SUBSPACE SEGMENTATION AND HIGH-DIMENSIONAL DATA ANALYSIS

By

Ali Şafak Sekmen

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Mathematics

May, 2012

Nashville, Tennessee

Approved:

Professor Akram Aldroubi

Professor Douglas Hardin

Professor Alexander Powell

Professor Larry Schumaker

Professor Mitchell Wilkes

To my family,

my beloved wife, İlknur

and

my sons, Mert and Emir

ACKNOWLEDGMENTS

I must say that I enjoyed every minute of the whole process of Ph.D. in Mathematics. It is a challenging, yet awarding experience. I was fortunate to work with so many bright mathematicians with different perspectives.

First and foremost, I would like to express my gratitude to my advisor, Akram Aldroubi. I truly appreciate our endless discussions and brainstormings that always encouraged me to do more. He is an extraordinary person who has always challenged me to think mathematically. I learned a lot from him about how a Ph.D. student advisement should be.

I would like to thank the faculty of the Mathematics Department, especially Doug Hardin, Alexander Powell, and Larry Schumaker for many helpful and inspiring conversations, not only on mathematics. I would also like to thank Mitch Wilkes and Wei Chen for their support and giving another perspective for this work. I am grateful to Tamara Rogers and Tim Wallace for their feedback.

My thanks also go to my friends in the graduate program, in particular to Jeremy Lecrone, Xuemei Chen, Anneliese Spaeth and Nattapong Bosuwan, with whom I spent a lot of time discussing mathematical problems.

Finally, I would like to thank Dr. Dietmar Bisch, the Head of the Mathematics Department at Vanderbilt University and Dr. S. Keith Hargrove, the Dean of the College of Engineering at Tennessee State University for their support and encouragements.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

This thesis develops mathematical theory and algorithms for clustering high dimensional data that lives in a union of lower dimensional subspaces. The first focus is to develop theory for modeling signals in terms of union of subspaces. The second focus is to develop algorithms for clustering high dimensional data that can be modeled as a union of subspaces. The third focus is to apply the proposed techniques and algorithms in some computer vision problems including motion segmentation.

1.1 Subspace Segmentation Problem

The problem of subspace clustering is to find a nonlinear model of the form $\mathscr{U} = \bigcup_{i\in I} S_i$ where $\{S_i\}_{i\in I}$ is a set of subspaces that is nearest to a set of data $\mathbf{W} = \{w_1, ..., w_N\} \in \mathbb{R}^D$. The model can then be used to classify the data $\mathbf{W}$ into classes called clusters.

In many engineering and mathematics applications, data lives in a union of low dimensional subspaces [1, 2, 3, 4]. For instance, consider a moving affine camera that captures F frames of a scene that contains multiple moving objects. Let p be a point of one of these objects and let $x_i(p), y_i(p)$ be the coordinates of p in frame i. Define the *trajectory vector* of p as the vector $w(p) = (x_1(p), y_1(p), x_2(p), y_2(p), \ldots, x_N(p), y_N(p))^t$ in $\mathbb{R}^{2F}$. It can be shown that the trajectory vectors of all points of an object in a video belong to a vector subspace in $\mathbb{R}^{2F}$ of dimension no larger than four [5, 6]. Thus, trajectory vectors in videos can be modeled by a union $\mathscr{M} = \cup_{i\in I} V_i$ of l subspaces, where l is the number of moving objects. It can also be shown that human facial motion and other non-rigid motions can be approximated by linear subspaces [7, 8]. Another clustering problem that can be modeled as union of subspaces is recognition of faces. Specifically, the set of all two dimensional images of a given face i, obtained under different illuminations and facial

positions, can be modeled as a set of vectors belonging to a low dimensional subspace, S_i, living in a higher dimensional space $\mathbb{R}^D$ [9, 10, 4]. A set of such images from different faces is then a union $\mathscr{U} = \bigcup_{i \in I} S_i$. Similar nonlinear models arise in sampling theory where $\mathbb{R}^D$ is replaced by an infinite dimensional Hilbert space $\mathscr{H}$, e.g., $L^2(\mathbb{R}^D)$ [11, 12, 1, 13].

This area of research has attracted high interest from computer science, engineering, and applied mathematics in recent years. Most of the notable research has been developed very recently and this work complements and extends theory and techniques from subspace clustering and (compressive) sampling theory. Interactions between certain areas of mathematics and computer science (such as non-linear approximation, optimization, probability theory, and algorithms) are required for solving the subspace segmentation problem.

The goal of subspace clustering is to identify all of the subspaces that a set of data $\mathbf{W} = \{w_1, ..., w_N\} \in \mathbb{R}^D$ is drawn from and assign each data point w_i to the subspace it belongs to. The number of subspaces, their dimensions, and a basis for each subspace are to be determined, even in the presence of noise, missing data, and outliers. The subspace clustering or segmentation problem can be simply stated as follows (as more detailed statement is given in Problem 3 of Section 4.1):

Problem 1. *Subspace Segmentation Problem*

Let $\mathscr{U} = \bigcup_{i=1}^{M} S_i$ *where* $\{S_i \subset \mathscr{H}\}_{i=1}^{M}$ *is a set of subspaces of a Hilbert space* $\mathscr{H}$. *Let* $\mathbf{W} = \left\{w_j \in \mathscr{H}\right\}_{j=1}^{N}$ *be a set of data points drawn from* $\mathscr{U}$. *Then,*

1. *determine the number of subspaces M,*
2. *determine the set of dimensions* $\{d_i\}_{i=1}^{M}$,
3. *find an orthonormal basis for each subspace* S_i,
4. *collect the data points belonging to the same subspace into the same cluster.*

Note that often the data may be corrupted by noise, may have outliers or the data may not be complete, e.g., there may be missing data points. In some subspace clustering problems, the number M of subspaces or the dimensions of the subspaces $\{d_i\}_{i=1}^{M}$ are known.

1.2 Main Approaches

A number of approaches have been devised to solve the problem above or some of its special cases. The key approaches can be summarized as follows:

1.2.1 Sparsity Methods

In a general compressive sampling framework [14, 15], a k-sparse signal $x \in \mathbb{R}^D$ (at most k nonzero entries) can be reconstructed by solving the following convex optimization problem.

$$\min ||\widetilde{x}||_1 \quad \text{subject to} \quad y = A\widetilde{x} \tag{1.1}$$

where $y \in \mathbb{R}^N$ (with $N < D$), and A a measurement matrix satisfying the so called *restricted isometry property* (RIP) [16].

Using ideas similar to the ones in compressed sensing, Eldar [17] recently considered the recovery of signals that lie in a structured union of subspaces instead of a single subspace. Specifically, a signal x is assumed to lie in a union of k disjoint subspaces with known bases. Although x is in one of the subspaces, it is not known which *a priori*. She then shows that the problem of reconstructing x can be cast as a sparse recovery problem, in which a sparse vector with particular sparsity pattern is recovered based on minimizing an ℓ_2/ℓ_1 norm from given measurements.

Elhamifar *et al*. extended Eldar's work and developed an algorithm for linear and affine subspace clustering using sparse representation of vectors [18, 19]. They assume that the data points are drawn from a union of independent [18] or disjoint [19] linear or affine subspaces. They further assume that the collection of data points are self-expressive, i.e., any data point $y \in V$ in the collection, where V is a d dimensional subspace, can be expressed as a linear combination of any other d points that are in the collection and in V. This method, combined with a spectral clustering, gives good results for motion segmentation and it is more general than Eldar's work in compressed sensing [17]. However, the proof of the main theorem in [18] is not convincing since the ℓ_1 norm could be replaced by the ℓ_2 norm without a change in the proof. Also, as stated in [20, 21], this method

is more suitable to model union of bouquets (a concentrated subregion of a subspace) rather than the subspaces.

Liuo *et al*. [22, 20] developed a method that finds the lowest rank representation of the data matrix. The lowest rank representation is used to define the similarity of an undirected graph, which is then followed by spectral clustering. Favaro *et al.* in [23] extends [18, 19, 22, 20].

1.2.2 Algebraic Methods

Generalized Principle Component Analysis (GPCA) is the main algebraic approach for subspace clustering [4, 24, 25]. It models $\mathscr{U} = \bigcup_{i \in I} S_i$ with a set of polynomials whose derivatives at a point are used to determine a set of basis vectors for the subspace passing through that point. The basis vectors are then used for segmenting the data. GPCA can distinguish subspaces of different dimensions. Since it is algebraic, it is computationally inexpensive, however, its complexity increases exponentially as the number of subspaces and their dimensions increase. It is also very sensitive to noise and outliers. Rao *et al*. [26] developed an algebraic method (called Robust Algebraic Segmentation) to partition image correspondences to the motions in a 3-D dynamic scene (that contains 3-D rigid body and 2-D planar structures) under perspective camera projection.

1.2.3 Iterative and Statistical Methods

Iterative methods such as nonlinear least squares [12, 3] and K-subspaces [27] start with an initial estimation of subspaces (or estimation of the bases of the subspaces). Then, a cost function reflecting the "distance" of a point to a each subspace is computed and the point is assigned to its closest subspace. After that, each cluster of data is used to reestimate each subspace. The procedure is repeated until the segmentation of data points does not change.

The statistical methods such as Multi Stage Learning (MSL) [2, 28] are typically based on Expectation Maximization (EM) [29]. The union of subspaces is modeled by a mixture of probability distributions. For example, each subspace is modeled by a Gaussian distribution. The model parameters are then estimated using *Maximum Likelihood Estimation*. This is done by using a

two-step process that optimizes the *log-likelihood* of the model which depends on some hidden (latent) variables. In *E-Step* (Expectation), the expectation of the *log-likelihood* is computed using the current estimate of the latent variables. In *M-Step* (Maximization), the values of the latent variables are updated by maximizing the expectation of the *log-likelihood*.

The success of the iterative and statistical methods highly depends on initialization of model parameters or segmentation. They generally assume that the number of subspaces as well as their dimensions are known, and they are robust to noise and outliers. RANdom SAmple Consensus (RANSAC) [30], which has been applied to numerous computer vision problems, is successful in dealing with noise and outliers, however, it assumes that the dimension of each subspace is known and each subspace has the same dimension. RANSAC uses certain number of samples to fit a model to the samples. Then, it applies a threshold to the residual of each point in the data set to the model to determine inliers and outliers. The process is repeated until the number of inliers are acceptable. In [31], a Grassmannian minimization approach is used for segmenting linear subspaces by determining the subspace with the maximum number of inliers (maximum consensus subspace).

1.2.4 Spectral Clustering and Other Methods

A detailed treatment of spectral clustering is given in Section 2.3. Spectral clustering [32] is often used in conjunction with other methods as the final step in clustering. Some of the latest subspace clustering algorithms (such as [18, 19, 33]) aim at defining an appropriate similarity matrix between data points which then can be used for further processing using the spectral clustering method (see Algorithm 1 below). An application of spectral clustering to motion segmentation can be found in [34]. [35] provides a spectral clustering algorithm that aims at reducing the computational complexity. The motion segmentation algorithm (Local Subspace Affinity - LSA) developed by Yan and Pollefeys [36] first estimates a local linear manifold for each trajectory data and then computes an affinity matrix based on the principle subspace angles between each pair of local linear manifolds. The algorithm then uses spectral clustering for segmenting the trajectories of

independent, articulated, rigid, and non-rigid body motions. Spectral curvature clustering (SCC) [37, 38] is a variant of LSA. SCC uses polar curvature as the similarity measure instead of principle angles (between the estimated local subspaces). It also estimates the local subspaces using iterative random sampling. [39] uses local linear embedding for clustering and [40] extends [39] for clustering data lying in different submanifolds of a Riemannian space. [41] gives a detailed treatment of various related algorithms.

Algorithm 1 Spectral Clustering Algorithm

Require: Assume $\{x_i\}_{i=1}^{N}$ are data points in $\mathbb{R}^D$. Subspace clustering algorithms based on spectral clustering generally gets the followings as input:

- A similarity (affinity) matrix $S = (s_{ij})$, where s_{ij} determines how "close" x_i is to x_j. For example, $s_{ij} = exp(-||x_i - x_j||_2^2/2\sigma^2)$ if $i \neq j$ and $s_{ii} = 1$.
- The number of subspaces, m.

1: Compute the diagonal degree matrix $D = Diag(d_1, \cdots, d_N)$ where $d_i = \sum_{j=1}^{N} s_{ij}$.
2: Compute the normalized graph Laplacian matrix $L = D^{-1/2} S D^{-1/2}$. L is positive semi-definite with the smallest eigenvalue 0.
3: Compute the m eigenvectors $u_1, \cdots, u_m$ corresponding to m highest eigenvalues.
4: Build the matrix $W = [u_1\ u_2\ \cdots\ u_m] \in \mathbb{R}^{N \times m}$.
5: Apply a traditional clustering technique (such as k-means) in $\mathbb{R}^m$ to the rows of W.

The reduction methods initially build a data matrix that contains the data points as columns. They perform dimensionality reduction and noise elimination. This is typically done by factoring the data matrix by Singular Value Decomposition (SVD). These methods then generate an interaction (similarity) matrix to be used to cluster the data points in the original data matrix. Some related work can be found in [42, 43, 44, 45, 36]. The main drawback of these methods is their sensitivity to noise and outliers. Although some of the noise is reduced by SVD, it becomes difficult to eliminate the remaining noise. Yan and Pollefeys [36] developed an algorithm that uses local affinity of the data points to cluster them (the subspace angles are explained in Section 2.1). Each data point is first projected on a unit sphere and some neighboring points are found by calculating the angle between the points. Then, a local subspace is fitted to the neighboring points. In other words, each point is represented by a local subspace. The distance between the local subspaces are calculated

and a similarity matrix is computed. Finally, spectral clustering is applied to the similarity matrix. This method fails to cluster the data points around the intersection of the subspaces and it is not guaranteed that the algorithm will work even with perfect data matrix (noise and outliers free with no missing data points).

1.3 Thesis Overview

1.3.1 Thesis Goal

There is a growing interest in computer science, engineering, and mathematics for modeling signals in terms of union of subspaces and manifolds. Subspace segmentation and clustering of high dimensional data drawn from a union of subspaces are especially important with many practical applications in computer vision, image and signal processing, communications, and information theory. *The research goal of this thesis is to develop mathematical theory and algorithms for modeling and clustering high dimensional data that lives in a union of lower dimensional subspaces.* This work conducts balanced research that brings the theoretical foundation of subspace segmentation and high dimensional data clustering together with practical applications in computer science and engineering.

1.3.2 Associated Objectives

The associated objectives are three-fold. The first objective is to develop mathematical theory for modeling signals in terms of union of subspaces. The second objective is to develop mathematical algorithms for clustering high dimensional data that can be modeled as a union of subspaces. The third objective is to apply the proposed techniques and algorithms in some computer vision problems including motion segmentation.

Theory for Signal Modeling

For subspace clustering, the data set is assumed to be drawn from a union of subspaces. The data set is typically corrupted by noise, some data points are simply outliers, and some data vectors may have missing components. For example, motion segmentation can be considered as a special case of subspace segmentation. First, a $2F \times N$ data matrix $\mathbf{W}$ is constructed by using N feature points that are tracked across F frames. Then, each column of $\mathbf{W}$ (i.e. the path of a feature point) is treated as a data vector and it is shown that all of the data vectors that correspond to the same moving object lie in at most 4-dimensional subspace of $\mathbb{R}^{2F}$. The number of subspaces corresponds to the number of moving objects. However, the components of the data vectors may be corrupted by noise, some data vectors may be outliers due to unreliable computer vision algorithms, or some of the component of the data vectors may be missing due to occlusion. *The first objective is to establish data (signal) models for such high dimensional data sets. The models will be robust to noise, outliers, and missing data components.*

Algorithms for Subspace Segmentation and Data Clustering

The second objective is to develop novel algorithms for subspace segmentation and high dimensional data clustering. Almost all of the existing algorithms assume that the noise is light-tailed (e.g. Gaussian distributed noise). However, in many practical applications, such as tracking fast moving targets, the noise is heavy-tailed (e.g. Laplacian distributed noise), and the traditional noise reduction techniques (e.g. SVD) are not effective. This work develops algorithms that can handle light any heavy tailed noise.

Applications and Evaluation

The third objective is to apply the theory and algorithms to some important applications in computer vision and image processing. The algorithms are evaluated using multiple means. First, a set of synthetic data is generated for evaluating various cases of subspace segmentation problem. This is especially used extensively in evaluating the reduced row echelon form based subspace segmen-

tation algorithms. Different types of noise are added to the synthetic data to measure robustness for light-tailed and heavy-tailed noise. Second, the algorithms are evaluated using the Hopkins 155 Dataset and compared with some state-of-the-art subspace and motion segmentation algorithms.

1.4 Thesis Contributions

1. This work develops theory and algorithms for solving the general subspace segmentation and data clustering problem described in Section 1.1. We prove that, in the absence of noise, the Reduced Row Echelon Form (RREF)-based algorithm fully determines the subspaces and it clusters the data points. The algorithm is based on the binary reduced row echelon form of a data matrix.

    - An approach based on reduced echelon form was proposed by Gear in [43] for the special case related to motion segmentation. His approach was based on the observation that the reduced echelon form gives a matrix decomposition that can be used for the segmentation. However, as stated by Gear, this fact was based on observation, and he did not provide a mathematical proof. In our case, we solve the general subspace segmentation problem with full mathematical proofs and justifications.
    - We provide a comprehensive theoretical analysis of our algorithm and determine its limitations and strengths in the presence of light-tailed/heavy-tailed noise distributions and outliers.

2. We also present a clustering algorithm for high dimensional data that are drawn from a union of low dimensional subspaces of equal and known dimensions. The algorithm is applicable to the motion segmentation problem and uses some fundamental linear algebra concepts. Some of our ideas are similar to those of Yan and Pollefeys [36]. However, our algorithm differs from theirs fundamentally as described below:

    - Yan and Pollefeys' method estimate a subspace S_i for each point x_i, and then computes the principle angles between those subspaces as an affinity measure. In our work,

we also estimate a subspace for each point, however, these local subspaces are used differently. They are used to compute the distance between each point x_j to the local subspace S_i for the data point x_i.

- In their method, an exponential function for affinity of two points x_i and x_j is used, and this exponential function depends on the principle angles between the subspaces S_i and S_j that are associated with x_i and x_j, respectively. In our case, the affinity measure is different. We first find the distance between x_j and S_i and then apply a threshold, computed from the data, to obtain a binary similarity matrix for all data points.
- The method of Yan and Pollefeys uses spectral clustering on the normalized graph Laplacian matrix of the similarity matrix they propose. However, our approach does not use the spectral clustering on the normalized graph Laplacian of our similarity matrix. Instead, our constructed binary similarity matrix converts our original data clustering problem to a simpler clustering of data from 1-dimensional subspaces which can be solved by any traditional data clustering algorithm.

3. Our algorithm is reliable in the presence of noise and applied to the Hopkins 155 Dataset it generates the best results to date for motion segmentation. The two motion, three motion, and overall recognition rates for the video sequences are 99.43%, 98.69%, and 99.24%, respectively.

4. Many of the subspace segmentation algorithms use SVD to represent the data matrix $\mathbf{W}$ as $\mathbf{W} = U\Sigma V^t$ and then replace $\mathbf{W}$ with the first r rows of V^t, where r is the effective rank of $\mathbf{W}$. This work provides a formal justification for this in Proposition 4.2.1.

1.5 Thesis Organization

Chapter 2 introduces some concepts including angles and distances between subspaces, independent subspaces, and spectral clustering. We heavily use the concepts of subspace principle angles in Chapter 4. A detailed treatment of motion segmentation problem, which is a special case of

general subspace segmentation problem, is given in Chapter 3. Chapter 4 presents theory and associated algorithms for solving general subspace segmentation problem. The limitations of the theory and algorithms are discussed in detail. Some simulations with synthetic and some experiments with real world data are also provided in this chapter. Chapter 5 focuses on a devised method that is suitable for subspaces of equal and known dimensions.

CHAPTER 2

PRELIMINARIES

For the purpose of subspace segmentation, it is important to define a measure of distance or separation that can describe the relative positions of two subspaces. The principle (or canonical) angles can be used to quantify the separation of two subspaces. In Chapter 4, it is shown how the subspace separation affects the subspace segmentation accuracy in the presence of noise. In this chapter, we first define the principle angles between subspaces and then we provide some related theory pertaining to independent subspaces. Finally, we introduce the spectral clustering technique, which is highly utilized by many of the existing state-of-the-art subspace segmentation methods. Although we do not directly use spectral clustering, our subspace segmentation method in Chapter 5 is implicitly related to some fundamental concepts of spectral clustering and we point out this in the associated discussion of Chapter 5.

2.1 Angles and Distances between Subspaces

In order to separate two subspaces of $\mathbb{R}^D$, we may measure the (principle or canonical) angle between them. However, defining the angle between subspaces may not be easy for the subspaces of $\mathbb{R}^D$ for $D > 3$ compared to the subspaces of $\mathbb{R}^2$ or $\mathbb{R}^3$ due to the difficulty of visualization.

Definition 1. (*Minimal Angle*) Let $\mathscr{F}$ and $\mathscr{G}$ be subspaces of $\mathbb{R}^D$. The minimal angle between $\mathscr{F}$ and $\mathscr{G}$ is defined as

$$\theta_{\min} = \arccos \left[\max_{\substack{f \in \mathscr{F} \\ g \in \mathscr{G} \\ ||f||_2 = ||g||_2 = 1}} f^T g \right] \tag{2.1}$$

Definition 2. $\mathscr{F}$ and $\mathscr{G}$ are complementary subspaces of $\mathbb{R}^D$ if $\mathscr{F} \bigoplus \mathscr{G} = \mathbb{R}^D$.

The minimal angle defined above is useful for complementary subspaces, however, it may not

be very useful for subspaces with nontrivial intersection. If $\mathscr{F}$ and $\mathscr{G}$ have a nontrivial intersection (i.e., $\mathscr{F} \bigcap \mathscr{G} \neq 0$), then $\theta_{\min} = 0$. However, this does not necessarily mean that there is no separation between $\mathscr{F}$ and $\mathscr{G}$. For example, in Figure 2.1 the minimal angle is 0 but separations are different.

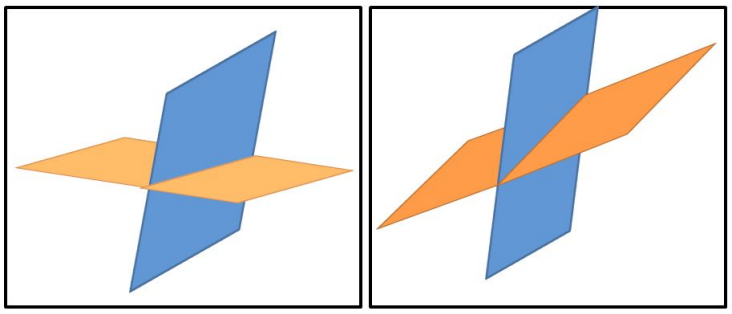

Figure 2.1: Subspace angles.

