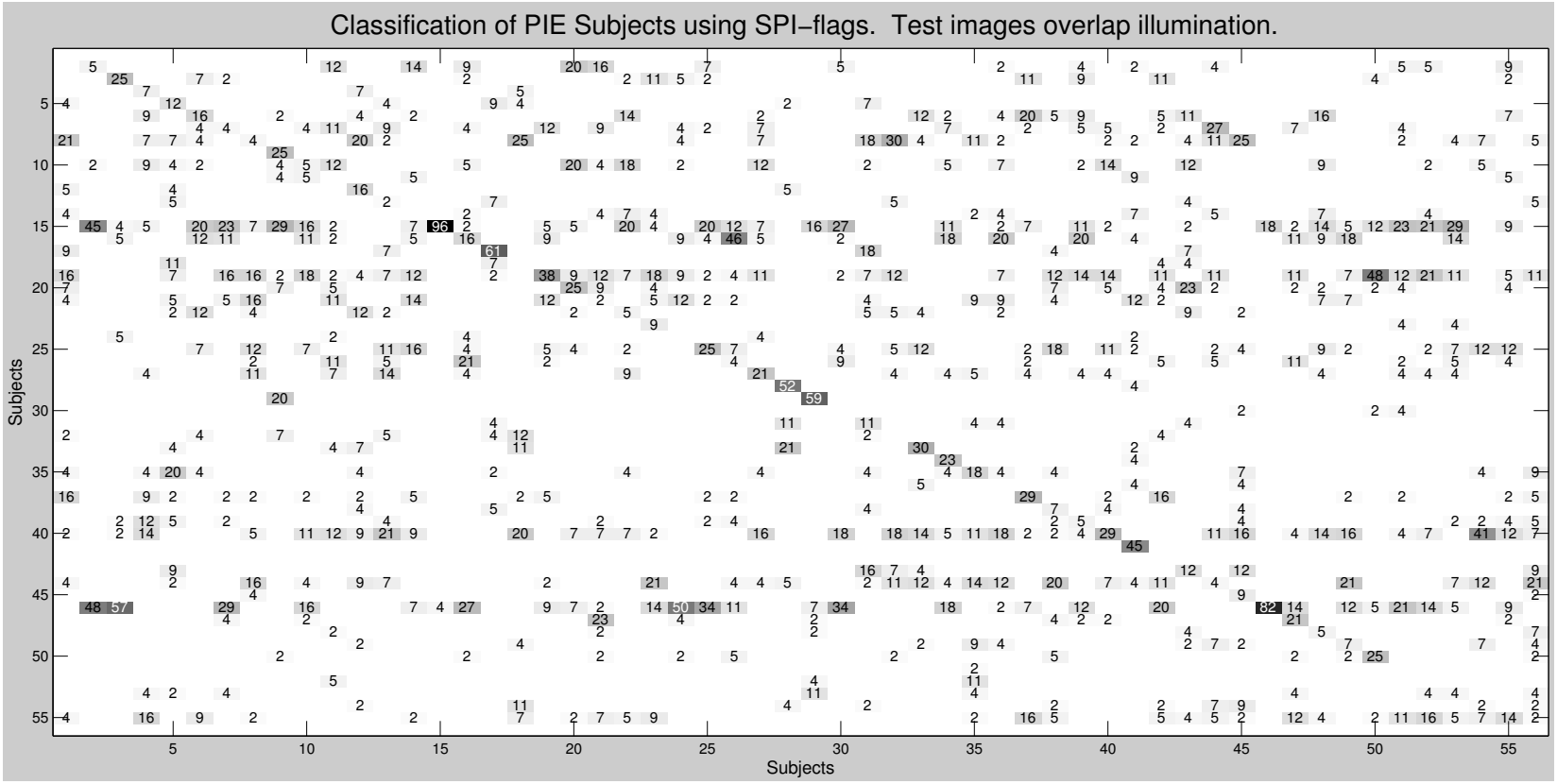


FIGURE 4.19. Confusion matrix for Subject classification using SPI-flags with test images whose illuminations overlap training samples.