Therefore we can define another concept to measure the gap between $\mathscr{F}$ and $\mathscr{G}$. This is shown in Figure 2.2.

$$d(\mathscr{F},\mathscr{G}) = \max_{\substack{g\in\mathscr{G} \\ ||g||_2=1}} dist(g,\mathscr{F}) = \max_{\substack{g\in\mathscr{G} \\ ||g||_2=1}} ||(I-P_{\mathscr{F}})g||_2 \tag{2.2}$$

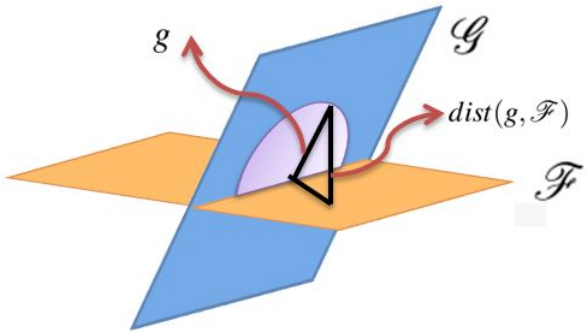

Figure 2.2: Distance between two subspaces.

Note that $d(\mathscr{F},\mathscr{G})$ does not need to be equivalent to $d(\mathscr{G},\mathscr{F})$ and therefore $d(\mathscr{F},\mathscr{G})$ is a directed distance. Also

$$d(\mathscr{F},\mathscr{G}) = \max_{\substack{g\in\mathscr{G} \\ ||g||_2=1}} ||(I-P_{\mathscr{F}})g||_2 \leq \max_{\substack{g\in\mathscr{G} \\ ||g||_2=1}} ||(I-P_{\mathscr{F}})||_2||g||_2 = 1 \tag{2.3}$$

So, $d(\mathscr{F},\mathscr{G}) \leq 1$. Therefore the distance between $\mathscr{F}$ and $\mathscr{G}$ is defined as

$$gap(\mathscr{F},\mathscr{G}) = \min(d(\mathscr{F},\mathscr{G}), d(\mathscr{G},\mathscr{F})) \tag{2.4}$$

Definition 3. (*Maximal Angle*) The maximal angle between $\mathscr{F}$ and $\mathscr{G}$ is defined as

$$\theta_{\max} = \arcsin(gap(\mathscr{F},\mathscr{G})),$$

where $0 \leq \theta_{\max} \leq \pi/2$ [46].

Note that the minimal angle is useful for complementary subspaces and the maximal angle is useful for the subspaces of equal dimension. However, the general subspaces require a more comprehensive definition of separation. For example, consider two subspaces of $\mathbb{R}^D$ that have different dimensions (with a nontrivial intersection). Then, $\theta_{\min} = 0$ and $\theta_{\max} = \pi/2$ and they do not convey too much information. Therefore, we will define some other principle angles that are between $\theta_{\min}$ and $\theta_{\max}$ [46, 47].

Definition 4. Let $\mathscr{F}$ and $\mathscr{G}$ be subspaces of $\mathbb{R}^D$. Let $k = \min(dim\mathscr{F}, dim\mathscr{G})$. Then, the principle angles $\theta_1, \theta_2, \ldots, \theta_k$ are the numbers $0 \leq \theta_i \leq \pi/2$ and they are defined as

$$\cos\theta_i = \max_{\substack{f\in\mathscr{F}_i \\ g\in\mathscr{G}_i \\ f||_2=||g||_2=1}} f^t g = f_i^t g_i \quad i = 1,\ldots,k \tag{2.5}$$

where $\mathscr{F}_1 = \mathscr{F}$ and $\mathscr{G}_1 = \mathscr{G}$, $||f_i||_2 = 1$, $||g_i||_2 = 1$, $\mathscr{F}_i = f_{i-1}^{\perp} \bigcap \mathscr{F}_{i-1}$, and $\mathscr{G}_i = g_{i-1}^{\perp} \bigcap \mathscr{G}_{i-1}$. Note that $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_k$.

The vectors $\{f_i\}_{i=1}^k$ and $\{g_i\}_{i=1}^k$ are called principle vectors. The principle vectors f_1, g_1 and the principle angle θ_1 are first computed. In order to find the second principle angle θ_2, two subspaces orthogonal to f_1 and g_1 respectively are computed. This process continues until all of the principle angles and vectors are determined.

Definition 5. Grassmannian Space $G_r(m,\mathbb{R}^D)$ is the set of all m-dimensional subspaces of $\mathbb{R}^D$ with manifold structure.

Let $\mathscr{F},\mathscr{G} \in G_r(m,\mathbb{R}^D)$. Since $\mathscr{F}$ and $\mathscr{G}$ are curved, the shortest distance between $\mathscr{F}$ and $\mathscr{G}$ is geodesic. It is shown in [48] that

$$d(\mathscr{F},\mathscr{G}) = ||\Theta||_2 = \sqrt{\sum_{i=1}^{k} \theta_i^2} \tag{2.6}$$

Since this distance is not differentiable everywhere, another measure of geodesic distance (called chordal distance) is defined in [49] as

$$d(\mathscr{F},\mathscr{G}) = ||\sin\Theta||_2 = \sqrt{\sum_{i=1}^{k} (\sin\theta_i)^2} \tag{2.7}$$

2.2 Independent Subspaces

Definition 6. Subspaces $\{S_i \subset \mathbb{R}^D\}_{i=1}^n$ are called *independent* if their dimensions satisfy the following relationship:

$$dim(S_1 + \cdots + S_n) = dim(S_1) + \cdots + dim(S_n) \leq D.$$

Lemma 2.2.1. *Let* $\mathbf{W} = \{S_i\}_{i=1}^n$ *be a set of independent subspaces. Then, any non-empty subset of* $\mathbf{W}$ *is a set of independent subspaces.*

Proof. Since $\mathbf{W}$ is a set of independent subspaces, we have

$$dim(\sum_{i=1}^{n} S_i) = \sum_{i=1}^{n} dim(S_i) \tag{2.8}$$

Let $\mathscr{G} = \{S_i\}_{j \in J \subset \{1,\ldots,n\}} \subset \mathbf{W}$.

$$dim(\sum_{i=1}^{n} S_i) \leq dim(\sum_{j \in J \subset \{1,\ldots,n\}} S_j) + dim(\sum_{j \in \{1,\ldots,n\}-J} S_j)$$

Assume $\mathscr{G}$ is not a set of independent subspaces, then $dim(\sum_{j \in J \subset \{1,\ldots,n\}} S_j) < \sum_{j \in J \subset \{1,\ldots,n\}} dim(S_j)$. Therefore,

$$dim(\sum_{i=1}^{n} S_i) < \sum_{j \in J \subset \{1,\ldots,n\}} dim(S_j) + dim(\sum_{j \in J \subset \{1,\ldots,n\}-J} S_j)$$

Since $\sum_{j \in J \subset \{1,\ldots,n\}} dim(S_j) + dim(\sum_{j \in \{1,\ldots,n\}-J} S_j) \leq \sum_{i=1}^{n} dim(S_i)$, we get $\sum_{i=1}^{n} dim(S_i) < \sum_{i=1}^{n} dim(S_i)$. Thus, $\mathscr{G}$ is a set of independent subspaces. □

Lemma 2.2.2. *Let S and V be two linear subspaces. Then, $dim(S \cap V) \geq 1$ if and only if $dim(S + V) < dim(S) + dim(V)$.*

Proof. Obvious. □

Corollary 2.2.3. *The intersection of two independent subspaces is $\{0\}$.*

Proof. Let S and V be two linearly independent subspaces. This implies that $dim(S+V) = dim(S) + dim(V)$. By Lemma 2.2.2, $dim(S \cap V) = 0$. Therefore, $S \cap V = \{0\}$. □

Theorem 2.2.4. *Let $\{S_i\}_{i=1}^{n}$ be subspaces of a vector space V. Then, the following are equivalent.*

1. *Any $\{v_i\}_{i=1}^{n}$ such that $v_i \neq 0$ and $v_i \in S_i$ is a set of linearly independent vectors.*
2. *$\{S_i\}_{i=1}^{n}$ are independent subspaces.*
3. *Let $\theta_{\min}(S_i, \sum_{j=1, j\neq i}^{n} S_j)$ be the smallest principle angle between S_i and $\sum_{j=1, j\neq i}^{n} S_j$. Then, $\min_i(\theta_{\min}(S_i, \sum_{j=1, j\neq i}^{n} S_j)) > 0$.*
4. *$S_i \cap (\sum_{j=1, j\neq i}^{n} S_j) = \{0\}$ for all i.*

Proof. (3) ⇒ (4)

If $\min_i(\theta_{\min}(S_i, \sum_{j=1, j\neq i}^{n} S_j)) > 0$, then $\theta_{\min}(S_i, \sum_{j=1, j\neq i}^{n} S_j) > 0$ for all i. Then, $S_i \cap (\sum_{j=1, j\neq i}^{n} S_j) =$

$\{0\}$, otherwise $\theta_{\min}(S_i, \sum_{j=1,j\neq i}^{n} S_j) = 0$.

(4) $\Rightarrow$ (3)

Obvious.

(1) $\Rightarrow$ (2)

By way of contradiction, assume $\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n = 0$ implies that $\alpha_i = 0$ for all i but the spaces are not independent. Then,

$$dim(S_1 + S_2 + \cdots + S_n) < dim(S_1) + dim(S_2) + \cdots + dim(S_n) \tag{2.9}$$

We claim that there exits $v_1 \in S_i \cap \sum_{j=1,j\neq i}^{n} S_j$ with $v_1 \neq 0$. Otherwise, we have

$$dim(S_i + \sum_{j=1,j\neq i}^{n} S_j) = dim(S_i) + \sum_{j=1,j\neq i}^{n} dim(S_j)$$

(2.9) implies that

$$dim(\sum_{j=1,j\neq i}^{n} S_j) < \sum_{j=1,j\neq i}^{n} dim(S_j)$$

By induction, we obtain a subspace S_k for which $dim(S_k) < dim(S_k)$ for a $k \in \{1,\ldots,n\}$. Thus, $dim(S_i + \sum_{j=1,j\neq i}^{n} S_j) < dim(S_i) + \sum_{j=1,j\neq i}^{n} dim(S_j)$, i.e., $S_i \cap \sum_{j=1,j\neq i}^{n} S_j \neq \{0\}$.

Therefore, $v_1 = v_2 + v_3 + \cdots + v_n$, where $v_j \in S_j$ for $j \neq i$. Since $v_1 \neq 0$, some of v_j for $j \neq i$ are not zero. Thus, $v_1 + (-1) \times \sum_{j,v_j\neq 0} v_j + 0 \times \sum_{j,v_j=0} v_j = 0$, which is a contradiction.

(2) $\Rightarrow$ (4)

Suppose $dim(S_i \cap (\sum_{j\neq i} S_j)) \geq 1$. Then, by Lemma 2.2.2, $dim(S_i + \sum_{i\neq i} S_j) < dim(S_i) + dim(\sum_{j\neq i} S_j)$, which contradicts with the independence assumption.

(4) $\Rightarrow$ (1)

Let $\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n = 0$. Then, $\underbrace{\alpha_i v_i}_{\in S_i} = \underbrace{-\sum_{j \neq i} \alpha_j v_j}_{\in \sum_{j \neq i} S_j}$ for all i. Since, $S_i \cap (\sum_{j \neq i}^{n} S_j) = \{0\}$ for all i, we must have $v_i \neq 0$ for all i. This implies that α_i for all i must be 0. □

2.3 Spectral Clustering

Spectral clustering is a relatively new clustering technique, however, it has become popular due to its simplicity and efficiency. This section briefly gives the motivation behind the algorithm described in Algorithm 1. A more detailed treatment is given in [32].

Spectral clustering algorithms are based on similarity graphs. Consider a weighted undirected graph $G = (V, E)$ with the data points $V = \{x_1, x_2, \ldots, x_n\}$ as vertices. The weight w_{ij} for the edge between x_i and x_j depends on a similarity function s with $s(x_i, x_j) \geq 0$. Then, a *weighted adjacency matrix* W is constructed as

$$W = (w_{ij})_{i,j=1}^{n}$$

It is important that the similarity graph represents or models locality well. Given $V = \{x_i\}_{i=1}^{n}$, we can consider different ways of generating weights between vertices. In the fully connected graph approach, w_{ij} is set to be $s(x_i, x_j)$ for each vertex. In this case, the similarity function should model the local neighborhood relationships well. The Gaussian similarity function $s(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$ is an example of such similarity functions (σ simply controls the degree of the local relationships). In the k-nearest neighborhood approach, x_i is connected to x_j if x_j is among the k-nearest neighbors of x_i using some distance measure. Again the weight w_{ij} is set to be $s(x_i, x_j)$. Even though this approach leads to a directed graph (x_j is in the k-nearest neighborhood of x_i does not imply that x_i is in the k-nearest neighborhood of x_j), the graph can be converted to a undirected graph (for example, x_i and x_j are connected if x_j is in the k-nearest neighborhood of x_i or x_i is in the k-nearest neighborhood of x_j). In the ε-neighborhood approach, two vertices x_i and x_j are connected if the distance (typically inverse of the similarity) is smaller than for some ε. The weights are (typically) set to be uniform (for example $w_{ij} = 1$ for connected vertices and $w_{ij} = 0$

otherwise). Any one of the mentioned graphs can be used for spectral clustering although special care should be given depending on the application. If we choose a symmetric similarity function, then the weighted adjacency matrix W becomes symmetric as well. In the following discussion, we assume W is symmetric.

As the next step of Spectral Clustering, a *graph Laplacian* is computed. Let the degree d_i of a vertex x_i be defined as

$$d_i = \sum_{j=1}^{n} w_{ij}$$

Note that if the graph us not weighted, then d_i is the number of the edges from x_i. We then can define a *degree matrix* as:

$$D = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}.$$

Using D, *the graph Laplacian* matrix is defined as follows:

$$L = D - W = \begin{bmatrix} -w_{11} + \sum_{j=1}^{n} w_{1j} & -w_{12} & \cdots & -w_{1n} \\ -w_{21} & -w_{22} + \sum_{j=1}^{n} w_{2j} & \cdots & -w_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ -w_{n1} & -w_{n2} & \cdots & -w_{nn} + \sum_{j=1}^{n} w_{nj} \end{bmatrix}.$$

Note that L does not depend on the diagonal elements of W, i.e., the self-edges do not affect the graph Laplacian. Clearly, 0 is an eigenvalue of L with $\bar{1}$ (constant one vector) as a corresponding eigenvector. All of the eigenvalues of L are real since L is symmetric. It is also easy to show that 0 is the smallest eigenvalue and therefore L is positive semidefinite. The algebraic multiplicity of the eigenvalue 0 simply determines the number of clusters. For example, let $V_1 = \{x_1, \ldots, x_k\}$, $V_2 = \{x_{k+1}, \ldots, x_m\}$, and $V_3 = \{x_{m+1}, \ldots, x_n\}$ be the vertices from three sets that are disconnected

from each other. In this case,

$$L = \begin{bmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ 0 & 0 & L_3 \end{bmatrix}$$

where L_1, L_2, and L_3 are the graph Laplacians of the three disconnected sets (components). Then, 0 is an eigenvalue with an algebraic multiplicity 3 and the corresponding eigenvectors are $\bar{1}_{V_1} = \begin{pmatrix} \bar{1} \\ 0 \\ 0 \end{pmatrix}$, $\bar{1}_{V_2} = \begin{pmatrix} 0 \\ \bar{1} \\ 0 \end{pmatrix}$, and $\bar{1}_{V_3} = \begin{pmatrix} 0 \\ 0 \\ \bar{1} \end{pmatrix}$, which are the indicator vectors with 1 at entry i if $x_i \in V_i$ for $i = 1,2,3$. If we generalize this, we can conclude that if the graph is decomposed into m connected components V_1, V_2, ..., V_m, then there are m zero eigenvalues and the eigenspace of eigenvalue 0 is spanned by the indicator vectors $\bar{1}_{V_1}$, $\bar{1}_{V_2}$, ..., $\bar{1}_{V_m}$. Although this finding of the eigenspace of eigenvalue seems to solve the clustering problem, computation techniques will not generate eigenvectors in the format of indicators functions. We will end up with an arbitrary eigenbasis which is any combination of the indicators functions $\bar{1}_{V_1}$, $\bar{1}_{V_2}$, ..., $\bar{1}_{V_m}$. Also, in a general clustering setting, we do not require that the components are totally disconnected. Therefore, instead of finding the eigenvectors of 0 eigenvalue, we find eigenvectors corresponding to k smallest eigenvalues. That is, we will use the eigenbasis corresponding to k smallest eigenvalues instead of the eigenbasis of eigenvalue 0. We should also mention that normalized graph Laplacians are also used instead of graph Laplacians [32]. For example,

$$L_{sym} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$
$$L_{rw} := D^{-1} L = I - D^{-1} W$$

where L_{sym} is a symmetric matrix and L_{rw} is a matrix whose entries are closely related to the probabilities assigned to each node in a random walk process.

CHAPTER 3

MOTION SEGMENTATION PROBLEM

The motion segmentation problem can simply be defined as *identifying independently moving rigid objects in a video*. It can be formally stated as follows:

Problem 2. *Motion Segmentation Problem*

Assume that F frames of a scene with k independently moving objects are given. Let $\{p_{i1},\dots,p_{iN}\}_{i=1}^{F}$ *with* $p_{ij}\in\mathbb{R}^3$ *be the N feature points tracked across the F frames. Then,*

1. *determine the number of moving objects, k.*
2. *determine clusters* $\{\mathscr{C}_i\}_{i=1}^{k}$ *of* $\{p_{ij}\}_{i=1,j=1}^{i=F,j=N}$ *so that* $\mathscr{C}_i$ *includes only the feature points that belong to the* i^{th} *moving object.*

We will below show that all of the feature points that belong to the same moving object lie in at most 4-dimensional subspace of $\mathbb{R}^{2F}$. Thus, when Problem 2 is compared with Problem 1 of Section 1.1, it is concluded that the motion segmentation problem is a special case of the general subspace segmentation problem. In Chapters 4 and 5, in order to check accuracy of our techniques, we use a dataset that consists of videos of 2 or 3 moving objects.

3.1 Mathematical Derivation

Assume that there is a rigid body that rotates around a vector in a given coordinate frame as shown in Figure 3.1. We define two coordinate frames: (1) World Frame (X,Y,Z) and (2) Object Frame (x,y,z). Initially, the World Frame and the Object Frame coincide. As the object rotates, the Object Frame deviates from the World Frame. Let $\mathbf{r}_1,\mathbf{r}_2$, and $\mathbf{r}_3$ be the orthogonal unit vectors of the Object Frame (which forms an orthonormal basis for $\mathbb{R}^3$). Let $p\in\mathbb{R}^3$ be a feature point on the object. Let $\mathbf{p}_w=[X_p\ Y_p\ Z_p]^t$ and $\mathbf{p}_o=[a_p\ b_p\ c_p]^t$ be coordinates of p with respect to the World

Frame and the Object Frame, respectively. Since $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$ forms an orthonormal basis for $\mathbb{R}^3$, we have $\mathbf{p}_w = a_p\mathbf{r}_1 + b_p\mathbf{r}_2 + c_p\mathbf{r}_3$. That is, $\mathbf{p}_w = [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3]^t\mathbf{p}_o = R\mathbf{p}_o$, where R is a rotation matrix. If the object both rotates and translates (Figure 3.2), then $\mathbf{p}_w = R\mathbf{p}_o + \mathbf{t}_w$, where $\mathbf{t}_w$ is the world

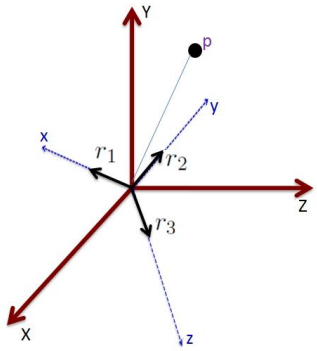

Figure 3.1: Point p on an object that rotates around the World Frame.

coordinates of the center of the object. If the object is sufficiently away from the camera, then the

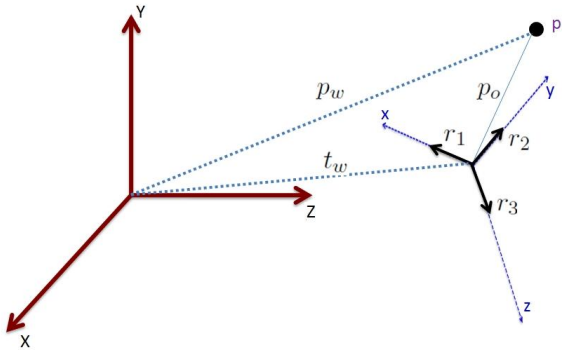

Figure 3.2: Point p on an object that rotates and translates with respect to the World Frame.

camera projection can be modeled as an affine projection [50]. In Figure 3.3, the Z-axis is assumed to be the optical axis of the camera. Therefore, the projection is parallel to the Z-axis. We then can

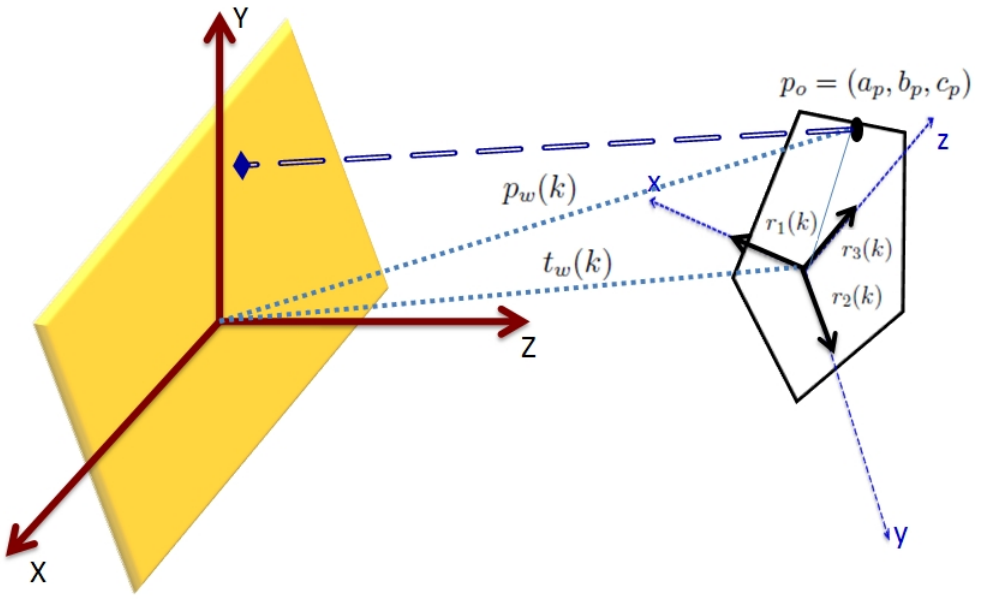

Figure 3.3: Affine camera projection model.

write

$$\mathbf{p}_w(k) = R(k)\mathbf{p}_o + \mathbf{t}_w(k)$$

$$\begin{bmatrix} X_{p(k)} \\ Y_{p(k)} \\ Z_{p(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1(k) & \mathbf{r}_2(k) & \mathbf{r}_3(k) \end{bmatrix} \begin{bmatrix} a_p \\ b_p \\ c_p \end{bmatrix} + \begin{bmatrix} X_{t(k)} \\ Y_{t(k)} \\ Z_{t(k)} \end{bmatrix}$$

$$\begin{bmatrix} X_{p(k)} \\ Y_{p(k)} \end{bmatrix} = a_p\tilde{\mathbf{r}}_1(k) + b_p\tilde{\mathbf{r}}_2(k) + c_p\tilde{\mathbf{r}}_3(k) + \tilde{\mathbf{t}}_w(k) \tag{3.1}$$

where $\tilde{\mathbf{r}}_1(k), \tilde{\mathbf{r}}_2(k), \tilde{\mathbf{r}}_3(k)$, and $\tilde{\mathbf{t}}_w(k)$ correspond to $\mathbf{r}_1(k), \mathbf{r}_2(k), \mathbf{r}_3(k)$, and $\mathbf{t}_w(k)$ with 3^{rd} rows truncated, where k denotes the k^{th} frame.

Figure 3.4 illustrates the projection of a feature point on the camera frame as the object moves. Since the moving body is considered to be rigid, all of the feature points move together with the same translations and rotations. Let $X_{\mathbf{s}(k)}$ and $Y_{\mathbf{s}(k)}$ be coordinates (in World Frame) of the s^{th} feature point in frame k.

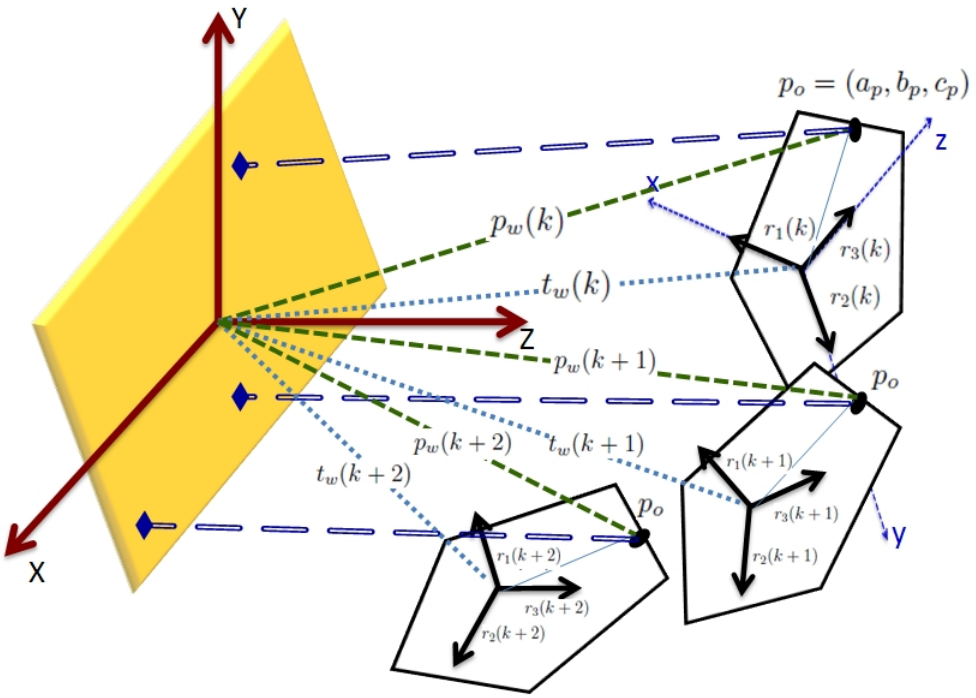

Figure 3.4: Affine projection of path.

We can then define the following data matrix for N feature points collected across F frames:

$$\mathbf{W} = \begin{bmatrix} X_{1(1)} & X_{2(1)} & \cdots & X_{s(1)} & \cdots & X_{N(1)} \\ Y_{1(1)} & Y_{2(1)} & \cdots & Y_{s(1)} & \cdots & Y_{N(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{1(F)} & X_{2(F)} & \cdots & X_{s(F)} & \cdots & X_{N(F)} \\ Y_{1(F)} & Y_{2(F)} & \cdots & Y_{s(F)} & \cdots & Y_{N(F)} \end{bmatrix}_{2F \times N} \tag{3.2}$$

where s^{th} column of $\mathbf{W}$ corresponds to the coordinates of the s^{th} feature point across F frames and $(2i-1)^{th}$ and $(2i)^{th}$ rows correspond to the coordinates of N feature points in i^{th} frame. We get the following by using (3.1):

$$Path_s = \begin{bmatrix} X_{s(1)} \\ Y_{s(1)} \\ \vdots \\ X_{s(F)} \\ Y_{s(F)} \end{bmatrix} = a_s \begin{bmatrix} X_{\tilde{\mathbf{r}}_1(1)} \\ Y_{\tilde{\mathbf{r}}_1(1)} \\ \vdots \\ X_{\tilde{\mathbf{r}}_1(F)} \\ Y_{\tilde{\mathbf{r}}_1(F)} \end{bmatrix} + b_s \begin{bmatrix} X_{\tilde{\mathbf{r}}_2(1)} \\ Y_{\tilde{\mathbf{r}}_2(1)} \\ \vdots \\ X_{\tilde{\mathbf{r}}_2(F)} \\ Y_{\tilde{\mathbf{r}}_2(F)} \end{bmatrix} + c_s \begin{bmatrix} X_{\tilde{\mathbf{r}}_3(1)} \\ Y_{\tilde{\mathbf{r}}_3(1)} \\ \vdots \\ X_{\tilde{\mathbf{r}}_3(F)} \\ Y_{\tilde{\mathbf{r}}_3(F)} \end{bmatrix} + \begin{bmatrix} X_{\tilde{\mathbf{t}}(1)} \\ Y_{\tilde{\mathbf{t}}(1)} \\ \vdots \\ X_{\tilde{\mathbf{t}}(F)} \\ Y_{\tilde{\mathbf{t}}(F)} \end{bmatrix}. \tag{3.3}$$

Therefore, the points $\{Path_1, Path_2, \cdots, Path_N\}$ in $\mathbb{R}^{2F}$ belongs to the 4-dimensional subspace of $\mathbb{R}^{2F}$ spanned by the four vectors on the right hand side of (3.3). In fact, if the motion is translational in the XY plane and rotational in the Z axis, then $\mathbf{r}_3(k) = [0\ 0\ 1]^t$ and $\tilde{\mathbf{r}}_3(k) = [0\ 0]^t$ and therefore the third vector on the right hand side of (3.3) is zero. Hence, $\{Path_1, Path_2, \cdots, Path_N\}$ lies in a 3-dimensional subspace. A similar derivation is described in [51]. Another proof of this result uses "motion and shape matrix factorization" and is given in [45].

3.2 Relation to Subspace Principle Angles

The following lemma (the proof can be found in [47]) describes the relationship between the principle angles of two subspaces and the singular values of a data matrix whose columns are drawn from these two subspaces.

Lemma 3.2.1. *Let $\mathscr{F}$ and $\mathscr{G}$ be two subspaces of $\mathbb{R}^n$ with $p = dim(\mathscr{F}) \leq dim(\mathscr{G}) = q$. Let columns of matrices $Q_{\mathscr{F}} \in \mathbb{R}^{n\times p}$ and $Q_{\mathscr{G}} \in \mathbb{R}^{n\times q}$ form orthonormal bases for the subspaces $\mathscr{F}$ and $\mathscr{G}$. The reduced SVD of $Q_{\mathscr{F}}^t Q_{\mathscr{G}} = Y diag(\sigma_1, \ldots, \sigma_p) Z^t$ with $1 \geq \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, where $Y \in \mathbb{R}^{p\times q}$ and $Z \in \mathbb{R}^{q\times q}$ both have orthonormal columns. Then, the principle angles can be computed as*

$$\theta_k = \arccos(\sigma_k) \quad k = 1, \ldots, p \tag{3.4}$$

where $0 \leq \theta_1 \leq \cdots \leq \theta_p \leq \pi/2$ and the principle vectors are $u_k = Q_{\mathscr{F}} y_k$ and $v_k = Q_{\mathscr{G}} z_k$ for $k = 1, \ldots, p$.

3.2.1 Motion and Subspace Angles

Let us assume that there are only two moving rigid objects ($O_{\mathscr{F}}$ and $O_{\mathscr{G}}$) in the following scenarios.

Case1: The objects do not rotate but translate with the same speed in the same direction.

This case assumes that both of the objects only translate in the same direction with the same speed (with different starting points). In this case, *the subspaces are 3-dimensional and all of three principle angles are zero*. We can see this by checking the basis vectors in (3.3). Let

$$
\begin{aligned}
u_1 &= \begin{bmatrix} X_{\tilde{\mathbf{r}}_1(1)} & Y_{\tilde{\mathbf{r}}_1(1)} & \dots & X_{\tilde{\mathbf{r}}_1(1)} & Y_{\tilde{\mathbf{r}}_1(1)} \end{bmatrix}^t \\
u_2 &= \begin{bmatrix} X_{\tilde{\mathbf{r}}_2(1)} & Y_{\tilde{\mathbf{r}}_2(1)} & \dots & X_{\tilde{\mathbf{r}}_2(1)} & Y_{\tilde{\mathbf{r}}_2(1)} \end{bmatrix}^t \\
u_3 &= \begin{bmatrix} X_{\tilde{\mathbf{r}}_3(1)} & Y_{\tilde{\mathbf{r}}_3(1)} & \dots & X_{\tilde{\mathbf{r}}_3(1)} & Y_{\tilde{\mathbf{r}}_3(1)} \end{bmatrix}^t \\
u_4 &= \begin{bmatrix} X_{\tilde{\mathbf{t}}(1)} & Y_{\tilde{\mathbf{t}}(1)} & \dots & X_{\tilde{\mathbf{t}}(F)} & Y_{\tilde{\mathbf{t}}(F)} \end{bmatrix}^t \\
u_5 &= \begin{bmatrix} X_{\tilde{\mathbf{t}}(1)}+a & Y_{\tilde{\mathbf{t}}(1)}+b & \dots & X_{\tilde{\mathbf{t}}(F)}+a & Y_{\tilde{\mathbf{t}}(F)}+b \end{bmatrix}^t
\end{aligned}
\tag{3.5}
$$

The vectors $B_1 = \{u_1, u_2, u_3, u_4\}$ and $B_2 = \{u_1, u_2, u_3, u_5\}$ are the basis vectors for the trajectories of $O_{\mathscr{F}}$ and $O_{\mathscr{G}}$, respectively. The motion starting point difference for the objects is represented by (a,b). We can apply the Gram-Schmidt process to find a set of orthonormal vectors for each basis. The first two vectors for each basis after the Gram-Schmidt process will be the same since $\{u_1, u_2, u_3\}$ are common for each basis and the rank of $\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}$ is two. Let us call them as v_1 and v_2. Let v_3 and v_4 be the third orthogonal vector for B_1 and B_2, respectively, after the Gram-Schmidt orthogonalization. Note that we can always assume that we also normalize each basis vector in the process. Then,

$$
v_3 = \frac{u_4 - (u_4, v_1)v_1 - (u_4, v_2)v_2}{||u_4 - (u_4, v_1)v_1 - (u_4, v_2)v_2||_2} \tag{3.6}
$$

$$
v_4 = \frac{u_5 - (u_5, v_1)v_1 - (u_5, v_2)v_2}{||u_5 - (u_5, v_1)v_1 - (u_5, v_2)v_2||_2} \tag{3.7}
$$

Simple calculation shows that v_3 and v_4 are identical. Since $Q_{\mathscr{F}} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$ and $Q_{\mathscr{G}} = \begin{bmatrix} v_1 & v_2 & v_4 \end{bmatrix}$ (as in Lemma 4.3.2) and $v_3 = v_4$, we conclude that all of the singular values of $Q_{\mathscr{F}}^t Q_{\mathscr{G}}$ are one and therefore all of the principle angles are zero. *This means that the objects $O_{\mathscr{F}}$ and $O_{\mathscr{G}}$ are indistinguishable and motion cannot be segmented.*

Case 2: The objects have the same rotation.

This case assumes that both of the objects have exactly the same rotation in each frame and they translate freely. In this case, *the subspaces are 4-dimensional and three of the principle angles are zero*. Let

$$
\begin{aligned}
u_1 &= \begin{bmatrix} X_{\tilde{\mathbf{r}}_1(1)} & Y_{\tilde{\mathbf{r}}_1(1)} & \dots & X_{\tilde{\mathbf{r}}_1(F)} & Y_{\tilde{\mathbf{r}}_1(F)} \end{bmatrix}^t \\
u_2 &= \begin{bmatrix} X_{\tilde{\mathbf{r}}_2(1)} & Y_{\tilde{\mathbf{r}}_2(1)} & \dots & X_{\tilde{\mathbf{r}}_2(F)} & Y_{\tilde{\mathbf{r}}_2(F)} \end{bmatrix}^t \\
u_3 &= \begin{bmatrix} X_{\tilde{\mathbf{r}}_3(1)} & Y_{\tilde{\mathbf{r}}_3(1)} & \dots & X_{\tilde{\mathbf{r}}_3(F)} & Y_{\tilde{\mathbf{r}}_3(F)} \end{bmatrix}^t \\
u_4 &= \begin{bmatrix} X_{\tilde{\mathbf{t}}(1)} & Y_{\tilde{\mathbf{t}}(1)} & \dots & X_{\tilde{\mathbf{t}}(F)} & Y_{\tilde{\mathbf{t}}(F)} \end{bmatrix}^t \\
u_5 &= \begin{bmatrix} X_{\hat{\mathbf{t}}(1)} & Y_{\hat{\mathbf{t}}(1)} & \dots & X_{\hat{\mathbf{t}}(F)} & Y_{\hat{\mathbf{t}}(F)} \end{bmatrix}^t
\end{aligned}
\tag{3.8}
$$

Note that $\{u_1, u_2, u_3, u_5\}$ in (3.5) and (3.8) are different. As in Case 1, the vectors $B_1 = \{u_1, u_2, u_3, u_4\}$ and $B_2 = \{u_1, u_2, u_3, u_5\}$ are the basis vectors for the trajectories of $O_{\mathbf{W}}$ and $O_{\mathscr{G}}$, respectively. We can apply the Gram-Schmidt process to find a set of orthonormal vectors for each basis. Since the rank of $\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}$ is three, each subspace (corresponding to each trajectory) is 4-dimensional. We apply the same process described in Case 1, we end up with the same three orthonormal basis vectors after the Gram-Schmidt process for each basis. This means that three of the singular values of $Q_{\mathscr{F}}^t Q_{\mathscr{G}}$ are one, that is, the first three principle angles are zero.

Case 3: The objects have the same rotation about z-axis.

This case assumes that the objects rotate only around z-axis and at the same rate in each frame. They can freely translate. In this case, *the subspaces are 3-dimensional and the first two principle angles are zero*. Note that this is similar to Case 2. The difference is the rotation is always around the z-axis. Therefore, the third basis vector in (3.3) is 0-vector. The same argument of Case 2 applies to this case.

Case 4: The objects have different rotation about z-axis.

This case assumes that the objects rotate only around the z-axis, although not necessarily with the same rate in each frame. The objects can also freely translate. In this case, the third basis vector in (3.3) is 0-vector and the subspaces are 3-dimensional.

CHAPTER 4

SUBSPACE SEGMENTATION

Given a set of data $\mathbf{W} = \{w_1, ..., w_N\} \in \mathbb{R}^D$ that comes from a union of subspaces, subspace segmentation focuses on determining a nonlinear model of the form $\mathscr{U} = \bigcup_{i \in I} S_i$, where $\left\{S_i \subset \mathbb{R}^D\right\}_{i \in I}$ is a set of subspaces, that is nearest to $\mathbf{W}$. The model is then used to classify $\mathbf{W}$ into clusters.

Our approach in this chapter is based on the binary reduced row echelon form of a data matrix. We prove that, in absence of noise, our approach can find the number of subspaces, their dimensions, and an orthonormal basis for each subspace S_i. We provide a comprehensive analysis of our theory and determine its limitations and strengths in the presence of outliers and noise. Chapter 5 will devise another technique for the special case when the subspaces have equal and known dimensions.

4.1 General Subspace Segmentation Problem

The subspace segmentation problem, for both the finite and infinite dimensional space cases, can be formulated as follows:

Let $\mathscr{B}$ be a Banach space, $\mathbf{W} = \{w_1, \ldots, w_m\}$ a finite set of vectors in $\mathscr{B}$. For $i = 1, \ldots, l$, let $\mathscr{C} = C_1 \times C_2 \times \ldots C_l$ be the cartesian product of l families C_i of closed subspaces of $\mathscr{B}$. Thus, an element $\mathbf{S} \in \mathscr{C}$ is a sequence $\{S_1, \ldots, S_l\}$ of l susbpaces of $\mathscr{B}$ with $S_i \in C_i$.

Problem 3. *General Subspace Segmentation Problem*

1. *Given a finite set* $\mathbf{W} \subset \mathscr{B}$*, a fixed* p *with* $0 < p \leq \infty$*, and a fixed integer* $l \geq 1$*, find the infimum of the expression*

$$e(\mathbf{W}, \mathbf{S}) := \sum_{w \in \mathbf{W}} \min_{1 \leq j \leq l} d^p(w, S_j),$$

over $\mathbf{S} = \{S_1, \ldots, S_l\} \in \mathscr{C}$*, and* $d(x,y) := \|x - y\|_{\mathscr{B}}$.

2. *Find a sequence of l-subspaces* $\mathbf{S}^o = \{S_1^o, \ldots, S_l^o\} \in \mathscr{C}$ *(if it exists) such that*

$$e(\mathbf{W}, \mathbf{S}^o) = \inf\{e(\mathbf{W}, \mathbf{S}) : \mathbf{S} \in \mathscr{C}\}. \tag{4.1}$$

Definition 7. For $0 < p \leq \infty$, a set of closed subspaces C of a Banach space $\mathscr{B}$ has the Minimum Subspace Approximation Property p-(MSAP) if for every finite subset $\mathbf{W} \subset \mathscr{B}$ there exists an element $S \in C$ that minimizes the expression $e(\mathbf{W}, S) = \sum_{w \in \mathbf{W}} d^p(w, S)$ over all $S \in C$.

Under the assumption that each family of subspaces C_i satisfies p-(MSAP), problem 3 has a minimizer:

Theorem 4.1.1. *Assume that each* $i = 1, \ldots l$, C_i *satisfies p-(MSAP), then problem 3 has a minimizer.*

Proof. Let $\mathscr{P}(\mathbf{W})$ be the set of all partitions of $\mathbf{W}$ into l subsets, i.e., $P = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\} \in \mathscr{P}(\mathbf{W})$ if $\mathbf{W} = U_i \mathbf{W}_i$, and $\mathbf{W}_i \cap \mathbf{W}_j = \emptyset$. Let $P = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$ be a partition in $\mathscr{P}(\mathbf{W})$. For each subset $\mathbf{W}_i$ in the partition P, find the subspace $S_i^o(P) \in C_i$ that minimizes the expression $e(\mathbf{W}_i, S) = \sum_{w \in \mathbf{W}_i} d^p(w, S)$ over all $S \in C_i$. Let $m = min\{\sum_{i=1}^{l} e(\mathbf{W}_i, S_i^o(P)) : P \in \mathscr{P}(\mathbf{W})\}$, and denote by $P^o = \{\mathbf{W}_1^o, \ldots, \mathbf{W}_l^o\}$ any partition for which $m = \sum_{i=1}^{l} e(\mathbf{W}_i^o, S_i^o(P^o))$. Then, for any $\mathbf{S} = \{S_1, \ldots, S_l\} \in \mathscr{C}$ we have that

$$e(\mathbf{W}, \mathbf{S}) = \sum_{j=1}^{l} e(X_j, S_j) \geq \sum_{j=1}^{l} e(X_j, S_j^o(P_{\mathbf{S}})) \geq \sum_{j=1}^{l} e(\mathbf{W}_j^o, S_j^o(P^o)) = e(\mathbf{W}, \mathbf{S}^o)$$

where $P_{\mathbf{S}} = \{X_1, \ldots, X_l\}$ is any partition of $\mathbf{W}$ generated using $\mathbf{S}$ by

$$X_j = \{w \in \mathbf{W} : d(w, S_j) \leq d(w, S_i),\ i = 1, \ldots, l\}.$$

It follows that $e(\mathbf{W}, \mathbf{S}^o) = m = \inf\{e(\mathbf{W}, \mathbf{S}) : \mathbf{S} \in \mathscr{C}\}$. □

Theorem 4.1.1 suggest a search algorithm for the optimal solution $\mathbf{S}^o$. Obviously, this solution

can be obtained by Algorithm 2. This algorithm will work well if a good initial partition is chosen. Otherwise, the algorithm may terminate in a local optima instead of the global optima.

Algorithm 2 Optimal Solution $\mathbf{S}^o$

1: Pick any partition $P \in \mathscr{P}(\mathbf{W})$
2: For each subset $\mathbf{W}_i$ in the partition P find the subspace $S_i^o(P) \in C_i$ that minimizes the expression $e(\mathbf{W}_i, S) = \sum_{w \in \mathbf{W}_i} d^p(w, S)$
3: **while** $\sum_{i=1}^{l} e(\mathbf{W}_i, S_i^o(P)) > e(\mathbf{W}, \mathbf{S}^o(P))$ **do**
4: **for all** i from 1 to l **do**
5: Update $\mathbf{W}_i = \{w \in \mathbf{W} : d(w, S_i^o(P)) \leq d(w, S_k^o(P)),\ k = 1, \ldots, l\}$
6: Update $S_i^o(P) = \underset{S \in C_i}{\operatorname{argmin}}\ e(\mathbf{W}_i, S)$
7: **end for**
8: Update $P = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$
9: **end while**
10: $\mathbf{S}^o = \{S_1^o(P), \ldots, S_l^o(P)\}$

Remark 4.1.2. In Step-2 of Algorithm 2, we need to determine a subspace $S_i^o(P) \in C_i$ that minimizes the expression $e(\mathbf{W}_i, S) = \sum_{w \in \mathbf{W}_i} d^p(w, S)$. If the data is contaminated with a light-tailed noise distribution (such as Gaussian distributed noise), we set $p = 2$ and minimize $||\mathbf{W}_i - U_i V_i^t||_2$, where the columns of U_i form a basis for $S_i^o(P)$. It is known that SVD (which is an ℓ_2-based approach) can achieve this. However, SVD is not a very effective subspace matching approach if the data is contaminated with a heavy-tailed noise distribution (such as Laplacian distributed noise). In this case, a better way is to estimate a subspace that minimizes $e(\mathbf{W}_i, S) = \sum_{w \in \mathbf{W}_i} d(w, S)$ (p is set to 1). This is the ℓ_1-based approximation of $S_i^o(P)$, in which the minimization $||\mathbf{W}_i - U_i V_i^t||_1$ generally leads to a non-convex optimization problem. However, it can be recast as convex optimization with an iterative reformulation of the problem. The justifications for ℓ_1-based and ℓ_2-based approaches are given in Section 4.4.1 and Section 4.4.2, respectively. The iterative approach of the ℓ_1-based approximation is explained in detail in Section 4.4.2 (and given in Algorithm 5).

4.2 RREF-Based Subspace Segmentation for Noiseless Data

In this section we consider the problem in which a set of vectors $\mathbf{W} = \{w_1, \ldots, w_m\}$ are drawn from a union $\mathscr{U} = \bigcup_{i \in I} S_i$ of l subspaces $S_i \in \mathbb{R}^D$ of dimension d_i. In order to find the l subspaces from the data set $\mathbf{W}$ it is clear that we need $\mathbf{W} = \{w_1, \ldots, w_m\}$ to be of sufficient size. In particular for the problem of subspace segmentation, it is necessary that the set $\mathbf{W}$ can be partitioned into l sets $\mathbf{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$ such that $\operatorname{span} \mathbf{W}_i = S_i$, $i = 1, \ldots, l$. Thus, we need to assume that we have enough data for solving the problem, and that the data is drawn randomly and independently. In particular, we assume that any $k \leq d$ vectors drawn from a subspace S of dimension d are linearly independent, and we make the following definition.