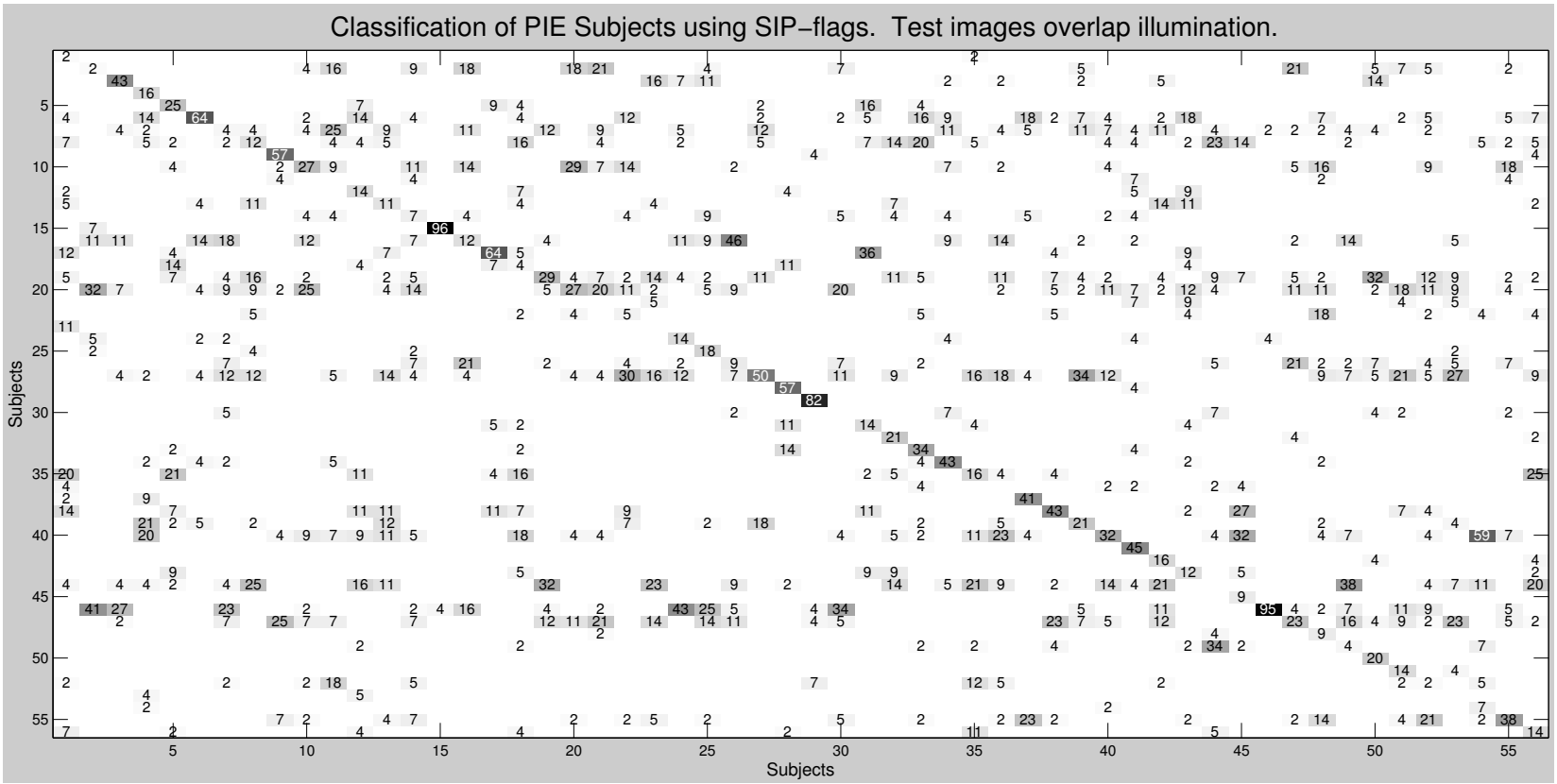


FIGURE 4.20. Confusion matrix for Subject classification using SIP-flags with test images whose illuminations overlap training samples.