Definition 8. Let S be a linear subspace of $\mathbb{R}^D$ with dimension d. A set of data $\mathbf{W}$ drawn from $S \subset \mathbb{R}^D$ with dimension d is said to be *generic* if (i) $|\mathbf{W}| > d$ and (ii) every d vectors from $\mathbf{W}$ form a basis for S. In particular, $\operatorname{span} \mathbf{W} = S$.

Another assumption that we will make is that the union of subspaces $\mathscr{U} = \bigcup_{i \in I} S_i$ from which the data is drawn consists of independent subspaces (see Definition 6).

In particular, if $\{S_i \subset \mathbb{R}^D\}_{i=1}^n$ are independent, then $S_i \cap S_j = \{0\}$ for $i \neq j$. Note that if the data $\mathbf{W} = \{w_1, \ldots, w_m\}$ is generic and is drawn from a union $\mathscr{U} = \bigcup_{i \in I} S_i$ of l independent subspaces $S_i \in \mathbb{R}^D$ of dimension d_i, then the solution to Problem 3 is precisely the subspaces S_i from which $\mathbf{W}$ is drawn. However, for this case, the solution can be obtained in a more efficient and direct way as will be developed below.

We note that to find the subspaces S_i it would suffice to find the partition $P(\mathbf{W}) = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$ of the data $\mathbf{W}$. From this partition, the subspaces can be obtained simply by $S_i = \operatorname{span} \mathbf{W}_i$. Conversely, if we knew the subspaces S_i, it would be easy to find the partition $P(\mathbf{W}) = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$ such that $\mathbf{W}_i \subset S_i$. However, all we are given is the data $\mathbf{W}$, and we do not know the partition $P(\mathbf{W})$ or the subspaces $\mathbf{W}_i$. Our goal for solving Problem 3 from this case is to find the partition $P(\mathbf{W}) = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$ of $\mathbf{W}$. To do this, we construct a matrix $\mathbf{W} = [w_1, \ldots, w_m]$ whose columns are the data vectors $w_i \in \mathbb{R}^D$. The matrix $\mathbf{W}$ is a $D \times m$ matrix, where D maybe large, thus our first

goal is to replace $\mathbf{W}$ by another matrix $\widetilde{\mathbf{W}}$ while preserving the clustering:

Proposition 4.2.1. *Let A and B be $m \times n$ and $n \times k$ matrices. Let $C = AB$. Assume $J \subset \{1, 2, \cdots, k\}$.*

1. *If $b_i \in span\left\{b_j : j \in J\right\}$ then $c_i \in span\left\{c_j : j \in J\right\}$.*
2. *If A is full rank and $m \geq n$ then $b_i \in span\left\{b_j : j \in J\right\} \iff c_i \in span\left\{c_j : j \in J\right\}$*

Proof. The relation $b_i = \sum_{j \in J} \alpha_j b_j$ implies that $Ab_i = \sum_{j \in J} \alpha_j A b_j$, and (1) follows from the fact that the columns c_l of C and b_l of B are related by $c_l = Ab_l$. For (2), we note that $A^t A$ is invertible and $(A^t A)^{-1} A^t C = B$. We then apply part (1) of the proposition. □

It can be paraphrased by saying that for any matrices A, B, C, a cluster of the columns of B is also a cluster of the columns of $C = AB$. A cluster of C however is not necessarily a cluster B, unless A has full rank.

The proposition above suggests that, for the purpose of column clustering, we can replace a matrix B by matrix C as long as A has the stated properties. Thus by choosing A appropriately, the matrix B can be replaced by a more suitable matrix C, e.g. C has fewer rows, is better conditioned or is in a format where columns can be easily clustered. One such useful format is if C is a row echelon form matrix, as will be demonstrated in reduction method of Section 4.3. In fact, the first r rows of the reduced echelon form of $C = AB$ and B are the same if B has rank r:

Proposition 4.2.2. *Let A be an $m \times n$ full rank matrix, B be an $n \times k$ matrix with $m \geq n$. Then*

$$rref(AB) = \begin{bmatrix} rref(B) \\ 0 \end{bmatrix}.$$

In particular, if B has rank r then $rref(B)$ can be obtained by the first r rows of $rref(AB)$.

In particular in the absence of noise, a data matrix $\mathbf{W}$ with the SVD $\mathbf{W} = U\Sigma V^t$ has the same reduced row echelon form as that of V^t up to its rank r. This fact together with Proposition 4.2.1 will help us devise a reduction algorithm for subspace clustering. Before proving Proposition 4.2.2,

recall that there are three elementary row operations that can be used to transform a matrix to its unique reduced row echelon form. The three elementary row operations can be performed by the elementary row operation matrices.

Proof of Propostion 4.2.2. Since the reduced row echelon form of A can be obtained by taking product of the elementary matrices corresponding to the elementary row operations, we have

$$\text{rref}(A) = E_k \cdots E_1 A = \begin{bmatrix} I_n \\ 0 \end{bmatrix}. \tag{4.2}$$

Applying the same elementary row operations to AB, we get

$$D := (E_k \cdots E_1)AB = (E_k \cdots E_1 A)B = \begin{bmatrix} I_n \\ 0 \end{bmatrix} B = \begin{bmatrix} B \\ 0 \end{bmatrix}, \tag{4.3}$$

from which we obtain

$$\text{rref}(D) = \text{rref}(AB) = \text{rref}(\begin{bmatrix} B \\ 0 \end{bmatrix}) = \begin{bmatrix} \text{rref}(B) \\ 0 \end{bmatrix}. \tag{4.4}$$

□

Corollary 4.2.3 will be utilized in the development of our subspace segmentation algorithm based on the reduced row echelon form.

Corollary 4.2.3. *Assume that* $rank(\mathbf{W}) = r$ *and let* $U\Sigma V^t$ *be the singular value decomposition of* $\mathbf{W}$. *Then*

$$rref(\mathbf{W}) = \begin{bmatrix} rref\big((V^t)_r\big) \\ 0 \end{bmatrix},$$

where $(V^t)_r$ *is the first r rows of* V^t.

Proof. Using Proposition 4.2.2, we have that

$$\text{rref}(\mathbf{W}) = \text{rref}(U^t\mathbf{W}) = \text{rref}(\Sigma V^t) = \text{rref}\begin{bmatrix} D(V^t)_r \\ 0 \end{bmatrix} = \begin{bmatrix} \text{rref}\big(D(V^t)_r\big) \\ 0 \end{bmatrix} = \begin{bmatrix} \text{rref}\big((V^t)_r\big) \\ 0 \end{bmatrix}$$

where $D = \text{diag}(\sigma_1, \ldots, \sigma_r)$ is an $r \times r$ diagonal matrix whose diagonal are the r (nonzero) singular values of $\mathbf{W}$. □

Definition 9. Matrix R is said to be the *binary reduced row echelon form* of matrix A if all non-pivot column vectors of the reduced row echelon form of A are converted to binary vectors, i.e., non-zero entries are set to one.

Theorem 4.2.4. *Let $\{S_i\}_{i=1}^k$ be a set of non-trivial linearly independent subspaces of $\mathbb{R}^D$ with corresponding dimensions $\{d_i\}_{i=1}^k$. Let $\mathbf{W} = [w_1 \cdots w_N] \in \mathbb{R}^{D\times N}$ be a matrix whose columns are drawn from $\bigcup_{i=1}^k S_i$. Assume the data is drawn from each subspace and that it is generic. Let Brref$(\mathbf{W})$ be the binary reduced row echelon form of $\mathbf{W}$. Then*

1. *The inner product (e_i, b_j) of a pivot column e_i and a non-pivot column b_j in Brref$(\mathbf{W})$ is one, if and only if the corresponding column vectors $\{w_i, w_j\}$ in $\mathbf{W}$ belong to the same subspace S_l for some $l = 1, \ldots, k$.*

2. *Moreover, $\dim(S_l) = \|b_j\|_1$, where $\|b_j\|_1$ is the ℓ_1-norm of b_j.*

3. *Finally, $w_p \in S_l$ if and only if $b_p = b_j$ or $(b_p, b_j) = 1$.*

This theorem suggests a very simple, yet effective, approach to cluster the data points. The data $\mathbf{W}$ can be partitioned into k clusters $\{\mathbf{W}_1, \ldots, \mathbf{W}_k\}$, such that span$\mathbf{W}_l = S_l$. The clusters can be formed as follows: Pick a non-pivot element b_j in Brref$(\mathbf{W})$, and group together all columns b_p in Brref$(\mathbf{W})$ such that $(b_j, b_p) > 0$. Repeat the process with a different non-pivot column until all columns are exhausted. For example, consider the following data matrix that contains data points

drawn from the union of three (3) linearly independent subspaces as column vectors.

$$\mathbf{W} = \begin{bmatrix} 2872 & 138 & 342 & 263 & 1956 & 2016 & 1793 & 801 & 195 & 360 & 1076 & 1882 & 1918 & 2350 & 83 \\ 4041 & 249 & 467 & 516 & 129 & 288 & 2612 & 769 & 312 & 174 & 241 & 176 & 3019 & 3270 & 219 \\ 2906 & 4292 & 352 & 7240 & 2861 & 3072 & 1847 & 665 & 6968 & 646 & 1709 & 2794 & 2080 & 2366 & 1012 \\ 5803 & 1405 & 657 & 2498 & 549 & 864 & 3854 & 687 & 2158 & 390 & 629 & 628 & 4711 & 4654 & 545 \\ 5124 & 744 & 2092 & 1335 & 662 & 1056 & 2835 & 1116 & 1131 & 484 & 774 & 762 & 4867 & 4546 & 309 \\ 6701 & 3192 & 757 & 5420 & 775 & 1248 & 4502 & 578 & 5148 & 578 & 919 & 896 & 5638 & 5354 & 812 \\ 7102 & 1625 & 802 & 2862 & 888 & 1440 & 4793 & 522 & 2522 & 672 & 1064 & 1030 & 6059 & 5666 & 585 \\ 495 & 223 & 117 & 577 & 322 & 960 & 266 & 247 & 169 & 668 & 866 & 520 & 275 & 430 & 388 \\ 1184 & 2910 & 192 & 8282 & 435 & 1152 & 755 & 200 & 1482 & 762 & 1011 & 654 & 951 & 970 & 6320 \\ 2065 & 1117 & 287 & 3027 & 4040 & 4800 & 1376 & 159 & 715 & 1360 & 2920 & 4100 & 1797 & 1662 & 2172 \end{bmatrix} \tag{4.5}$$

The computed (binary) reduced row echelon form of $\mathbf{W}$ is $\mathbf{W}_b$:

$$\mathbf{W}_b = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.6}$$

Then, our algorithm (Algorithm 3, below) correctly clusters columns of $\mathbf{W}$ as $\{1,3,7,8,13,14\}$, $\{2,4,9,15\}$, and $\{5,6,10,11,12\}$.

Proof of Theorem 4.2.4. The reduced row echelon form of $\mathbf{W}$ is of the form

$$\text{rref}(\mathbf{W}) = \begin{bmatrix} R \\ 0 \end{bmatrix}. \tag{4.7}$$

Let P be an $N \times N$ permutation matrix such that $\mathbf{W}P = \begin{bmatrix} U & V \end{bmatrix}$, where the columns of U are the columns associated with the pivots $\text{rref}(\mathbf{W})$ and preserving their left to right order. Thus, U

forms a basis for $\bigcup_{i=1}^{k} S_i$. This can be done, since the data is drawn from each subspace and it is generic, and that the $\{S_i\}$ are independent. In particular, U includes exactly d_i points from each S_i, and $U \subset \mathbb{R}^{D\times r}$ with rank $r = \sum_{i=1}^{k} d_i$. Moreover, because of the generic assumption of the data, $|V| \geq k$. In addition, every column of V is a linear combination of the columns of U, that is, there exists an $r \times (N-r)$ matrix Q with $V = UQ$. Therefore

$$\mathbf{W}P = \begin{bmatrix} U & V \end{bmatrix} = U \begin{bmatrix} I_r & Q \end{bmatrix}, \tag{4.8}$$

where I_r is $r \times r$ identity matrix. Let $E := E_l \cdots E_1$ be the product of elementary row operation matrices such that $E\mathbf{W}P = \text{rref}(\mathbf{W}P)$. Then,

$$E\mathbf{W}P = EU \begin{bmatrix} I_r & Q \end{bmatrix} = \begin{bmatrix} I_r & X \\ 0 & 0 \end{bmatrix}. \tag{4.9}$$

Thus $EU = \begin{bmatrix} I_r \\ 0 \end{bmatrix}$, and $X = Q$. By the choice of U above, we get that $\begin{bmatrix} I_r & Q \end{bmatrix} = RP$. It follows that, $\mathbf{W}P = U \begin{bmatrix} I_r & Q \end{bmatrix} = URP$, and since P is invertible, $\mathbf{W} = UR$.

$(e_i, b_j) = 1$ if and only if $(e_i, r_j) \neq 0$ where r_j is the column in R that corresponds to the column b_j in Brref($\mathbf{W}$). Now $r_j = \sum_{i=1}^{r} c_i e_i$. If $(e_i, r_j) \neq 0$, then $c_i \neq 0$. Thus $w_j = Ur_j = c_i w_i + \sum_{k\neq i} c_l U e_l$. If $w_i \in S_l$, then $w_i = Ue_i$ is one of the basis vectors of S_l, and since $c_i \neq 0$, independence of the subspaces implies that $w_j \in S_l$. Conversely, if $w_j = Ur_j$ and $w_i = Ue_i$ belong to the same subspace S_l, then $w_j = c_i w_i + \sum_{Ue_k \in S_l} c_k U e_k$, due to independence of the subspaces. This, together with the assumption that the data is generic implies that $c_i \neq 0$. Hence $r_j = c_i e_i + \sum_k c_k e_k$, and we get $(e_i, r_j) = c_i \neq 0$. This proves part (1).

Now let us assume that $w_j \in S_l$. Since the data is generic and subspaces are independent, w_j can be written as a linear combination of exactly d_l columns of U. This means there are d_l nonzero entries in the corresponding column r_j in R. Since all the nonzero entries are set to 1 for Brref($\mathbf{W}$), the ℓ_1-norm of the corresponding non-pivot columns must be d_l. This proves part (2).

Finally consider part (3). If w_p and w_j belong to S_l, then if $w_p = Ue_p$ then part (1) implies $(e_p, b_j) = 1$. Otherwise the fact the subspaces are independent and the data generic imply that $b_p = b_j$.

Now let b_p be a column of Brref($\mathbf{W}$) with $b_p = b_j$. Let r_p, r_j be the corresponding columns in R. Then, $w_p = Ur_p$ and $w_j = Ur_j$. Since $w_j \in S_l$, and w_p and w_j are in the span of the same column vectors of U corresponding to S_l, it follows, $w_p \in S_l$. Finally if $b_p \neq b_j$ and $(b_p, b_j) = 1$, then r_p is a pivot column of R. Part (1) then implies that $\{w_p, w_j\}$ belong to the same subspace S_l. □

4.2.1 Segmentation Algorithm for Noiseless Data

Algorithm 3 summarizes the algorithm for subspace clustering when the data points are not corrupted by noise. The algorithm can find a basis for each subspace and it correctly clusters all of the data points.

4.3 Subspace Segmentation for Noisy Data

In practice the data $\mathbf{W}$ is corrupted by noise. In this case, the rref-based algorithm cannot work, even under the assumption of Theorem 4.2.4, since the noise will have two effects: 1) The rank of the data corrupted by noise $\mathbf{W} + \eta \subset \mathbb{R}^D$ becomes full; i.e., rank$(\mathbf{W} + \eta) = D$; and 2) Even under the assumption that $r = D$, none of the entries of the non-pivot columns of rref$(\mathbf{W} + \eta)$ will be zero. One way of circumventing this problem is to use the rref-based algorithm in combination with thresholding to set to zero those entries that are small. The choice of the threshold depends on the noise characteristics and the position of the subspaces relative to each other. Thus the goal of this section to estimate this error in terms of these factors.

Let us first assume that $\mathbf{W} \subset \mathbb{R}^{D \times N}$ and that $dim(\sum_{i=1}^{k} S_i) = D$. Thus, under the assumption that the data is generic, rank$(\mathbf{W}) = D$. Without loss of generality, let us assume that $\mathbf{W} = \begin{bmatrix} A & B \end{bmatrix}$ where the columns of A form basis for $\mathbb{R}^D$, i.e., the columns of A consist of d_i linearly independent vectors from each subspace S_i, $i = 1, \ldots, k$. Let $\widetilde{\mathbf{W}} = \mathbf{W} + \mathbf{N}$ be the data with additive noise. Then

Algorithm 3 Subspace Segmentation - Row Echelon Form Approach - No Noise

Require: $D \times N$ data matrix $\mathbf{W}$.

Find rref($\mathbf{W}$) of $\mathbf{W}$.
Find Brref($\mathbf{W}$) of $\mathbf{W}$ by setting all non-zero entries of rref($\mathbf{W}$) to 1.
for all j from 1 to N **do**
    Pick the j^{th} column b_j of Brref($\mathbf{W}$).
    if b_j is pivot **then**
        continue
    end if
    for all i from 1 to $j-1$ **do**
        if b_i is non-pivot and $(b_i, b_j) > 0$ **then**
            Place $\{b_i, b_j\}$ in the same cluster C_i.
            break
        end if
    end for
end for
for all C_i **do**
    Pick any $b \in C_i$.
    Separate b into unit vectors $u_i^1, \ldots, u_i^{d_i}$. {These vectors form a basis for a subspace S_i with dimension d_i.}
    for all k from 1 to N **do**
        if $b_k \in \{u_i^1, \ldots, u_i^{d_i}\}$ **then**
            Place b_k in the same cluster C_i. {This is for handling pivot columns.}
        end if
    end for
    Place the corresponding columns in $\mathbf{W}$ into the same cluster $\mathbf{W}_i$.
end for
Renumber indices i's of S_i starting from 1.

the reduced echelon form applied to $\widetilde{\mathbf{W}}$ is given by $\text{rref}(\widetilde{\mathbf{W}}) = \begin{bmatrix} I & \widetilde{A}^{-1}\widetilde{B} \end{bmatrix}$. Let b_i and $\tilde{b}_i$ be the columns of B and $\widetilde{B}$ respectively, and let $e_i = \widetilde{A}^{-1}\tilde{b}_i - A^{-1}b_i$. Let $\Delta = \widetilde{A} - A$, and $v_i = \tilde{b}_i - b_i$ we have

$$e_i = \widetilde{A}^{-1}\tilde{b}_i - A^{-1}b_i = (I + A^{-1}\Delta)^{-1}A^{-1}(b_i + v_i) - A^{-1}b_i$$

Let $\sigma_{\min}$ denote the smallest singular value of A, then if $||\Delta|| \leq \sigma_{\min}(A)$, we get

$$\begin{aligned}
\|e_i\|_2 &= \Big\| \Big(I - A^{-1}\Delta + (A^{-1}\Delta)^2 - (A^{-1}\Delta)^3 + \dots \Big) A^{-1}(b_i + v_i) - A^{-1}b_i \Big\|_2 \\
&= \Big\| A^{-1}\varepsilon + \Big(-A^{-1}\Delta A^{-1} + (A^{-1}\Delta)^2 A^{-1} - (A^{-1}\Delta)^3 A^{-1} + \dots \Big)(b_i + v_i) \Big\|_2 \\
&\leq \|A^{-1}\| \|v_i\|_2 + \Big(\|A^{-1}\|^2 \|\Delta\| + \|A^{-1}\|^3 \|\Delta\|^2 + \|A^{-1}\|^4 \|\Delta\|^3 + \dots \Big)(||b_i||_2 + ||v_i||_2) \\
&= \frac{\|v_i\|_2}{\sigma_{\min}(A)} + \frac{\|\Delta\|}{\sigma^2_{\min}(A)} \left(\frac{1}{1 - \frac{\|\Delta\|}{\sigma_{\min}(A)}} \right) (\|b_i\|_2 + \|v_i\|_2)
\end{aligned} \tag{4.10}$$

where $\|\cdot\|$ denotes the operator norm $\|\cdot\|_{\ell^2 \to \ell^2}$. Unless specified otherwise, the noise $\mathbf{N}$ will be assumed to consist of entries that are i.i.d. $\mathcal{N}(0, \sigma^2)$ Gaussian noise with zero mean and variance σ^2. For this case, the expected value of $\|\Delta\|$ can be estimated by $\mathbb{E}\|\Delta\| \leq C\sqrt{D}\sigma$ using the following Theorem in [52, 53].

Theorem 4.3.1 (Latala's Theorem)**.** *Let A be a random matrix whose entries a_{ij} are independent and centered random variables with finite fourth moment. Then,*

$$\mathbb{E}\sigma_{\max}(A) \leq C \Big[\max_i \big(\sum_j \mathbb{E}a_{ij}^2 \big)^{1/2} + \max_j \big(\sum_i \mathbb{E}a_{ij}^2 \big)^{1/2} + \big(\sum_{i,j} \mathbb{E}a_{ij}^4 \big)^{1/4} \Big] \tag{4.11}$$

where C is a universal constant and $\sigma_{\max}$ is the largest singular value of A.

Note that to estimate the error in (4.10) we still need to estimate $\sigma_{\min}(A)$. This singular value depends on the position of the subspaces $\{S_i\}_{i=1}^k$ relative to each other which can be measured by the principle angles between them. The principle angles between two subspaces $\mathscr{F}, \mathscr{G}$, can be obtained using any pair of orthogonal bases for $\mathscr{F}, \mathscr{G}$ as described in the following Lemma [47]:

Lemma 4.3.2. *Let $\mathscr{F}$ and $\mathscr{G}$ be two subspaces of $\mathbb{R}^D$ with $p = dim(\mathscr{F}) \leq dim(\mathscr{G}) = q$. Let $Q_{\mathscr{F}} \in \mathbb{R}^{D\times p}$ and $Q_{\mathscr{G}} \in \mathbb{R}^{D\times q}$ be matrices whose columns form orthonormal bases for the subspaces $\mathscr{F}$ and $\mathscr{G}$. Let $1 \geq \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$ be the singular values of $Q_{\mathscr{F}}^t Q_{\mathscr{G}}$. Then, the principle angles are given by*

$$\theta_k = \arccos(\sigma_k) \quad k = 1, \ldots, p. \tag{4.12}$$

Theorem 4.3.3. *Let $\{S_i\}_{i=1}^k$ be independent subspaces of $\mathbb{R}^D$ with corresponding dimensions $\{d_i\}_{i=1}^k$ such that $\sum_{i=1}^k d_i = D$. Let $\{\theta_j(S_i)\}_{j=1}^{\min(d_i, D-d_i)}$ be the principle angles between S_i and $\sum_{\ell\neq i} S_\ell$. Let $A = \begin{bmatrix} a_1 & \ldots & a_D \end{bmatrix}$ be a matrix whose columns $\{a_1, \ldots, a_D\} \subset \cup_{i=1}^k S_i$ form a basis for $\mathbb{R}^D$, with $\|a_i\|_2 = 1$, $i = 1, \ldots, D$. Then,*

$$\sigma_{\min}^2(A) \leq \min_i \left(\prod_{j=1}^{\min(d_i, D-d_i)} \left(1 - \cos^2(\theta_j(S_i))\right) \right)^{1/D} \tag{4.13}$$

where $\sigma_{\min}(A)$ is the smallest singular value of A.