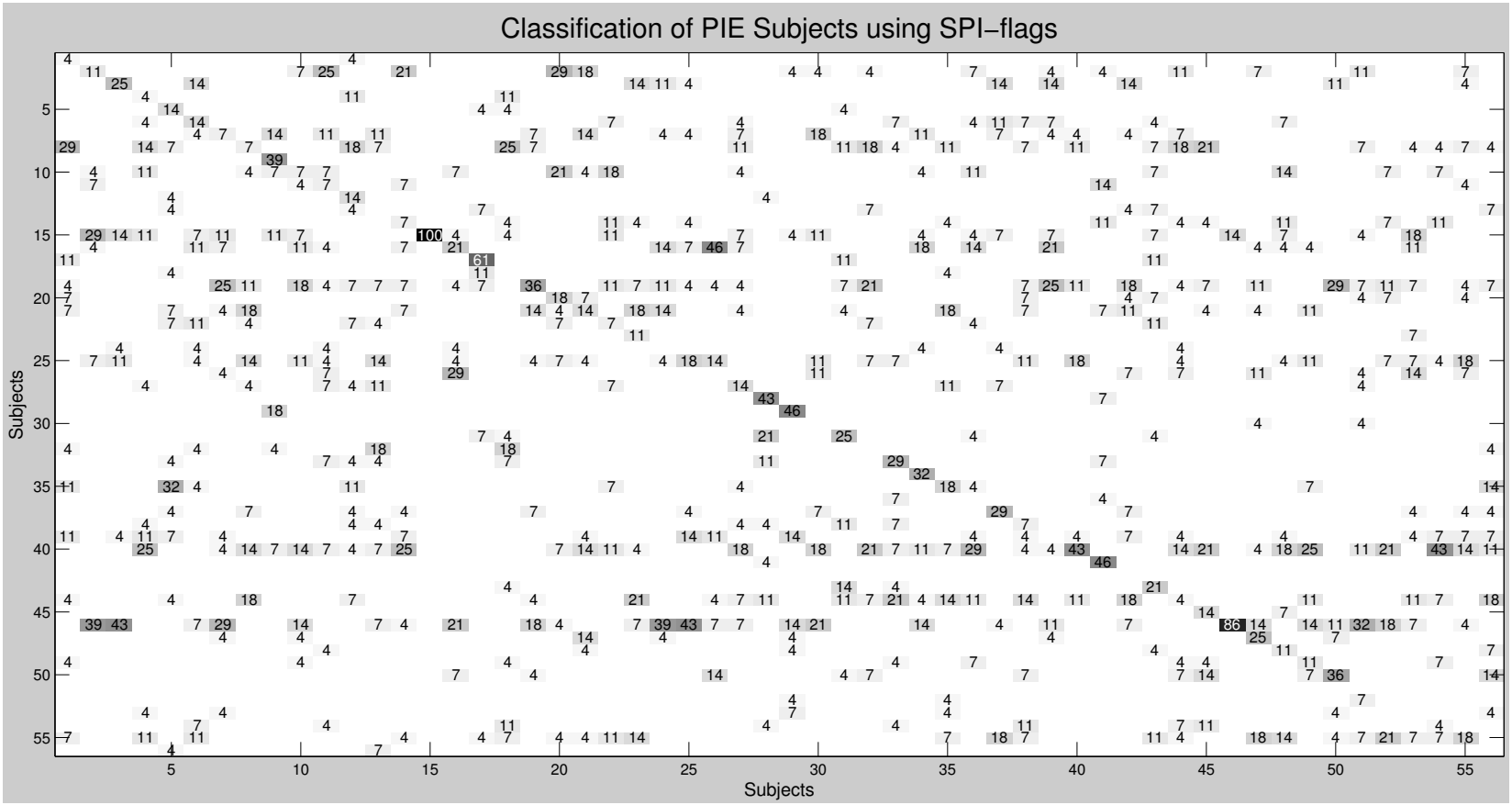


FIGURE 4.21. Confusion matrix for Subject classification using SPI-flags with test images that do not overlap training samples.

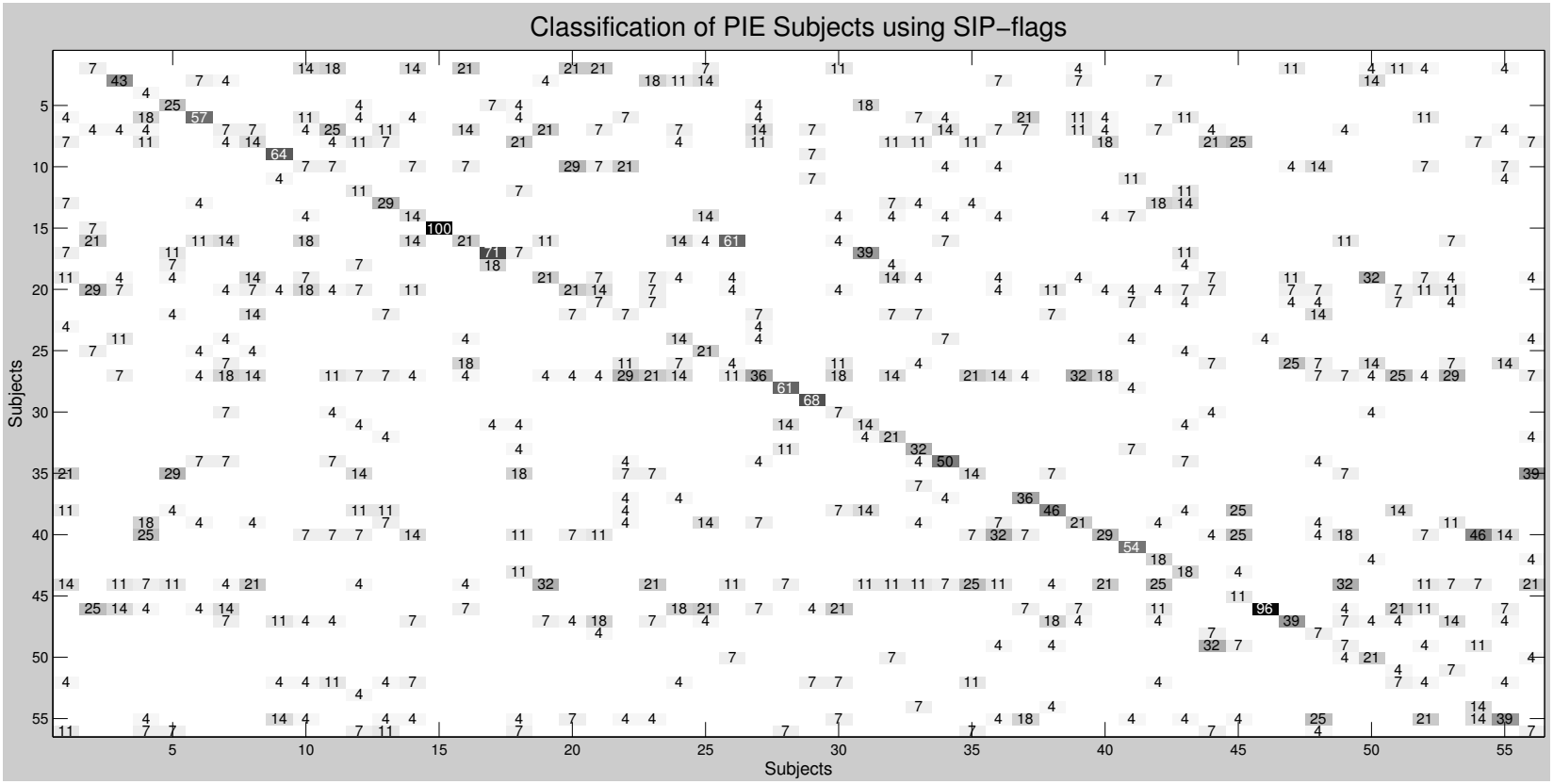


FIGURE 4.22. Confusion matrix for Subject classification using SIP-flags with test images that do not overlap training samples.

CHAPTER 5

Concluding Remarks

This thesis presented a way to represent a collection of subspaces with a flag of best fit, and applied it to examples to demonstrate its practicality. In particular, the thesis used a natural, geometric optimization criterion based on the projection Frobenius norm to average a set of points in the disjoint union of a collection of Grassmann manifolds. It additionally developed some theory with respect to comparing flags to other geometric objects with a goal of classification, that can be used as a starting point for further applications. The flag mean was compared to directly to the Karcher mean and the extrinsic manifold mean, and differences between the flag mean and the L_2-median were discussed.

5.1. Future work

There are many directions that this work could continue in the future. The most immediate extension is to apply the similarity scores used in Section 4.3 to the comparisons with the Karcher mean in Section 4.2. One of the assumed reasons for the poor performance of both the Karcher mean and the flag mean on the Mind's Eye data was that there is a great deal of variability present. The samples were hand labeled to represent the action they contained, but it is very possible that that types of variation that humans key in on are not the only types present. Perhaps if we base classifications on the entire flag created from a collection of video subspaces, and visually inspect them, we will notice what forms of variation dominate the classification.