Corollary 4.3.4. *Under the same conditions of Theorem 4.3.3, a simpler but possibly larger upper bound is given by:*

$$\sigma_{\min}^2(A) \leq \min_i (1 - \cos(\theta_1(S_i))^{1/D} 4^{1/D}, \tag{4.14}$$

where $\theta_1(S_i)$ is the minimum angle between S_i and $\sum_{\ell\neq i} S_\ell$.

Corollary 4.3.5. *Let $\{S_i\}_{i=1}^k$ be independent subspaces of $\mathbb{R}^D$ with corresponding dimensions $\{d_i\}_{i=1}^k$ such that $\sum_{i=1}^k d_i = D$. Let $\{\theta_j(S_i)\}_{j=1}^{\min(d_i, D-d_i)}$ be the principle angles between S_i and $\sum_{\ell\neq i} S_\ell$. Let $\mathbf{W} = [w_1 \cdots w_N] \in \mathbb{R}^{D\times N}$ be a matrix whose columns are drawn from $\bigcup_{i=1}^k S_i$. Assume the data is drawn from each subspace and that it is generic. Let P be a permutation matrix such that $\mathbf{W}P = \begin{bmatrix} A_P & B_P \end{bmatrix}$, and A_P is invertible. Then*

$$\sup_P \{\sigma_{\min}^2(A_P)\} \leq \min_i \left(\prod_{j=1}^{\min(d_i, D-d_i)} \left(1 - \cos^2(\theta_j(S_i))\right) \right)^{1/D}. \tag{4.15}$$

In particular,

$$\sup_P\{\sigma^2_{\min}(A_P)\} \leq \min_i(1-\cos(\theta_1(S_i))^{1/D}4^{1/D}, \tag{4.16}$$

where $\theta_1(S_i)$ is the minimum angle between S_i and $\sum_{\ell\neq i} S_\ell$.

Remark 4.3.6. If we know an estimate of $\min_i(1-\cos(\theta_1(S_i))^{1/D}4^{1/D}$, then no matter which A_P we choose, we cannot do any better. Therefore, our best case is always controlled by this estimate. Clearly, if the minimum principle angle between the subspaces is small, we do not expect the upper bound to be high. If we know the angles between subspaces and if we choose a permutation matrix P, we may assess how good it is by checking $\sigma_{min}(A_P)$. Also, if we know the angles between subspaces, we can state the best case scenario for A_P by maximizing $\sigma_{min}(A_P)$, which cannot be bigger than $\min_i(1-\cos(\theta_1(S_i))^{1/D}4^{1/D}$.

4.3.1 Proof of Theorem 4.3.3

Theorem 4.3.7. *Let S_1 and S_2 be subspaces of $\mathbb{R}^n$ with dimensions d_1 and d_2, respectively, with $d_1 \leq d_2$. Let Q_1 and Q_2 be orthonormal bases for S_1 and S_2, respectively, $\lambda_1^2 \geq \lambda_2^2, \ldots, \geq \lambda_{d_1}^2 \geq 0$ be the singular values of $Q_1^tQ_2$, and let $A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$. Then,*

1. *If $d_2 > d_1$, then the spectrum $\sigma(A^tA) = \{1\} \cup \{1-\lambda_i^2, 1+\lambda_i^2\}_{i=1}^{d_1}$.*
2. *If $d_2 = d_1$, then the spectrum $\sigma(A^tA) = \{1-\lambda_i^2, 1+\lambda_i^2\}_{i=1}^{d_1}$.*

Remark 4.3.8. Note that Q_1 is $n \times d_1$, Q_2 is $n \times d_2$, and $d_1 + d_2 = n$.

$$Q_1^tQ_2 = U\Sigma V^t \Longrightarrow \Sigma = \begin{bmatrix} \lambda_1^2 & 0 & \ldots & 0 & \ldots & 0 \\ 0 & \lambda_2^2 & \ldots & 0 & \ldots & 0 \\ 0 & 0 & \ldots & \lambda_{d1}^2 & \ldots & 0 \end{bmatrix}_{d_1 \times d_2}$$

Let $\theta_1, \theta_2, \ldots, \theta_{d_1}$ be the principle angles between S_1 and S_2. Then, $\cos(\theta_i) = \lambda_i^2$ for $i = 1, \ldots, d_1$ by Lemma 4.3.2.

Proof of Theorem 4.3.7. A^tA is given by

$$A^tA = \begin{bmatrix} Q_1^t \\ Q_2^t \end{bmatrix} \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} = \begin{bmatrix} Q_1^tQ_1 & Q_1^tQ_2 \\ Q_2^tQ_1 & Q_2^tQ_2 \end{bmatrix} = \begin{bmatrix} I_{d_1} & C \\ C^t & I_{d_2} \end{bmatrix} \tag{4.17}$$

where $C := Q_1^tQ_2$, and I_d denotes the $d \times d$ identity matrix . Then,

$$C^tC = V\Sigma^t\Sigma V^t.$$

$$\Sigma = \begin{bmatrix} \lambda_1^2 & 0 & \dots & 0 & \dots & 0 \\ 0 & \lambda_2^2 & \dots & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_{d1}^2 & \dots & 0 \end{bmatrix}_{d_1 \times d_2}$$

$$\Sigma^t = \begin{bmatrix} \lambda_1^2 & 0 & \dots & 0 \\ 0 & \lambda_2^2 & \dots & 0 \\ 0 & 0 & \dots & \lambda_{d_1}^2 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}_{d_2 \times d_1}$$

$$\Sigma^t\Sigma = \begin{bmatrix} \lambda_1^4 & 0 & \dots & 0 & \dots & 0 \\ 0 & \lambda_2^4 & \dots & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_{d1}^4 & \dots & 0 \\ 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix}_{d_2 \times d_2}$$

So, $\Sigma^t\Sigma = diag\{\lambda_1^4, \lambda_2^4, \dots, \lambda_{d_1}^4, \underbrace{0, \dots, 0}_{d_2 - d_1}\}$, i.e., the diagonal elements are the eigenvalues of C^tC.

Using (4.17), μ^2 is an eigenvalue of A^tA, if and only if

$$\begin{bmatrix} I_{d_1} & C \\ C^t & I_{d_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mu^2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

for some $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \neq 0$ where x_1 is $d_1 \times 1$ and x_2 is $d_2 \times 1$. Thus, we have

$$Cx_2 = (\mu^2 - 1)x_1,$$

$$C^t x_1 = (\mu^2 - 1)x_2,$$

from which we have, $C^tCx_2 = (\mu^2-1)^2x_2$. Thus, if $x_2 \neq 0$ then $(\mu^2-1)^2$ belongs to the eigenvalues $\{\lambda_1^4, \lambda_2^4, \ldots, \lambda_{d_1}^4, \underbrace{0, \ldots, 0}_{d_2-d_1}\}$ of C^tC. If $x_2 = 0$ then $\mu^2 = 1$, and x_1 is an eigenvector for CC^t, corresponding to the eigenvalue $\lambda_{d_1} = 0$.

Thus, If $d_2 > d_1$, then $\sigma(A^tA) \subset \{1\} \cup \{1-\lambda_i^2, 1+\lambda_i^2\}_{i=1}^{d_1}$, and if $d_2 = d_1$, $\sigma(A^tA) \subset \{1-\lambda_i^2, 1+\lambda_i^2\}_{i=1}^{d_1}$.

To show the other inclusions, let $\lambda^4 \in \{\lambda_1^4, \lambda_2^4, \ldots, \lambda_{d_1}^4\}$ and let $x_2 \neq 0$ be the corresponding eigenvector. If $\lambda \neq 0$ define $x_1 = \frac{1}{\lambda^2}Cx_2$. Then, using (4.17) we get,

$$\begin{bmatrix} I_{d_1} & C \\ C^t & I_{d_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 + Cx_2 \\ C^t x_1 + x_2 \end{bmatrix} \tag{4.18}$$

Since $\lambda^2 x_1 = Cx_2$, we have that $\lambda^2 C^t x_1 = C^tCx_2 = \lambda^4 x_2$ so that (since $\lambda \neq 0$) we get $C^t x_1 = \lambda^2 x_2$. Thus for $\lambda \neq 0$ we have

$$\begin{bmatrix} I_{d_1} & C \\ C^t & I_{d_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 + \lambda^2 x_1 \\ \lambda^2 x_2 + x_2 \end{bmatrix} = (1+\lambda^2) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

In particular $1+\lambda^2$ is an eigenvalue of A^tA with the eigenvalue $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. If $\lambda = 0$, then C^tC is singular. Thus C, is singular as well. Let x_2 be a nonzero vector in the null space of C and define $x_1 = 0$. Then

$$\begin{bmatrix} I_{d_1} & C \\ C^t & I_{d_2} \end{bmatrix} \begin{bmatrix} 0 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0+Cx_2 \\ 0+x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ x_2 \end{bmatrix},$$

so that $1 \in \sigma(A^tA)$. Thus in all case $\lambda^4 \in \{\lambda_1^4, \lambda_2^4, \ldots, \lambda_{d_1}^4\}$ implies that $1+\lambda^2 \in \sigma(A^tA)$. A similar proof yield that $1-\lambda^2 \in \sigma(A^tA)$.

Finally, if $d_2 > d_1$, C has a nontrivial kernel. Let $x_2 \neq 0$ be such that $Cx_2 = 0$ and $x_1 = 0$. Then, an argument similar to the last one implies that $1 \in \sigma(A^tA)$. Thus we have proved that if $d_2 > d_1$ then $\{1\} \cup \{1-\lambda_i^2, 1+\lambda_i^2\}_{i=1}^{d_1} \subset \sigma(A^tA)$, and if $d_2 = d_1$, then $\{1-\lambda_i^2, 1+\lambda_i^2\}_{i=1}^{d_1} \subset \sigma(A^tA)$. $\square$

Proof of Theorem 4.3.3. We first consider two subspaces $\{S_1, S_2\} \subset \mathbb{R}^D$ with dimensions d_1 and d_2 respectively, and $d_1 + d_2 = D$. We note that if $A_P = AP$ where P is any permutation matrix, then A_P and A have the same singular values. Thus, without loss of generality, we assume that $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$, where the columns of A_1 and A_2 are unit norm bases of S_1 and S_2 respectively. Using the QR decomposition, we get

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} = \begin{bmatrix} Q_1R_1 & Q_2R_2 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} = QR$$

where Q_1 and Q_2 are orthonormal and R_1 and R_2 are upper triangular matrices with unit column vectors.

$$\det(A^tA) = \det(R^tQ^tQR) = \det(R^tR)\det(Q^tQ) \leq det(Q^tQ). \tag{4.19}$$

where for the last inequality we have used the fact that the column vectors of R_1 and R_2 have unit norm. Let $\{\tilde{\mu}_i\}_{i=1}^D$ and $\{\mu_i\}_{i=1}^D$ be the singular values of A and Q, respectively. Then, by (4.19) we get

$$\prod_i^D \tilde{\mu}_i^2 \leq \prod_i^D \mu_i^2.$$

Using Theorem 4.3.7 we get

$$\prod_i^D \tilde{\mu}_i^2 \leq \prod_i^D \mu_i^2 = (1-\lambda_1^2)(1-\lambda_2^2)\dots(1-\lambda_{d_1}^2)(1+\lambda_{d_1}^2)(1+\lambda_{d_2-1}^2)\dots(1+\lambda_1^2). \tag{4.20}$$

Thus, noting that $\tilde{\mu}_1$ is the smallest singular value for A, and using Lemma 4.3.2 we obtain

$$\sigma_{\min}^D(A) = (\tilde{\mu}_1^2)^D \leq (1-\lambda_1^4)(1-\lambda_2^4)(1-\lambda_3^4)\dots(1-\lambda_{d_1}^4) \tag{4.21}$$

$$\leq \prod_{j=1}^{d_1} \big(1-\cos^2(\theta_j(S_1))\big). \tag{4.22}$$

For the general case of k subspaces, we replace S_1 by S_i, and S_2 by $\sum_{\ell\neq i} S_\ell$, and d_1 by $\min(d_i, D-d_i)$ and let i run from 1 to k. □

Proof of Corollary 4.3.4. As in the previous proof, for two subspaces $\{S_1, S_2\} \subset \mathbb{R}^D$ with dimensions d_1 and d_2 respectively, and $d_1 + d_2 = D$, we use (4.21) to get

$$\begin{aligned}\sigma_{\min}^D(A) = (\tilde{\mu}_1^2)^D &\leq (1-\lambda_1^2)(1+\lambda_1^2)(1-\lambda_2^4)(1-\lambda_3^4)\dots(1-\lambda_{d_1}^4)\\ &\leq (1-\lambda_1^2)(1+\lambda_1^2)(1-\lambda_{d_1}^4)\\ &\leq (\mu_1^2)(1-\lambda_{d_1}^2)(2)^2.\end{aligned}$$

This implies that inequality gives

$$\sigma_{\min}(A) \leq \mu_1^{2/D} 4^{1/D} = (1-cos(\theta_1(S_1)))^{1/D} 4^{1/D}.$$

To finish the proof, as before, we replace S_1 by S_i, and S_2 by $\sum_{\ell\neq i} S_\ell$, and let i run from 1 to k. □

Figure 4.1 displays the relationship between $\tilde{\mu}_1$ and μ_1. Figure 4.2 shows the relationship between $\tilde{\mu}_1$ and θ_1, which is the first principle angle between the subspaces S_1 and S_2. This implies that when the minimum principle angle between the subspaces is large, we should try to

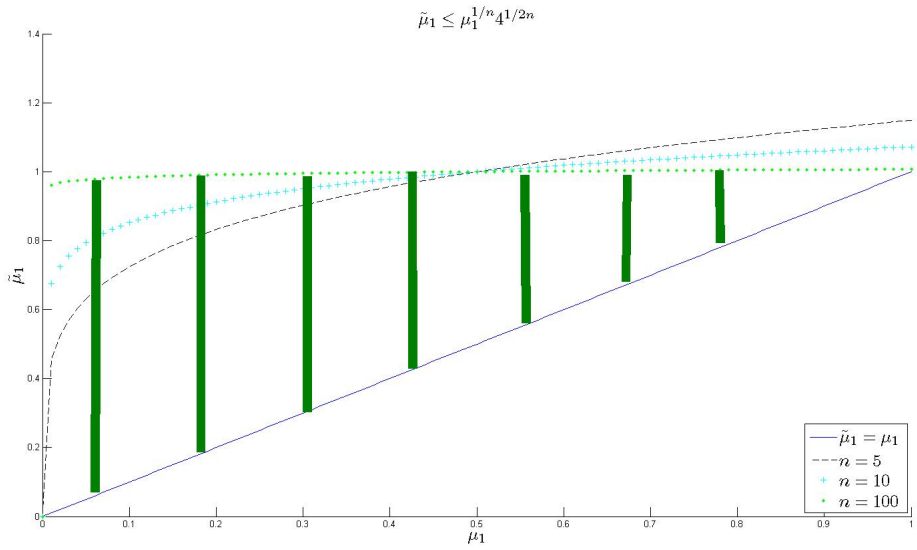

Figure 4.1: The relationship between μ_1 and $\tilde{\mu}_1$.

select orthonormal bases for Q_1 and Q_2. However, if the minimum principle angle is small, the selection of orthonormal bases does not matter too much. In other words, when μ_1 is small, it is highly unlikely that $\tilde{\mu}_1$ will be smaller than μ_1. But as μ_1 gets higher, $\tilde{\mu}_1$ is more likely to be lower than μ_1 (Figure 4.1). So as μ_1 gets higher, we should try to pick up an orthonormal basis to minimize the effect of noise. Similarly, as the angle between subspaces gets higher, we should pick an orthonormal basis (Figure 4.2).

Figures 4.3-4.5 show some simulations for Theorem 4.3.3 and Corollary 4.3.4. In Figure 4.3, two 2-dimensional subspaces of $\mathbb{R}^4$ that span $\mathbb{R}^4$ is randomly generated. Then, 7 data points from each subspace is randomly generated and they are placed as the columns of a data matrix $\mathbf{W}$. All possible 4×4 matrices from the columns of $\mathbf{W}$ are found. Among those matrices, the one with the highest minimum singular value is picked as the matrix A. All of the principle angles between the subspaces are computed and the upper bounds of Theorem 4.3.4 and Theorem 4.3.3 are calculated. This is repeated for 100 times and a scatter plot is provided in Figure 4.3. The same process is repeated for 3-dimensional and 5-dimensional subspaces of $\mathbb{R}^8$ (Figure 4.4). The same results for three 2-dimensional subspaces of $\mathbb{R}^6$ are shown in Figure 4.5.

Figure 4.6 shows the relationship between the minimum angle and the segmentation rate. For

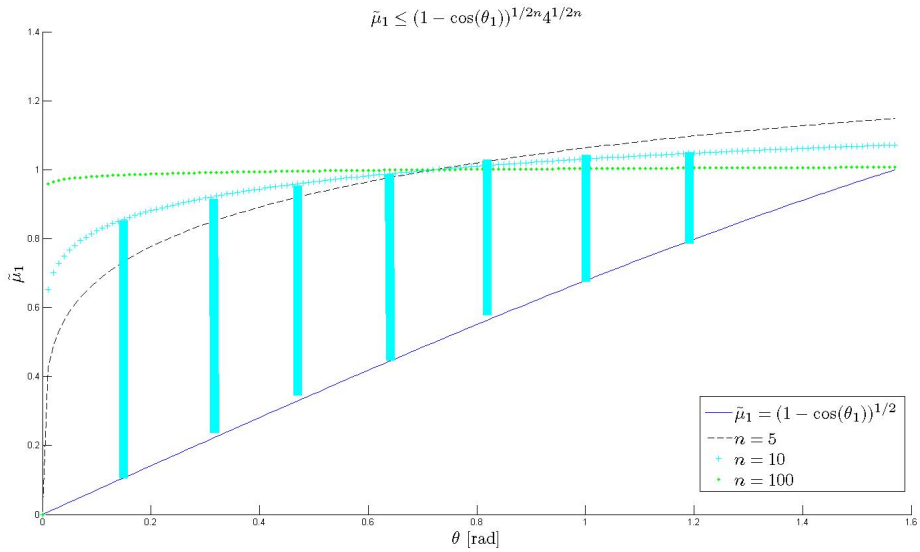

Figure 4.2: The relationship between θ_1 and $\tilde{\mu}_1$.

this simulation, 10 data points that come from two 2-dimensional subspaces of $\mathbb{R}^4$ was generated. The angles between the subspaces are computed. Then, some white noise was added to the data in a controlled fashion, i.e., the noise variance was increased from 0.00 to 0.40 with 0.01 increments. The segmentation rate for each step is calculated and then the average segmentation rate is computed. The experiment is repeated 200 times and the scatter plots for three techniques are displayed in Figure 4.6. The best-A method refers to the segmentation by using matrix A with the highest minimum singular value. The modified RREF method refers to the segmentation by giving priority to the highest pivoting rows and columns in reduced row echelon form calculations. the regular RREF method refers to the segmentation using the traditional reduced row echelon form calculation. After computing the reduced row echelon forms using those three techniques, a spectral clustering technique was applied. The similarity matrix entries consist of the inner products of the columns of the reduced row echelon form matrix. Figure 4.6 also shows the linear fitting of the scatter plots.

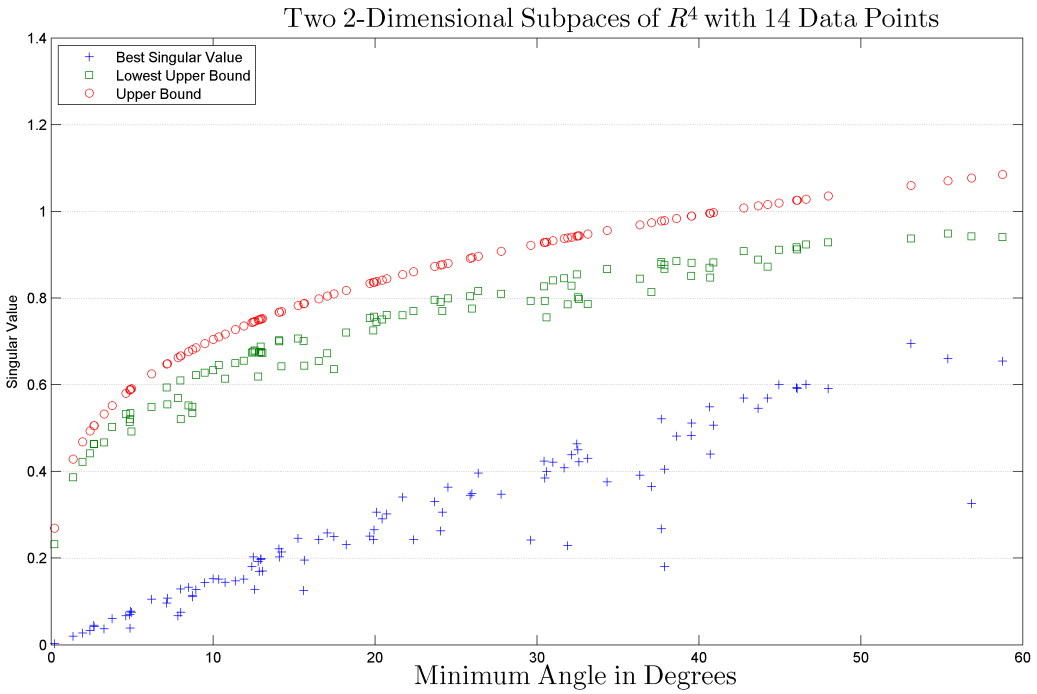

Figure 4.3: Two 2-dimensional subspaces of $\mathbb{R}^4$ with total of 14 data points.

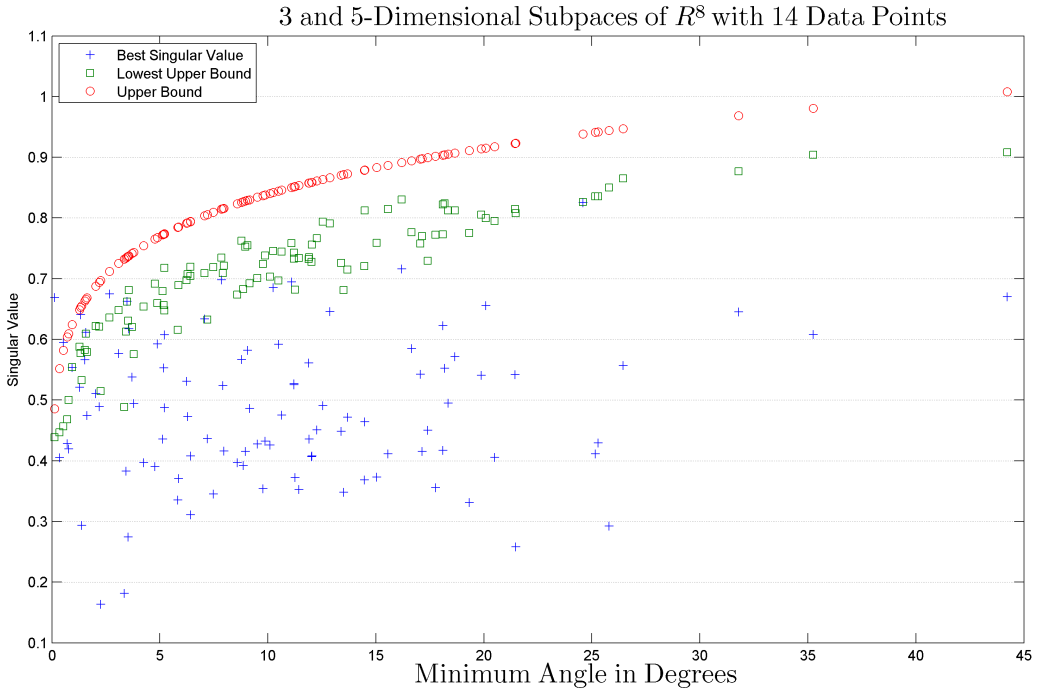

Figure 4.4: 3-dimensional and 5-dimensional subspaces of $\mathbb{R}^8$ with total of 14 data points.