More generally, we would like to extend this work to attempt to sort and cluster images with multiple forms of variation, like the PIE images, without supervision. The example

in Subsection 4.3.3 showed that using SPI-points and SIP-points resulted flags that were further apart than random ones. It is our goal to sort the PIE images by creating flags out of images selected without supervision that push the flags as far apart as possible.

Other directions for this work will include more rigorous theory around flags and the flag mean. We wish to do direct comparisons with the L_2-median, and hope to show that the flag mean is typically closer to that point than it is to the Karcher mean. Even though we cannot compute the finite sample breakdown point for these estimators on the Grassmann manifold, it would be nice to have intuition about when each type of average is most appropriate.

BIBLIOGRAPHY

[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. 13

[2] E. Begelfor and M. Werman. Affine invariance revisited. *CVPR*, 2:2087 – 2094, 2006. 2, 12, 13

[3] P.N. Belhumeur and D.J. Kriegman. What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision*, 28(3):245–260, 1998. 5

[4] J Ross Beveridge, Bruce A Draper, Jen-Mei Chang, Michael Kirby, Holger Kley, and Chris Peterson. Principal angles separate subject illumination spaces in ydb and cmu-pie. *PAMI*, 31(2):351–363, 2009. 39

[5] A. Björck and G.H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123):579–594, 1973. 8

[6] Yadolah Dodge and Valentin Rousson. Multivariate l1 mean. *Metrika*, 49(2):127–134, 1999. 15

[7] A. Edelman, T.A. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Analysis and Applications*, 20(2):303–353, 1998. 6, 9, 13

[8] Walter Crosby Eells. A mistaken conception of the center of population. *Journal of the American Statistical Association*, 25(169):33–40, 1930. 15

[9] P Thomas Fletcher, Suresh Venkatasubramanian, and Sarang Joshi. The geometric median on riemannian manifolds with application to robust atlas estimation. *NeuroImage*, 45(1 Suppl):S143, 2009. 16, 18

[10] C Gini and L Galvani. Di talune estensioni dei concetti di media ai caratteri qualitativi. *Metron*, 8(1-2):3–209, 1929. 15

[11] JC Gower. Algorithm as 78: The mediancentre. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 23(3):466–470, 1974. 15

[12] JBS Haldane. Note on the median of a multivariate distribution. *Biometrika*, 35(3-4):414–417, 1948. 15

[13] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977. 11, 13

[14] Michael Kirby and Lawrence Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *PAMI*, 12(1):103–108, 1990. 2, 5

[15] J.M. Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer, 1997. 6

[16] Y.M. Lui, J.R. Beveridge, and M. Kirby. Action classification on product manifolds. In *CVPR*, pages 833–839, 2010. 2, 4, 6, 32

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

[18] Hiroshi Murase and Shree K Nayar. Visual learning and recognition of 3-d objects from appearance. *International journal of computer vision*, 14(1):5–24, 1995. 2, 5

[19] Stephen O'Hara and Bruce A Draper. Scalable action recognition with a subspace forest. In *CVPR*, pages 1210–1217. IEEE, 2012. 4

[20] V. Patrangenaru and K.V. Mardia. Affine shape analysis and image analysis. *Proc. 22nd Leeds Ann. Statistics Research Workshop*, July 2003. 2

[21] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In *Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition*, 2002. 38

[22] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *JOSA A*, 4(3):519–524, 1987. 2, 5

[23] Christopher G Small. A survey of multidimensional medians. *International Statistical Review/Revue Internationale de Statistique*, pages 263–277, 1990. 15

[24] A. Srivastava and E. Klassen. Monte Carlo extrinsic estimators of manifold-valued parameters. *IEEE Transactions on Signal Processing*, 50(2):299–308, 2002. 16, 18, 22

[25] A. Srivastava and E. Klassen. Bayesian and geometric subspace tracking. *Advances in Applied Probability*, 36(1):43–56, 2004. 2

[26] 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. 1, 2, 6, 13

[27] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991. 5

[28] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *CVPR*, pages 1–8, 2007. 2

[29] E Weiszfeld and Frank Plastria. On the point for which the sum of the distances to n given points is minimum. *Annals of Operations Research*, 167(1):7–41, 2009. 16