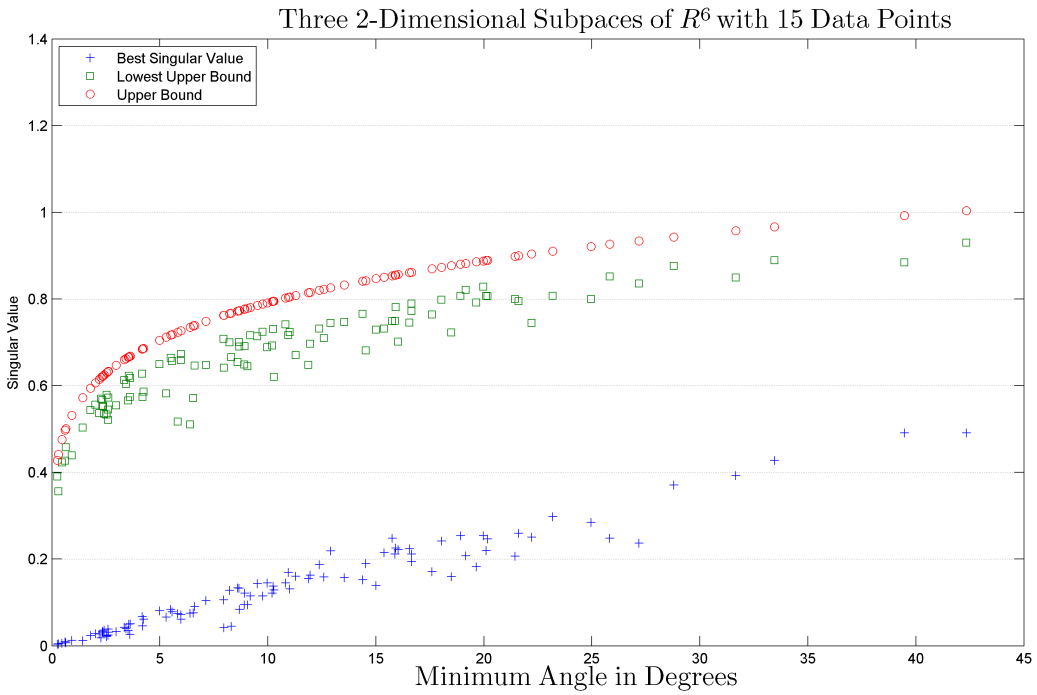

Figure 4.5: Three 2-dimensional subspaces of $\mathbb{R}^6$ with total of 15 data points.

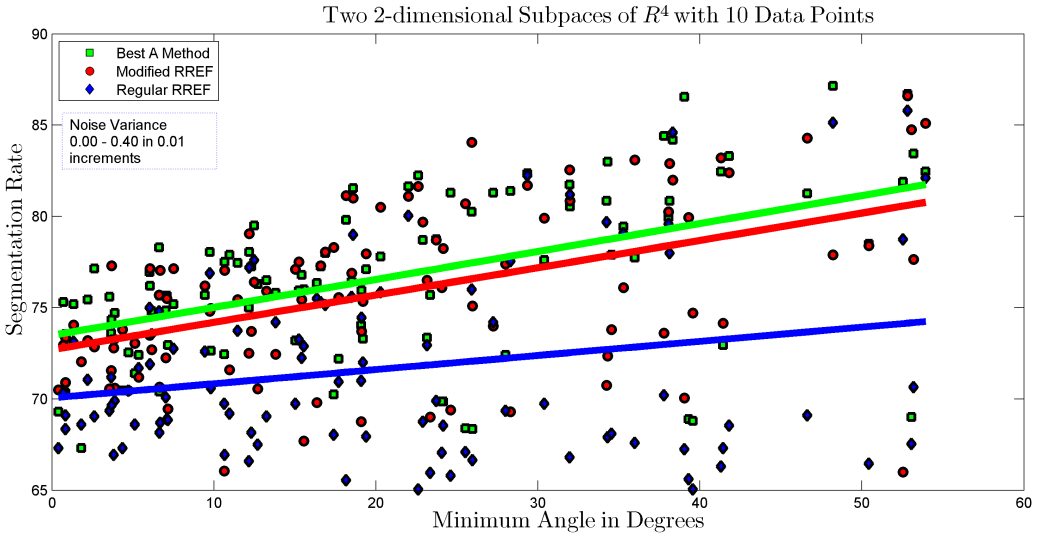

Figure 4.6: Relationship between minimum angle and three methods of RREF.

4.4 Combining Algorithms

Algorithm 2 described in Section 4.1 starts with a partition of the data matrix $\mathbf{W}$ and it may not find the global optimum solution if this initial partition is not good. Algorithm 3 described in Section 4.2.1 works perfectly for *noiseless* data (it determines a basis for each subspace and it correctly clusters all of the data points). However, it is likely to fail for noisy data due to difficulty of finding an appropriate threshold to set the small values of rref($\mathbf{W}$) to zero. rref($\mathbf{W}$) does not have the properties of those in Theorem 4.2.4, in general, and therefore cannot be used to determine the subspaces, their dimensions, or the clusters. However, the thresholded reduced echelon form can be used to determine a set of clusters that can in turn be used to determine a good initial set of subspaces in Algorithm 2.

This is achieved as follows if the number of subspaces, l, is known and each subspace is d-dimensional: First, the reduced row echelon form rref($\mathbf{W}$) of $\mathbf{W}$ is computed. Since the data is noisy, the non-pivot columns of rref($\mathbf{W}$) will most likely have all non-zero entries. The error in those entries will depend on the noise and the positions of the subspaces as in (4.3.3) and (4.3.3). Since each subspace is d-dimensional, the highest d entries of each non-pivot column is set to 1 and the all other entries are set to 0 to determine the binary reduced row echelon form Brref($\mathbf{W}$) of $\mathbf{W}$ (note that, according to Theorem 4.2.4, each non-pivot column of Brref($\mathbf{W}$) is supposed to have d entries). The next step is to have an l groups of the equivalent columns of Brref($\mathbf{W}$). Those l groups is then used as the initial partition for Algorithm 2. This process is described in Algorithm 4. Note that a dimensionality reduction is also performed (according to Corollary 4.2.3) to speed up the process.

In Step-7 of Algroithm 4, we find the subspace $S_i^o(P)$ that minimizes the expression $e(\mathbf{W}_i, S) = \sum_{w \in \mathbf{W}_i} d^p(w, S)$ for each subset $\mathbf{W}_i$ in the partition P. This can be achieved using ℓ_2-based SVD for data with light-tailed noise (e.g. Gausian distributed noise) and ℓ_1-based subspace approximation for heavy-tailed noise (e.g. Laplacian distributed noise) as described below (please see Remark 4.1.2).

Algorithm 4 Combined Algorithm - Optimal Solution $\mathbf{S}^o$

Require: Normalized data matrix $\mathbf{W}$.

Set $r = l \times d$.
Compute the SVD of $\mathbf{W}$ and find $(V_r)^t$ as in Corollary 4.2.3.
Replace the data matrix $\mathbf{W}$ with $(V_r)^t$.
Compute rref($\mathbf{W}$)
Compute Brref($\mathbf{W}$) by setting the highest d entries of each non-pivot column to 1 and all the others to 0.
Group the non-pivot equivalent columns of Brref($\mathbf{W}$) into l largest clusters $\{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$ and set the initial partition $P = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$.
For each subset $\mathbf{W}_i$ in the partition P find the subspace $S_i^o(P)$ that minimizes the expression $e(\mathbf{W}_i, S) = \sum_{w \in \mathbf{W}_i} d^p(w, S)$.
while $\sum_{i=1}^{l} e(\mathbf{W}_i, S_i^o(P)) > e(\mathbf{W}, \mathbf{S}^o(P))$ **do**
  for all i from 1 to l **do**
    Update $\mathbf{W}_i = \{w \in \mathbf{W} : d(w, S_i^o(P)) \leq d(w, S_k^o(P)),\ k = 1, \ldots, l\}$
    Update $S_i^o(P) = \underset{S}{\operatorname{argmin}}\, e(\mathbf{W}_i, S)$
  end for
  Update $P = \{\mathbf{W}_1, \ldots, \mathbf{W}_l\}$
end while
$\mathbf{S}^o = \{S_1^o(P), \ldots, S_l^o(P)\}$

4.4.1 Light-Tailed Noise

Let $\mathbf{W}$ be $D \times N$ dimensional matrix of data that is drawn from a single d dimensional subspace $S \in \mathbb{R}^D$. In order to find S, $\mathbf{W}$ can be factorized as $\mathbf{W} = UV^t$ where the columns of the $D \times k$ matrix U form a basis for S and V^t is a $k \times N$ matrix. However if the data is noisy, we must estimate U.

If the noise is additive and Gaussian, the maximum likelihood estimation of U (and V) can be stated as an optimization problem [54]. Let columns $\{W_1, \ldots, W_N\}$ of the measured data $\mathbf{W}$ be given by

$$W_i = \mathbf{w}_i + \mathbf{e}_i \qquad i = 1, ..., N \tag{4.23}$$

where $\mathbf{w}_i$ and $\mathbf{e}_i$ are the unknown vector and noise respectively. Then,

$$\mathbf{w}_i = U\mathbf{v}_i \tag{4.24}$$

for some $\mathbf{v}_i \in \mathbb{R}^k$.

Assume that the components of $\mathbf{e}_i$ are independent and $\mathbf{e}_i$ is modeled by independent and identically distributed (i.i.d) Gaussian distribution. Then,

$$p(W_i|\mathbf{w}_i) \sim \exp\{-\frac{||W_i - \mathbf{w}_i||_2^2}{\sigma^2}\} \tag{4.25}$$

where σ^2 is a scale parameter.

If we assume that $\{W_1, \ldots, W_N\}$ are independent measurements, then

$$p(W|\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N) = \prod_{i=1}^{N} p(W_i|\mathbf{w}_i) \sim \exp\{-\sum_{i=1}^{N} \frac{||W_i - U\mathbf{v}_i||_2^2}{\sigma^2}\} \tag{4.26}$$

In order to maximize (4.26), we need to minimize $\sum_{i=1}^{N} ||W_i - U\mathbf{v}_i||_2^2$. This is equivalent to minimizing

$$E(U,V) = ||\mathbf{W} - UV^t||_2^2 \tag{4.27}$$

It is known that the SVD- based matrix factorization gives the global minimum of (4.27). In Step-7 of Algroithm 4, we apply this approach for each $\mathbf{W}_i$, i.e., we factor $\mathbf{W}_i = U_i\Sigma_i V_i^t$ and assign $S_i^o(P) = \text{span}\{u_{i_1}, \ldots, u_{i_d}\}$ where $\{u_{i_1}, \ldots, u_{i_d}\}$ are the columns of U_i.

4.4.2 Heavy-Tailed Noise

In many computer vision applications such as motion segmentation and target tracking, noise is modeled as non-Gaussian heavy-tailed distribution based on empirical studies [55, 56, 57]. It is therefore important to analyze this case. Now, assume that the components of $\mathbf{e}_i$ in (4.23) are independent and $\mathbf{e}_i$ is modeled by i.i.d Laplacian distribution, which is a heavy-tailed distribution [58]. Then,

$$p(W_i|\mathbf{w}_i) \sim \exp\{-\frac{||W_i - \mathbf{w}_i||_1}{\sigma}\} \tag{4.28}$$

where σ is a scale parameter. Also,

$$p(W|\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_N)=\prod_{i=1}^{N}p(W_i|\mathbf{w}_i)\sim\exp\{-\sum_{i=1}^{N}\frac{||W_i-U\mathbf{v}_i||_1}{\sigma}\} \tag{4.29}$$

We now need to minimize $\sum_{i=1}^{N}||W_i-U\mathbf{v}_i||_1$. If we use matrix notation,

$$E(U,V)=||\mathbf{W}-UV^t||_1 \tag{4.30}$$

This is generally a non-convex optimization problem. However, if U is known, $E(U,V)$ becomes a convex function with respect to V and similarly if V is known, $E(U,V)$ becomes a convex function with respect to U. Therefore, we will need to determine U and V iteratively [54]. In Step-7 of Algroithm 4, we factor $\mathbf{W}_i=U_iV_i^t$ based on ℓ_1 norm approach as described in Algorithm 5 and assign $S_i^o(P)=\text{span}\{u_{i_1},\ldots,u_{i_d}\}$ where $\{u_{i_1},\ldots,u_{i_d}\}$ are the columns of U_i.

Algorithm 5 Iterative Solution for (4.30)

1: Initialize U by SVD: $\mathbf{W}=U\Sigma V^t$
2: **while** not converged **do**
3: $V=\underset{V}{\text{argmin}}||\mathbf{W}-UV^t||_1$
4: **for all** i from 1 to N **do**
5: $v_i=\underset{v}{\text{argmin}}||W_i-Uv||_1$. (Note that $||\mathbf{W}-UV^t||_1=\sum_{i=1}^{N}||W_i-Uv_i||_1$ where v_i^t is the i^{th} row of V.)
6: **end for**
7: $U=\underset{U}{\text{argmin}}||\mathbf{W}-UV^t||_1$
8: **for all** i from 1 to m **do**
9: $Q:=\mathbf{W}^t$
10: $u_i=\underset{u}{\text{argmin}}||Q_i-Vu||_1$. (Note that $||\mathbf{W}-UV^t||_1=||Q-VU^t||_1=\sum_{i=1}^{d}||Q_i-Vu_i||_1$ where u_i^t is the i^{th} row of U.)
11: **end for**
12: **end while**

4.4.3 Outliers and Missing Data Points

It is known that SVD-based matrix factorization cannot handle outliers and missing data [59, 54, 60, 61, 62]. ℓ_1 norm factorization approach can handle outliers robustly compared to least square

approach (ℓ_2 norm approach). Missing data points can be handled in Algorithm 5 by simply ignoring the missing data points (or steps corresponding to the missing data points).

Remark 4.4.1. In order to reduce the dimensionality of the problem, we compute the SVD of $\mathbf{W}$

$$\mathbf{W} = U\Sigma V^t \tag{4.31}$$

Algorithm 4 assumes that each subspace is d-dimensional and there are k subspaces. Therefore, it replaces $\mathbf{W}$ by $(V_r)^t$, where $r = k \times d$. If we do not know the rank r of $\mathbf{W}$, we may try to estimate it using a modal selection algorithm [36]:

$$r = \text{argmin}_r \frac{\sigma_{r+1}^2}{\sum_{i=1}^{r} \sigma_i^2} + \kappa r \tag{4.32}$$

where σ_j is the j^{th} singular value and κ is a suitable constant.

Remark 4.4.2. In Step-5 of Algroithm 4, Brref($\mathbf{W}$) is computed by setting the highest d entries of each non-pivot columns to 1 and the others to 0. If we do not know the dimensions of the subspaces, we may need to determine a threshold. Such a threshold depends on the noise level and the positions of the subspaces (please refer to (4.3.3) and (4.3.3))

4.5 Simulations and Experiments

4.5.1 Simulations

This section provides various simulations performed on synthetically generated data. The data is first added with Gaussian distributed noise (light-tailed noise). The data is then contaminated with Laplacian distributed noise (heavy-tailed noise). We also evaluated the effect of outliers and missing data points. In all of the experiments, subspaces with known dimensions are simulated to avoid computing a data driven threshold. Also, the rank of the data matrix is assumed to be known. This is to make sure that simulations evaluate intended cases properly.

Simulations - Light-Tailed Noise

Algorithm 4 is used for implementations of this section. Figure 4.7 shows a sample result for segmenting data that comes from union of two 4-dimensional subspace of $\mathbb{R}^{12}$. Each data point (each column of data matrix) was normalized using ℓ_2-norm. Gaussian distributed noise was added in each step of simulation. Since the data is normalized noise variance represent approximately percentage noise added to the data. Figure 4.8 shows another simulation for three 4-dimensional subspaces of $\mathbb{R}^{20}$ with different number of points. The algorithm is robust for around 15% noise level, which is a considerably high measurement noise rate.

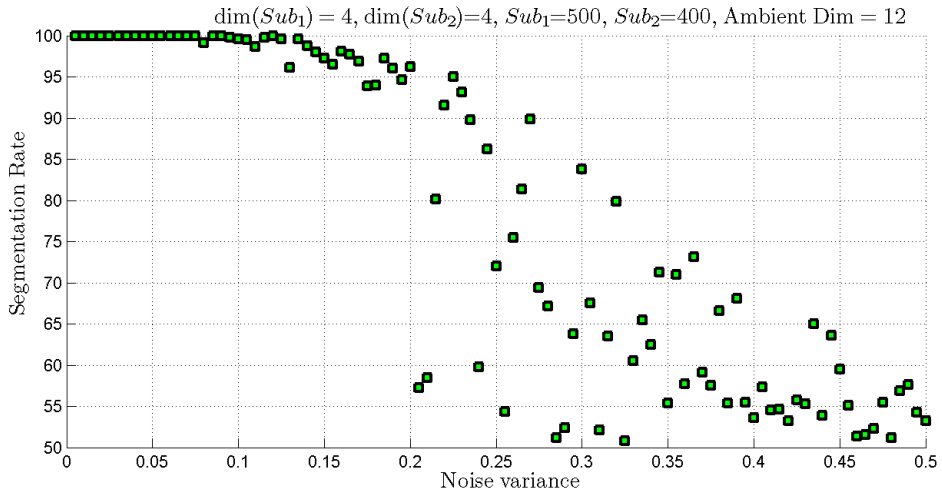

Figure 4.7: Segmentation rate for: Two subspaces of $\mathbb{R}^{12}$ with $\dim(Sub_1) = 4$, $\dim(Sub_2) = 4$, number of data points for $Sub_1 = 500$, number of data points for $Sub_2 = 300$, and contaminated with Gaussian distributed noise.

Simulations - Heavy-Tailed Noise

Figure 4.9 displays a sample result for segmenting data that comes from union of two 4-dimensional subspaces of $\mathbb{R}^{12}$. Each subspace contains 100 data points. We used a linear programming software library for implementing Algorithm 5. It is shown that the algorithm is robust for almost 15% noise level.

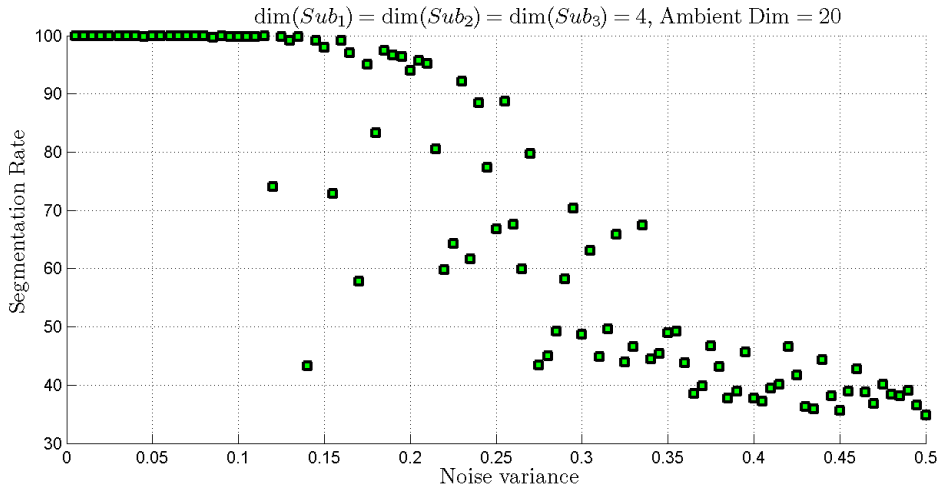

Figure 4.8: Segmentation rate for: Three subspaces of $\mathbb{R}^{20}$ with $\dim(Sub_1) = 4$, $\dim(Sub_2) = 4$, $\dim(Sub_3) = 4$, number of data points for $Sub_1 = 300$, $Sub_2 = 400$, $Sub_3 = 500$, and contaminated with Gaussian distributed noise.

Simulations - Outliers

Figure 4.10 shows the segmentation rates for noise-free data with outliers. In order to generate the outliers, certain number of data points from each subspace are randomly picked. Then, those points are randomly corrupted and Algorithm 5 is applied to the corrupted data. The data contains only outliers but no noise.

Discussion of Simulation Results

The simulations confirm validity of the proposed algorithms. The data matrix $\mathbf{W}$ of each subspace is factored as $\mathbf{W} = UV^t$, where U forms a basis for the best approximation of the subspace and V^t contains the projection of $\mathbf{W}$ onto the subspace spanned by the columns of U. SVD-based data matrix factorization is used for handling Gaussian noise and ℓ_1-norm based factorization is used for Laplacian noise as well as for handling outliers.

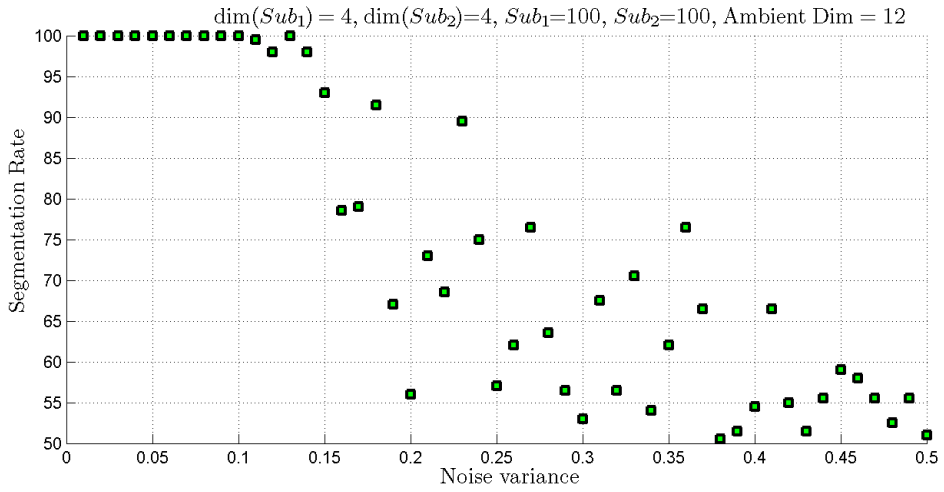

Figure 4.9: Segmentation rate for: Two subspaces of $\mathbb{R}^{12}$ with $\dim(Sub_1) = 4$, $\dim(Sub_2) = 4$, number of data points for $Sub_1 = 100$, number of data points for $Sub_2 = 100$, and contaminated with Laplacian distributed noise.

4.5.2 Experiments

The Hopkins 155 Dataset

The Hopkins 155 Dataset [25] was created as a benchmark database to evaluate motion segmentation algorithms. It contains two (2) and three (3) motion sequences. There are three (3) groups of video sequences in the dataset: (1) 38 sequences of outdoor traffic scenes captured by a moving camera, (2) 104 indoor checker board sequences captured by a handheld camera, and (3) 13 sequences of articulated motions such as head and face motions. Cornerness features that are extracted and tracked across the frames are provided along with the dataset. The ground truth segmentations are also provided for comparison. Figure 4.11 shows two (2) samples from the dataset with the extracted features.

Experimental Results

Table 4.1 displays the results when Algorithm 4 (with SVD-based subspace approximation) is applied to the two-motion data from the Hopkins 155 Dataset. Th RREF-based algorithm is extremely fast and works well with two-motion video sequences. The average error for all two-

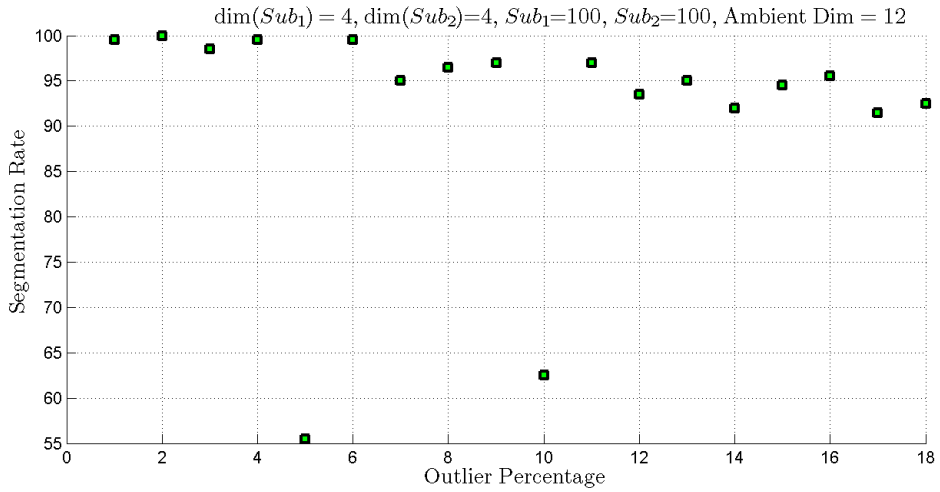

Figure 4.10: Outliers versus segmentation rate for: Two subspaces of $\mathbb{R}^{12}$ with $\dim(Sub_1) = 4$, $\dim(Sub_2) = 4$, number of data points for $Sub_1 = 100$, number of data points for $Sub_2 = 100$.

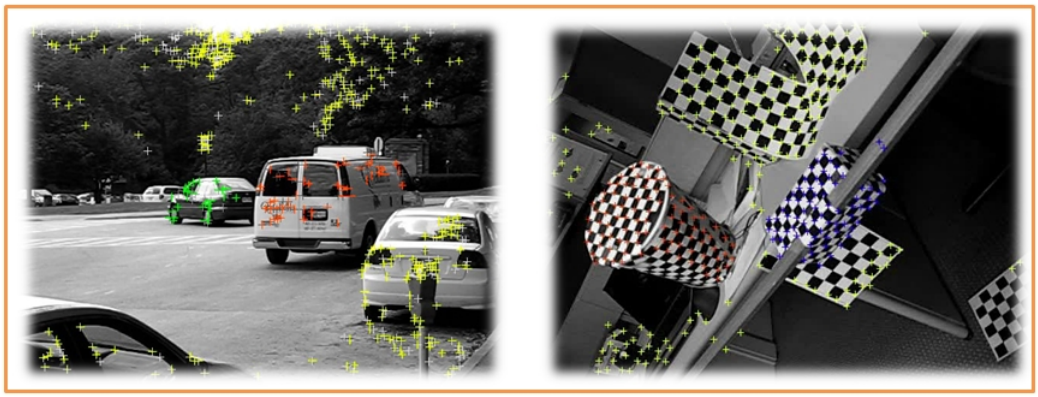

Figure 4.11: Samples from the Hopkins 155 Dataset.

motion sequences is 11.45%. However, the error is very high for three-motion sequences and obviously it does not work well with such video sequences. We believe that this is due to unknown nature of the noise in data.

Checker (78)	RREF-Based Approach
Average	8.81%
Median	5.44%
Traffic (31)	RREF-Based Approach
Average	16.04%
Median	11.94%
Articulated (11)	RREF-Based Approach
Average	17.25%
Median	12.69%
All (120 seq)	RREF-Based Approach
Average	11.45%
Median	6.78%

Table 4.1: % segmentation errors for sequences with two motions.

CHAPTER 5

NEARNESS TO LOCAL SUBSPACE APPROACH

We provided a detailed treatment of the motion segmentation problem as a special case of the subspace segmentation problem in Chapter 3. First, a $2F \times N$ data matrix $\mathbf{W}$ is constructed using N feature points that are tracked across F frames. Then, each column of $\mathbf{W}$ (i.e., the trajectory vector of a feature point) is treated as a data point and it is shown that all of the data points that correspond to the same moving object lie in an at most 4-dimensional subspace of $\mathbb{R}^{2F}$.

In this chapter, we develop a specialized algorithm for the case when the dimensions of the subspaces are equal and known. Such cases occur in many data clustering problems, such as motion segmentation and face recognition. The algorithm is reliable in the presence of noise and applied to the Hopkins 155 Dataset, it generates the best results to date for motion segmentation.

5.1 Algorithm for Subspace Segmentation

In this section, we develop a specialized algorithm for subspace segmentation and data clustering when the dimensions of the subspaces are equal and known. First, a local subspace is estimated for each data point. Then, the distances between the local subpaces and points are computed and a distance matrix is generated. This is followed by construction of a binary similarity matrix by applying a data-driven threshold to the distance matrix. Finally, the segmentation problem is converted to a one-dimensional data clustering problem. The precise steps are described in Algorithm 6 and in the explanation that follows.

5.1.1 Dimensionality Reduction and Normalization

Let $\mathbf{W}$ be an $D \times N$ data matrix whose columns are drawn from a union of subspaces of dimensions at most d, possibly perturbed by noise. In order to reduce the dimensionality of the problem, we

Algorithm 6 Subspace Segmentation

Require: The $D \times N$ data matrix $\mathbf{W}$ whose columns are drawn from subspaces of dimension d

Ensure: Clustering of the feature points.

1: Compute the SVD of $\mathbf{W}$ as in (5.1).
2: Estimate the rank of $\mathbf{W}$ (denoted by r) if it is not known. For example, using (5.2) or any other appropriate choice.
3: Compute $(V_r)^t$ consisting of the first r rows of V^t.
4: Normalize the columns of $(V_r)^t$.
5: Replace the data matrix $\mathbf{W}$ with $(V_r)^t$.
6: Find the angle between the column vectors of $\mathbf{W}$ and represent it as a matrix. {i.e., $\arccos(\mathbf{W}^t\mathbf{W})$.}
7: Sort the angles and find the closest neighbors of column vector.
8: **for all** Column vector x_i of $\mathbf{W}$ **do**
9: Find the local subspace for the set consisting of x_i and k neighbors (see (5.3)). {Theoretically, k is at least $d-1$. We can use the least square approximation for the subspace (see the section *Local Subspace Estimation*). Let A_i denote the matrix whose columns form an orthonormal bases for the local subspace associated with x_i.}
10: **end for**
11: **for** $i = 1$ to N **do**
12: **for** $j = 1$ to N **do**
13: define $H = (d_{ij}) = \left(||x_j - A_i^t x_j||_p + ||x_i - A_j^t x_i||_p \right)/2$
14: **end for**
15: **end for**{Build the distance matrix}
16: Sort the entries of the $N \times N$ matrix H from smallest to highest values into the vector h and set the threshold η to the value of the T^{th} entry of the sorted and normalized vector h, where T is such that $\|\chi_{[T,N^2]} - h\|_2$ is minimized, and where $\chi_{[T,N^2]}$ is the characteristic function of the discrete set $[T,N^2]$.
17: Construct a similarity matrix S by setting all entries of H less than threshold η to 1 and by setting all other entries to 0. {Build the binary similarity matrix}
18: Normalize the rows of S using ℓ_1-norm.
19: Perform SVD $S^t = U_n \Sigma_n (V_n)^t$.
20: Cluster the columns of $\Sigma_n (V_n)^t$ using k-means. $\Sigma_n (V_n)^t$ is the projection on to the span of U_n.

compute the SVD of $\mathbf{W}$

$$\mathbf{W} = U\Sigma V^t \tag{5.1}$$

where $U = \begin{bmatrix} u_1 & u_2 & \cdots & u_D \end{bmatrix}$ is an $D \times D$ matrix, $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_N \end{bmatrix}$ is an $N \times N$ matrix, and Σ is an $D \times N$ diagonal matrix with diagonal entries $\sigma_1, \ldots, \sigma_l$, where $l = \min\{D, N\}$.

To estimate the effective rank of $\mathbf{W}$, one can use the modal selection algorithm [36] to estimate the rank r if it is not known:

$$r = \text{argmin}_r \frac{\sigma_{r+1}^2}{\sum_{i=1}^{r} \sigma_i^2} + \kappa r \tag{5.2}$$

where σ_j is the j^{th} singular value and κ is a suitable constant. Another possible model selection algorithm can be found in [63]. $U_r \Sigma_r (V_r)^t$ is the best rank-r approximation of $W = U\Sigma V^t$, where U_r refers to a matrix that has the first r columns of U as its columns and V_r refers to the first r rows of V^t. In the case of motion segmentation, if there are k independent motions across the frames captured by a moving camera, the rank of $\mathbf{W}$ is between $2(k+1)$ and $4(k+1)$.

We can now replace the data matrix $\mathbf{W}$ with the matrix $(V_r)^t$ that consists of the first r rows of V^t (thereby reducing the dimensionality of data). This step is justified by Proposition 4.2.1. Also, [24] discusses the segmentation preserving projections and states that the number of subspaces and their dimensions are preserved by random projections, except for a zero measure set of projections. It should also be noted that this step reduces additive noise as well, especially in the case of light-tailed noise, e.g., Gaussian noise. The number of subspaces corresponds to the number of moving objects. Vidal *et al.* [64] uses an alternative method (power method) for SVD to project incomplete motion data (trajectories) into a 5-dimensional subspace and then applies GPCA and spectral clustering for subspace segmentation. Dimensionality reduction corresponds to Steps 1, 2, and 3 in Algorithm 6.

Another type of data reduction is normalization (Figure 5.1). Specifically, the columns of $(V_r)^t$ are normalized to lie on the unit sphere $\mathbb{S}^{r-1}$. This is because by projecting the subspace on the unit sphere, we effectively reduce the dimensionality of the data by one. Moreover, the normalization gives equal contribution of the data matrix columns to the description of the subspaces. Note

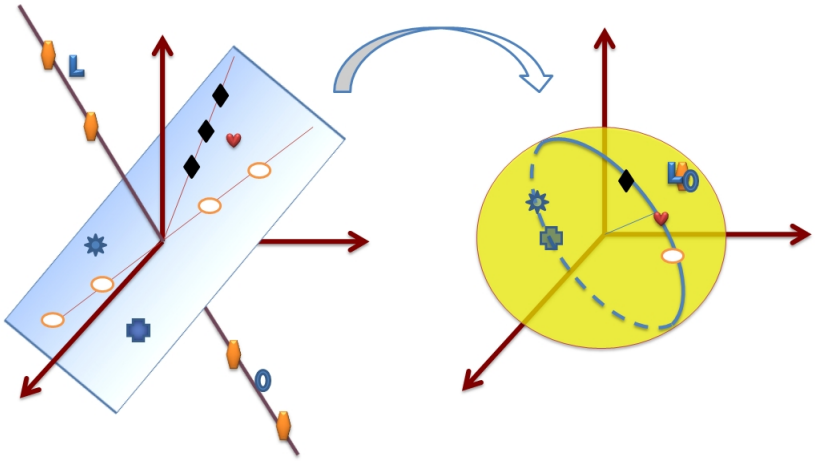

Figure 5.1: Projection onto unit sphere in $\mathbb{R}^3$.

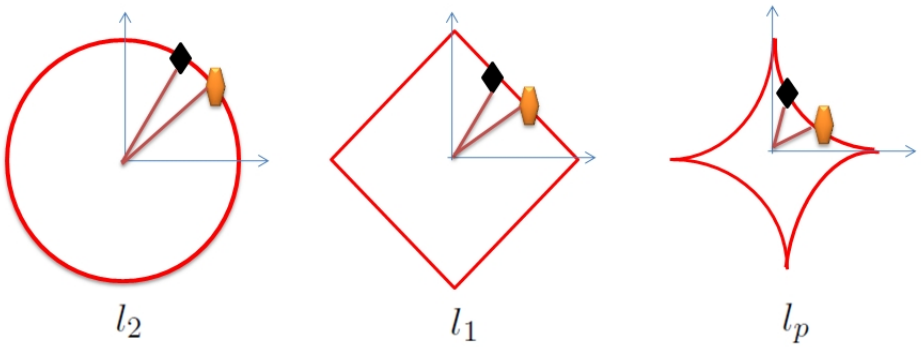

Figure 5.2: ℓ_2, ℓ_1, and ℓ_p balls with $p < 1$.

that the normalization can be done by using l_p norms of the columns of $(V_r)^t$ (Figure 5.2). This normalization procedure corresponds to Steps 4 and 5 in Algorithm 6.

5.1.2 Local Subspace Estimation

The data points (i.e., each column vector of $(V_r)^t$) that are close to each other are likely to belong to the same subspace. For this reason, we estimate a local subspace for each data point using its closest neighbors. This can be done in different ways. For example, if the ℓ_2-norm is used for normalization, we can find the angles between the points, i.e., we can compute the matrix $\arccos(V_r \times (V_r)^t)$. Then we can sort the angles and find the closest neighbors of each point. If we use l_p-norm for normalization, we can generate a distance matrix $(a_{ij}) = (||x_i - x_j||_p)$ and then

sort each column of the distance matrix to find the neighbors of each x_i, which is the i^{th} column of $(V_r)^t$.

Once the distance matrix between the points is generated, we can find, for each point x_i, a set of $k+1 \geq d$ points $\{x_i, x_{i_1}, ..., x_{i_k}\}$ consisting of x_i and its k closest neighbors. Then we generate a d-dimensional subspace that is nearest (in the least square sense) to the data $\{x_i, x_{i_1}, ..., x_{i_k}\}$. This is accomplished by using SVD

$$X = [x_i \, x_{i_1} \, ... \, x_{i_k}] = A\Sigma B^t. \tag{5.3}$$

Let A_i denote the matrix of the first d columns of A associated with x_i. Then, the column space $C(A_i)$ is the d-dimensional subspace nearest to $\{x_i, x_{i_1}, ..., x_{i_k}\}$. Local subspace estimation corresponds to Steps 6 to 10 in Algorithm 6.

5.1.3 Construction of Binary Similarity Matrix

So far, we have associated a local subspace S_i to each point x_i. Ideally, the points and only those points that belong to the same subspace as x_i should have zero distance from S_i. This suggests computing the distance of each point x_j to the local subspace S_i and forming a distance matrix H.

The distance matrix H is generated as $H = (d_{ij}) = \left(||x_j - A_i^t x_j||_p + ||x_i - A_j^t x_i||_p \right) /2$.
A convenient choice of p is 2. Note that as d_{ij} decreases, the probability of having x_j on the same subspace as x_i increases. Moreover, for $p = 2$, $||x_j - A_i^t x_j||_2$ is the Euclidean distance of x_j to the subspace associated with x_i.

Since we are not in the ideal case, a point x_j that belongs to the same subspace as x_i may have non-zero distance to S_i. However, this distance is likely to be small compared to the distance between x_j and S_k if x_j and x_k do not belong to the same subspace. This suggests that we compute a threshold that will distinguish between these two cases and transform the distance matrix into a binary matrix in which a zero in the (i, j) entry means x_i and x_j are likely to belong to the same subspace, whereas (i, j) entry of one means x_i and x_j are not likely to belong to the same subspace.

To do this, we convert the distance matrix $H = (d_{ij})_{N \times N}$ into a binary similarity matrix $S = (s_{ij})$. This is done by applying a data-driven thresholding as follows:

1. Create a vector h that contains the sorted entries of $H_{N\times N}$ from smallest to highest values. Scale h so that its smallest value is zero and its largest value is one.

2. Set the threshold η to the value of the T^{th} entry of the sorted vector h, where T is such that $\|\chi_{[T,N^2]} - h\|_2$ is minimized, and where $\chi_{[T,N^2]}$ is the characteristic function of the discrete set $[T,N^2]$. If the number of points in each subspace are approximately equal, then we would expect about $\frac{N}{n}$ points in each subspace, and we would expect $\frac{N^2}{n^2}$ small entries (zero entries ideally). However, this may not be the case in general. For this reason, we compute the data-driven threshold η that distinguishes the small entries from the large entries.

3. Create a similarity matrix S from H such that all entries of H less than the threshold η are set to 1 and the others are set to 0.

The construction of binary similary corresponds to Steps 11 to 17 in Algorithm 6. In [36], Yan and Pollofeys uses chordal distance (as defined in [48]) between the subspaces $\mathscr{F}(x_i)$ and $\mathscr{G}(x_j)$ as a measure of the distance between points x_i and x_j

$$d_c^2(\mathscr{F},\mathscr{G}) = \sum_{i=1}^{p} \sin^2(\theta_i) \tag{5.4}$$

where $\{\theta_i\}_{i=1}^{p}$ are the principle angles between p-dimensional local subspaces $\mathscr{F}$ and $\mathscr{G}$ with $\theta_1 \leq \cdots \leq \theta_p$. In this approach, the distance between any pairs of points from $\mathscr{F}$ and $\mathscr{G}$ is the same. We find distances between points and local subspaces and our approach distinguishes different points from the same subspace. To see this, let $v \in span\{Q_{\mathscr{F}}\}$, $||v||_2 = 1$, where the columns of $Q_{\mathscr{F}}$ form an orthonormal basis for $\mathscr{F}$. Thus $v = Q_{\mathscr{F}}x$ for some x with $||x||_2 = 1$. Let $Q_{\mathscr{G}}$ form an

orthonormal basis for $\mathscr{G}$, then the Euclidian distance from v to $\mathscr{G}$ squared is given by

$$\begin{aligned}
\|v - P_{\mathscr{G}}(v)||_2^2 &= \|Q_{\mathscr{F}}x - Q_{\mathscr{G}}Q_{\mathscr{G}}^t Q_{\mathscr{F}}x\|_2^2 \\
&= ||x||_2^2 - x^t Q_{\mathscr{F}}^t Q_{\mathscr{G}} Q_{\mathscr{G}}^t Q_{\mathscr{F}} x \\
&= ||x||_2^2 - x^t Y \Sigma Z^t Z \Sigma^t Y^t x \\
&= x^t Y Y^t x - x^t Y \Sigma \Sigma^t Y^t x \\
&= x^t Y Y^t x - x^t Y \Sigma^2 Y^t x \\
&= z\left(I - \Sigma^2\right) z
\end{aligned}$$

where $Y\Sigma Z^t$ is the SVD for $Q_{\mathscr{F}}^t Q_{\mathscr{G}}$ and $z := Y^t x$. Thus, using the relation $\cos\theta_i = \sigma_i$ between principle angles and singular values [47], we get

$$d^2(v,\mathscr{G}) = \sum_{i=1}^{p} z_i^2 \sin^2(\theta_i). \tag{5.5}$$

Hence, our approach discriminates distances from points in $\mathscr{F}$ to subspace $\mathscr{G}$. We also have $\sum_{i=1}^{p} z_i^2 \sin^2(\theta_i) \leq \sum_{i=1}^{p} \sin^2(\theta_i)$ and therefore d_c is more sensitive to noise.

Using (5.5), we get $0 < \sin\theta_1 \leq d \leq \sin\theta_p$. Assuming a uniform distribution of samples from $\mathscr{F}$ and $\mathscr{G}$, h can be approximated by a function depicted in Figure 5.1.3. The goal is to find the threshold at the jump discontinuity T from 0 to $\sin\theta_1$. Our method minimizes the highlighted area. Under this model, a simple computation shows that our data driven thresholding algorithm picks $T_d = T$ for $\sin\theta_1 / \sin\theta_p \geq 1/2$, e.g., if $\theta_1 \geq 30^o$. In other situations, our algorithm overshoots in estimating the threshold index depending on θ_1 and θ_p.

5.1.4 Segmentation

The last step is to use the similarity matrix S to segment the data. To do this, we first normalize the rows of S using ℓ_1-norm, i.e., $\tilde{S} = D^{-1}S$, where D is a diagonal matrix $(d_{ij}) = \sum_{j=1}^{N} s_{ij}$. Note that S and $\tilde{S}$ are not symmetric. $\tilde{S}$ is related to the random walk Laplacian L_r ($\tilde{S} = I - L_r$) [65].

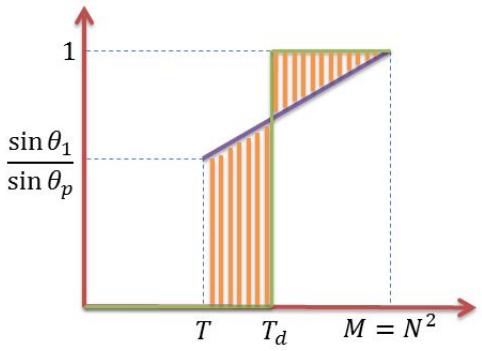

Figure 5.3: Linear modeling for h

Although other l_p normalizations are possible for $p \geq 1$, however, because of the geometry of the ℓ_1 ball, ℓ_1-normalization brings outliers closer to the cluster clouds (distances of outliers decrease monotonically as p decreases to 1). Since SVD (which will be used next) is associated with ℓ_2 minimization it is sensitive to outliers. Therefore ℓ_1 normalization works best when SVD is used.

Observe that the initial data segmentation problem has now been converted to segmentation of n 1-dimensional subspaces from the rows of $\tilde{S}$. This is because, in the ideal case, from the construction of $\tilde{S}$, if x_i and x_j are in the same subspace, the i^{th} and j^{th} rows of $\tilde{S}$ are equal. Since there are n subspaces, then there will be n 1-dimensional subspaces.

Now, the problem is again a subspace segmentation problem, but this time the data matrix is $\tilde{S}$ with each row as a data point. Also, each subspace is 1-dimensional and there are n subspaces. Therefore, we can apply SVD again to obtain

$$\tilde{S}^t = U_n \Sigma_n (V_n)^t.$$

Using Proposition 4.2.1, it can be shown that $\Sigma_n(V_n)^t$ can replace $\tilde{S}^t$ and we cluster the columns of $\Sigma_n(V_n)^t$, which is the projection of $\tilde{S}$ on to the span of U_n. Since the problem is only segmentation of subspaces of dimension 1, we can use any traditional segmentation algorithm such as k-means to cluster the data points. The segmentation corresponds to Steps 18 to 20 in Algorithm 6.

5.2 Experimental Results

Tables 5.1, 5.2, and 5.3 display some of the experimental results for the Hopkins 155 Dataset (please see Section 4.5.2 for more information about the Hopkins 155 Dataset). Our Nearness to Local Subspace (NLS) approach have been compared with six (6) motion detection algorithms: (1) GPCA [24], (2) RANSAC [30], (3) Local Subspace Affinity (LSA) [36], (4) MLS [2, 28], (5) Agglomerative Lossy Compression (ALC) [66], and (6) Sparse Subspace Clustering (SSC) [18]. An evaluation of those algorithms is presented in [18] with a minor error in the tabulated results for articulated three motion analysis of SSC-N. SSC-B and SSC-N correspond to Bernoulli and Normal random projections, respectively [18]. The minor error in [18] is the listing of error as 1.42% for articulated three motions. It is replaced with 1.60% in Table 5.2. In Tables 5.1-5.3, we used the number of neighbors $k = 3$. Since each point is drawn from a 4-dimensional subspace, a minimum of 3 neighbors are needed to fit a local subspace for each point. Using the same assumption as the algorithms that we compare with, we take the rank of the data matrix to be 8 for two motion and 12 for three motion. Table 5.1 displays the misclassification rates for the two motions video sequences. NLS outperforms all of the algorithms for the checkerboard sequences, which are linearly independent motions. The overall misclassification rate is 0.57%. This is 24% better than the next best algorithm. Table 5.2 shows the misclassification rates for the three motion sequences. NLS has 1.31% misclassification rate and performs 47% better than the next best algorithm (i.e. SSC-N). Table 5.3 presents the misclassification rates for all of the video sequences. Our algorithm NLS (with 0.76% misclassification rate) performs 39% better than the next best algorithm (i.e. SSC-N). In general, our algorithms outperforms SSC-N, which is given as the best algorithm for the two and three motion sequences together.

Table 5.4 shows the performance of the data driven threshold index T_d compared to various other possible thresholds. We provide the results for $\pm 20\%$, $\pm 10\%$, and $\pm 5\%$ deviations from T_d.

Table 5.5 displays the robustness of the algorithm with respect to the number of neighbors k. The second portion of the table excludes one pathological sequence from two-motion checker

Checker (78)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Average	6.09%	2.57%	6.52%	4.46%	1.55%	0.83%	1.12%	0.23%
Median	1.03%	0.27%	1.75%	0.00%	0.29%	0.00%	0.00%	0.00%
Traffic (31)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Average	1.41%	5.43%	2.55%	2.23%	1.59%	0.23%	0.02%	1.40%
Median	0.00%	1.48%	0.21%	0.00%	1.17%	0.00%	0.00%	0.00%
Articulated (11)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Average	2.88%	4.10%	7.25%	7.23%	10.70%	1.63%	0.62%	1.77%
Median	0.00%	1.22%	2.64%	0.00%	0.95%	0.00%	0.00%	0.88%
All (120 seq)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Average	4.59%	3.45%	5.56%	4.14%	2.40%	0.75%	0.82%	**0.57%**
Median	0.38%	0.59%	1.18%	0.00%	0.43%	0.00%	0.00%	0.00%

Table 5.1: % segmentation errors for sequences with two motions.

sequence for $k = 4$ and $k = 5$. When k is set to 3 - which is the minimum number of neighbors required - the algorithm performs better.

Table 5.6 displays the increase in the performance of the original LSA algorithm when our distance/similarity and segmentation techniques are applied separately. Both of them improves the performance of the algorithm, however, the new distance and similarity combination contributes more than the new segmentation technique.

Recently, the Low-Rank Representation (LRR) in [22, 20] was applied to the Hopkins 155 Datasets and it generated an error rate of 3.16%. The authors state that this error rate can be reduced to 0.87% by using a variation of LRR with some additional adjustment of a certain parameter.

Checker (26)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Average	31.95%	5.80%	25.78%	10.38%	5.20%	4.49%	2.97%	0.87%
Median	32.93%	1.77%	26.00%	4.61%	0.67%	0.54%	0.27%	0.35%
Traffic (7)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Average	19.83%	25.07%	12.83%	1.80%	7.75%	0.61%	0.58%	1.86%
Median	19.55%	23.79%	11.45%	0.00%	0.49%	0.00%	0.00%	1.53%
Articulated (2)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Average	16.85%	7.25%	21.38%	2.71%	21.08%	1.60%	1.60%	5.12%
Median	16.85%	7.25%	21.38%	2.71%	21.08%	1.60%	1.60%	5.12%
All (35 seq)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Average	28.66%	9.73%	22.94%	8.23%	6.69%	3.55%	2.45%	**1.31%**
Median	28.26%	2.33%	22.03%	1.76%	0.67%	0.25%	0.20%	0.45%

Table 5.2: % segmentation errors for sequences with three motions.

All (155 seq)	GPCA	LSA	RANSAC	MSL	ALC	SSC-B	SSC-N	NLS
Average	10.34%	4.94%	9.76%	5.03%	3.56%	1.45%	1.24%	**0.76%**
Median	2.54%	0.90%	3.21%	0.00%	0.50%	0.00%	0.00%	0.20%

Table 5.3: % segmentation errors for all sequences.

All-2 (120 seq)	Data Driven T_d	$0.8T_d$	$0.9T_d$	$0.95T_d$	$1.05T_d$	$1.10T_d$	$1.20T_d$
Average	0.57%	0.95%	1.17%	0.62%	0.58%	1.05%	0.77%
Median	0.00%	0.00%	0.35%	2.27%	2.27%	0.00%	0.00%
All-3 (35 seq)	Data Driven T_d	$0.8T_d$	$0.9T_d$	$0.95T_d$	$1.05T_d$	$1.10T_d$	$1.20T_d$
Average	1.31%	4.39%	3.18%	1.42%	1.20%	1.24%	2.06%
Median	0.45%	0.60%	0.57%	0.46%	0.45%	0.42%	0.37%
All (155 seq)	Data Driven T_d	$0.8T_d$	$0.9T_d$	$0.95T_d$	$1.05T_d$	$1.10T_d$	$1.20T_d$
Average	0.76%	1.84%	1.67%	0.83%	0.74%	1.10%	1.11%
Median	0.20%	0.00%	0.00%	0.20%	0.20%	0.18%	0.19%

Table 5.4: % comparison of the data driven threshold index T_d with other choices.

	ALL SEQ INCLUDED			*1 SEQ EXCLUDED*	
Checker-2 (78)	k=5	k=4	k=3	k=5	k=4
Average	0.65%	1.59%	0.23%	0.23%	0.97%
Median	0.00%	0.00%	0.00%	0.00%	0.00%
Traffic-2 (31)	k=5	k=4	k=3	k=5	k=4
Average	1.56%	1.66%	1.40%	1.56%	1.66%
Median	0.00%	0.00%	0.00%	0.00%	0.00%
Articulated-2 (11)	k=5	k=4	k=3	k=5	k=4
Average	2.44%	2.33%	1.77%	2.44%	2.33%
Median	0.00%	0.00%	0.88%	0.00%	0.00%
All-2 (120 seq)	k=5	k=4	k=3	k=5	k=4
Average	1.04%	1.75%	**0.57%**	0.77%	1.35%
Median	0.00%	0.00%	**0.00%**	0.00%	0.00%
Checker-3 (26)	k=5	k=4	k=3	k=5	k=4
Average	0.44%	0.43%	0.87%	0.44%	0.43%
Median	0.24%	0.22%	0.35%	0.24%	0.22%
Traffic-3 (7)	k=5	k=4	k=3	k=5	k=4
Average	6.59%	7.18%	1.86%	6.59%	7.18%
Median	1.81%	4.37%	1.53%	1.81%	4.37%
Articulated-3 (2)	k=5	k=4	k=3	k=5	k=4
Average	20.54%	4.05%	5.12%	20.54%	4.05%
Median	20.54%	4.05%	5.12%	20.54%	4.05%
All-3 (35 seq)	k=5	k=4	k=3	k=5	k=4
Average	2.82%	1.98%	**1.31%**	2.82%	1.98%
Median	0.65%	0.47%	**0.45%**	0.65%	0.47%
All (155 seq)	k=5	k=4	k=3	k=5	k=4
Average	1.50%	1.81%	**0.76%**	1.30%	1.50%
Median	0.21%	0.00%	**0.20%**	0.21%	0.00%

Table 5.5: % segmentation errors - NLS algorithm for various k.

Checker-2 (78)	LSA(Original)	LSA(New Dist/Similarity)	LSA(New Segmentation)
Average	2.57%	0.97%	1.71%
Median	0.27%	0.00%	0.00%
Traffic-2 (31)	LSA(Original)	LSA(New Dist/Similarity)	LSA(New Segmentation)
Average	5.43%	1.59%	4.99%
Median	1.48%	1.11%	0.65%
Articulated-2 (11)	LSA(Original)	LSA(New Dist/Similarity)	LSA(New Segmentation)
Average	4.10%	2.10%	4.26%
Median	1.22%	0.43%	1.21%
All-2 (120 seq)	LSA(Original)	LSA(New Dist/Similarity)	LSA(New Segmentation)
Average	3.45%	1.22%	2.27%
Median	0.59%	0.00%	0.35%
Checker-3 (26)	LSA(Original)	LSA(New Dist/Similarity)	LSA(New Segmentation)
Average	5.80%	2.66%	4.67%
Median	1.77%	0.30%	0.91%
Traffic-3 (7)	LSA(Original)	LSA(New Dist/Similarity)	LSA(New Segmentation)
Average	25.07%	6.38%	24.46%
Median	23.79%	1.28%	31.20%
Articulated-3 (2)	LSA(Original)	LSA(New Dist/Similarity)	LSA(New Segmentation)
Average	7.25%	6.18%	7.25%
Median	7.25%	6.18%	7.25%
All-3 (35 seq)	LSA(Original)	LSA(New Dist/Similarity)	LSA(New Segmentation)
Average	9.73%	2.45%	8.78%
Median	2.33%	0.20%	1.94%
All (155 seq)	LSA(Original)	LSA(New Dist/Similarity)	LSA(New Segmentation)
Average	4.94%	1.84%	3.96%
Median	0.90%	0.18%	0.61%

Table 5.6: % segmentation errors for LSA with various parameters.

CHAPTER 6

CONCLUSIONS

6.1 Conclusions

This thesis developed theory and associated algorithms to solve subspace segmentation problem. Given a set of data $\mathbf{W} = \{w_1, ..., w_N\} \in \mathbb{R}^D$ that comes from a union of subspaces, we focused on determining a nonlinear model of the form $\mathscr{U} = \bigcup_{i \in I} S_i$, where $\left\{S_i \subset \mathbb{R}^D\right\}_{i \in I}$ is a set of subspaces, that is nearest to $\mathbf{W}$. The model is then used to classify $\mathbf{W}$ into clusters.

Our first approach is based on the binary reduced row echelon form of data matrix. We prove that, in absence of noise, our approach can find the number of subspaces, their dimensions, and an orthonormal basis for each subspace S_i. We provide a comprehensive analysis of our theory and determine its limitations and strengths in presence of outliers and noise.

Our second approach is based on nearness to local subspaces approach and it can handle noise effectively, but it works only in special cases of the general subspace segmentation problem (i.e., subspaces of equal and known dimensions). Our approach is based on the computation of a binary similarity matrix for the data points. A local subspace is first estimated for each data point. Then, a distance matrix is generated by computing the distances between the local subspaces and points. The distance matrix is converted to the similarity matrix by applying a data-driven threshold. The problem is then transformed to segmentation of subspaces of dimension 1 instead of subspaces of dimension d. The algorithm was applied to the Hopkins 155 Dataset and generated the best results to date.

The binary reduced row echelon based subspace clustering approach solves the general subspace segmentation problem in the absence of noise, but it does not perform well when the data is noisy. The main reason for this is the difficulty of finding an appropriate threshold while constructing the reduced row echelon form of the data matrix $\mathbf{W}$. Such a threshold depends on the

noise level and the relative positioning of the subspaces. Therefore, it should likely be data-driven and can be applied at the each step of construction of the reduced row echelon form $rref(\mathbf{W})$ of $\mathbf{W}$. Although we have applied certain thresholding approaches (e.g. similar to the thresholding approach in Section 5.1.3), this problem may be explored more in the future.

In this research, we considered the ambient space $\mathscr{H}$ to be finite dimensional. There may be situations in which the ambient space is better modeled an infinite dimensional Hilbert space. Such cases can be found in analog signal processing and modeling. This topic has theoretical as well as practical appeal and maybe a subject of future exploration.

BIBLIOGRAPHY

[1] Yue M. Lu and Minh N. Do. "A theory for sampling signals from a union of subspaces". In: *IEEE Transactions on Signal Processing* 56.6 (June 2008). Pp. 2334–2345.

[2] Kenichi Kanatani and Yasuyuki Sugaya. "Multi-stage optimization for multi-body motion segmentation". In: *IEICE Trans. Inf. and Syst*. 2003. Pp. 335–349.

[3] Akram Aldroubi and Kourosh Zaringhalam. "Nonlinear least squares in $\mathbb{R}^n$". In: *Acta Applicandae Mathematicae* 107.1-3 (July 2009). Pp. 325–337.

[4] Rene Vidal, Yi Ma, and Shankar Sastry. *Generalized Principal Component Analysis*. unpublished, 2006.

[5] Kenichi Kanatani. "Motion segmentation by subspace separation and model selection". In: *8th International Conference on Computer Vision*. Vol. 2. 2001. Pp. 301–306.

[6] A. Aldroubi and A. Sekmen. "Reduction and null space algorithms for the subspace clustering problem". In: *arXiv.org* (2010). URL: `http://arxiv.org/abs/1010.2198`.

[7] C. Bregler, A. Hertzmann, and H. Biermann. "Recovering non-rigid 3D shape from image streams". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. 2000.

[8] M. Brand. "Morphable 3D models from video". In: *The IEEE Conference on Computer Vision and Pattern Recognition*. Pp. 456–463.

[9] Ronen Basri and David W. Jacobs. "Lambertian reflectance and linear subspaces". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (2 2003). Pp. 218–233. ISSN: 0162-8828.

[10] J. Ho et al. "Clustering appearances of objects under varying illumination conditions". In: *Computer Vision and Pattern Recognition*. 2003. Pp. 11–18.

[11] A. Aldroubi and R. Tessera. "On the existence of optimal unions of subspaces for data modeling and clustering". In: *Foundation of Computational Mathematics* (2011, to appear). arXiv:1008.4811v1.

[12] Akram Aldroubi, Carlos Cabrelli, and Ursula Molter. “Optimal non-linear models for sparsity and sampling”. In: *Journal of Fourier Analysis and Applications* 14.5 (Dec. 2009). Pp. 793–812.

[13] I. Maravic and M. Vetterli. “Sampling and reconstruction of signals with finite rate of innovation in the presence of noise”. In: *IEEE Transactions on Signal Processing* 53 (2005). Pp. 2788–2805.

[14] Emmanuel Candes and Michael Wakin. “An introduction compressive sampling”. In: *IEEE Signal Processing Magazine* 25.2 (Mar. 2008). Pp. 21–30.

[15] Justin Romberg. “Imaging via compressive sampling”. In: *IEEE Signal Processing Magazine* 25.2 (Mar. 2008). Pp. 14–20.

[16] Emmanuel Candes and Justin Romberg. “Sparsity and incoherence in compressive sampling”. In: *Inverse Problems* 23.3 (June 2007). Pp. 969–985.

[17] Yonina C. Eldar and Moshe Mishali. “Robust recovery of signals from a structured union of subspaces”. In: *IEEE Transactions on Information Theory* 55.11 (Nov. 2009). Pp. 5302–5316.

[18] Ehsan Elhamifar and Rene Vidal. “Sparse subspace clustering”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2009. Pp. 2790–2797.

[19] Ehsan Elhamifar and Rene Vidal. “Clustering disjoint subspaces via sparse representation”. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. 2010.

[20] G. Liu et al. “Robust recovery of subspace structures by low-rank representation”. In: *arXiv.org* (2010). URL: `http://arxiv.org/abs/1010.2955`.

[21] J. Wright and Y. Ma. “Dense error correction via l_1 minimization”. In: *IEEE Transactions on Information Theory* 56.7 (2010). Pp. 3540–3560.

[22] G. Liu, Z. Lin, and Y. Yu. “Robust subspace segmentation by low-rank representation”. In: *International Conference on Machine Learning*. 2010. Pp. 663–670.

[23] Paolo Favaro, Rene Vidal, and Avinash Ravichandran. “A closed form solution to robust subspace estimation and clustering”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011.

[24] Rene Vidal, Yi Ma, and Shankar Sastry. “Generalized Principal Component Analysis (GPCA)”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.12 (Dec. 2005). Pp. 1945–1959.

[25] Roberto Tron and Rene Vidal. “A benchmark for the comparison of 3-D motion segmentation algorithms”. In: *Computer Vision and Pattern Recognition*. 2007. Pp. 1–8.

[26] S. Rao et al. “Robust algebraic segmentation of mixed rigid-body and planar motions in two views”. In: *International Journal on Computer Vision* 88.3 (2010). Pp. 425–446.

[27] P. Tseng. “Nearest q-flat to m points”. In: *Journal of Optimization Theory and Applications* 105.1 (2000). Pp. 249–252.

[28] Amit Gruber and Yair Weiss. “Multibody factorization with uncertainty and missing data using the EM algorithm”. In: *International Conference on Computer Vision and Patern Recognition*. 2004. Pp. 707–714.

[29] Laurent Candillier et al. “SSC : Statistical Subspace Clustering”. In: *5mes Journes d’Extraction et Gestion des Connaissances (EGC’2005)*. Paris 2005. Pp. 177–182.

[30] M. Fischler and R. Bolles. “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (June 1981). Pp. 381–395.

[31] Nuno Silva and Joao Costeira. “Subspace segmentation with outliers: a Grassmannian approach to the maximum consensus subspace”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2008.

[32] U. Von. Luxburg. “A tutorial on spectral clustering”. In: *Statistics and Computing* 17 (2007). Pp. 395–416.

[33] Guangliang Chen and Gilad Lerman. “Spectral Curvature Clustering (SCC)”. In: *International Journal of Computer Vision* 81 (2009). Pp. 317–330.

[34] Fabien Lauer and Christoph Schnorr. “Spectral clustering of linear subspaces for motion segmentation”. In: *IEEE International Conference on Computer Vision*. 2009.

[35] Lihi Zelnik-Manor and Pietro Perona. “Self-tuning spectral clustering”. In: *Advances in Neural Information Processing Systems 17*. MIT Press, 2004. Pp. 1601–1608.

[36] Jingyu Yan and Marc Pollefeys. “A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and nondegenerate”. In: *9th European Conference on Computer Vision*. 2006. Pp. 94–106.

[37] Guangliang Chen, Stefan Atev, and Gilad Lerman. “Kernel Spectral Curvature Clustering (KSCC)”. In: *4th international workshop on Dynamical Vision*. 2009.

[38] Guangliang Chen and Gilad Lerman. “Motion segmentation by SCC on the Hopkins 155 Database”. In: *4th international workshop on Dynamical Vision*. 2009.

[39] A. Goh and R. Vidal. “Segmenting motions of different types by unsupervised manifold clustering”. In: *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*. 2007. Pp. 1 –6.

[40] A. Goh and R. Vidal. “Clustering and dimensionality reduction on Riemannian manifolds”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. 2008. Pp. 1 –7.

[41] R. Vidal. “A tutorial on subspace clustering”. In: *IEEE Signal Processing Magazine* (2010).

[42] Qifa Ke and Takeo Kanade. *Robust Subspace Computation Using L1 Norm*. Tech. rep. Carnegie Mellon University, 2003.

[43] C.W. Gear. “Multibody grouping from motion images”. In: *International Journal of Computer Vision* 29.2 (1998). Pp. 133–150.

[44] L. Zelnik Manor, M. Machline, and M. Irani. “Multi-body factorization with uncertainty: revisiting motion consistency”. In: *International Journal of Computer Vision* 68.1 (June 2006). Pp. 27–41.

[45] Joao Costeira and Takeo Kanade. “A multibody factorization method for independently moving objects”. In: *International Journal of Computer Vision* 29.3 (1998). Pp. 159–179.

[46] C.D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial Mathematics, 2000.

[47] G. H. Golub and C. F. Van Loan. *Matrix computations, 3rd Edition*. Johns Hopkins University Press, 1996.

[48] Y. C. Wong. “Differential geometry of Grassmann manifolds”. In: *Proc. Nat. Acad. Scie.* 57 (1967). Pp. 589–594.

[49] John H. Conway, Ronald H. Hardin, and Neil J. A. Sloane. “Packing lines, planes, etc.: Packings in Grassmannian Spaces”. In: *Experimental Mathematics* 5 (1996). Pp. 139–159.

[50] David Forsyth and Jean Ponce. *Computer vision: a modern approach*. Prentice Hall series in artificial intelligence, 2003.

[51] Kenichi Kanatani and Chikara Matsunaga. “Estimating the number of independent motions for multibody motion segmentation”. In: *5th Asian Conference on Computer Vision*. 2002. Pp. 7–9.

[52] R. Latala. “Some estimates of norms of Random Matrices”. In: *Proc. Amer. Math. Soc.* 133 (2005). Pp. 1273–1282.

[53] R. Vershynin. “Introduction to the non-asymptotic analysis of random matrices”. In: *arXiv.org* (2011). URL: `http://arxiv.org/abs/1011:3027v7`.

[54] Qifa Ke and Takeo Kanade. “Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2005. Pp. 592–599.

[55] Daojing Wang, Chao Zhang, and Xuemin Zhao. “Multivariate Laplace Filter: A heavy-tailed model for target tracking”. In: *International Conference on Pattern Recognition*. 2008. Pp. 1–4.

[56] M J Wainwright and E P Simoncelli. “Scale mixtures of Gaussians and the statistics of natural images”. In: vol. 12. 2000. Pp. 855–861.

[57] T. Eltoft, T. Kim, and T. Lee. “On the multivariate Laplace distribution”. In: *IEEE Signal Processing Letter* (May 2006).

[58] R. Marks et al. “Detection in Laplace noise”. In: *IEEE Transactions on Aerospace and Electronic Systems* AES-14.6 (1978). Pp. 866–872.

[59] Fernando De La Torre and Michael J. Black. “A Framework for Robust Subspace Learning”. In: *International Journal of Computer Vision* 54 (2003). P. 2003.

[60] A. Baccini, P. Besse, and A. de Faguerolles. “A L_1-norm PCA and heuristic approach”. In: *the International Conference on Ordinal and Symbolic Data Analysis*. 1996. Pp. 359–368.

[61] J.P. Brooks and J.H. Dula. *TheL_1-norm best fit hyperplane problem*. 2009. URL: `http://www.optimizationonline.org/DBFILE/2009/05/2291.pdf`.

[62] J.P. Brooks, J.H. Dula, and E.L. Boone. “A pure L_1-norm Principle Component Analysis”. In: *Optimization Online*. 2010.

[63] L. Zappella et al. “Enhanced local subspace affinity for feature-based motion segmentation”. In: *Pattern Recognition* 44 (2011). Pp. 454–470.

[64] R. Vidal, R. Tron, and R. Hartley. “Multiframe motion segmentation with missing data using PowerFactorization and GPCA”. In: *International Journal on Computer Vision* 79.1 (2008). Pp. 85–105.

[65] Marek Petrik. “An analysis of Laplacian methods for value function approximation in MDPs”. In: *Proceedings of the 20th international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., 2007. Pp. 2574–2579.

[66] S. Rao et al. “Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.10 (2010). Pp. 1832–1845.