Estimation of Subspace Arrangements with Applications in Modeling and Segmenting Mixed Data

Yi Ma, Allen Y. Yang, Harm Derksen, and Robert Fossum

Coordinated Science Laboratory
1308 West Main Street, Urbana, IL 61801
University of Illinois at Urbana-Champaign

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB NO. 0704-0188*

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE April 2006	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE Estimation of Subspace Arrangements with Applications in Modeling and Segmenting Mixed Data	5. FUNDING NUMBERS NSF IIS 03-47456 Career
6. AUTHOR(S) Yi Ma, Allen Y. Yang, Harm Derksen, and Robert Fossum	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Coordinated Science Laboratory University of Illinois at Urbana-Champaign 1308 West Main Street Urbana, Illinois 61801-2307	8. PERFORMING RGANIZATION REPORT NUMBER UILU-ENG-06-2202 DC-221

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Science Foundation 4201 Wilson Blvd. Arlington, VA, 22230	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official position, policy, or decision, unless so designated by other documentation

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

In recent years, subspace arrangements have become an increasingly popular class of mathematical objects to be used for modeling a multivariate mixed data set that is (approximately) piecewise linear. A subspace arrangement is a union of multiple subspaces. Each subspace can be conveniently used to model a homogeneous subset of the data. Hence, all the subspaces together can capture the heterogeneous structures within the data set. In this paper, we give a comprehensive introduction to one new approach for the estimation of subspace arrangements, known as generalized principal component analysis. We provide a comprehensive summary of important algebraic properties and statistical facts that are crucial for making the inference of subspace arrangements both efficient and robust, even when the given data are corrupted with noise or contaminated by outliers. This new method in many ways improves and generalizes extant methods for modeling or clustering mixed data. There have been successful applications of this new method to many real-world problems in computer vision, image processing, and system identification. In this paper, we will examine a couple of those representative applications.

14. SUBJECT TERMS	15. NUMBER OF PAGES 48
Subspace arrangement, Hilbert function, generalized principal component analysis, model selection, outlier detection.	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

ESTIMATION OF SUBSPACE ARRANGEMENTS WITH APPLICATIONS IN MODELING AND SEGMENTING MIXED DATA[*]

YI MA[†], ALLEN Y. YANG[‡], HARM DERKSEN[§], AND ROBERT FOSSUM[¶]

Abstract. In recent years, subspace arrangements have become an increasingly popular class of mathematical objects to be used for modeling a multivariate mixed data set that is (approximately) piecewise linear. A subspace arrangement is a union of multiple subspaces. Each subspace can be conveniently used to model a homogeneous subset of the data. Hence, all the subspaces together can capture the heterogeneous structures within the data set. In this paper, we give a comprehensive introduction to one new approach for the estimation of subspace arrangements, also known as *generalized principal component analysis*. In particular, we provide a comprehensive summary of important algebraic properties and statistical facts that are crucial for making the inference of subspace arrangements both efficient and robust, even when the given data are corrupted with noise or contaminated by outliers. This new method in many ways improves and generalizes extant methods for modeling or clustering mixed data. There have been successful applications of this new method to many real-world problems in computer vision, image processing, and system identification. In this paper, we will examine a couple of those representative applications. Although this paper is intended to be expository, we take liberty in filling in many gaps between the theory and the practice in the existing literature so that this paper will be a more complete reference for both theoreticians and practitioners.

Key words. Subspace arrangement, Hilbert function, generalized principal component analysis, model selection, minimum effective dimension, outlier detection.

AMS subject classifications. 52C35, 62H30, 68T45, 62H35.

1. Introduction. In scientific and engineering studies, one of the most common tasks is to find a parametric model for a given set of data. Depending on the purpose and the nature of the data, the model can either be a probabilistic distribution (e.g., a Gaussian distribution, a hidden Markov chain) or a geometric structure (e.g., a line, a curve, or a manifold). Nevertheless, among all the models, linear models such as a straight line and a subspace are possibly the most popular choice mainly because they are simple to understand and easy to represent and compute. Very often in the practice of data modeling, however, a given data set is not homogeneous and cannot be described well by a single linear model. This is especially the case of imagery data. For instance, a natural image typically contains multiple regions, which are significantly different in the complexity of texture. While it is generally true that each region can be modeled well by a simple linear model, the same model unlikely applies to other regions. It is therefore reasonable to use multiple models to describe different regions of the image.

The above example about images reveals a challenging problem that permeates many research areas such as image processing, computer vision, pattern recognition,

[†]Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W Main St., Urbana, Illinois, 61801 (`yima@uiuc.edu`).

[‡]Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W Main St., Urbana, Illinois, 61801 (`yangyang@uiuc.edu`).

[§]Department of Mathematics, University of Michigan, 530 Church St., Ann Arbor, Michigan 48109 (`hderksen@umich.edu`).

[¶]Department of Mathematics, University of Illinois at Urbana-Champaign, 1409 W. Green St., Urbana, Illinois, 61801 (`rmfossum@uiuc.edu`).
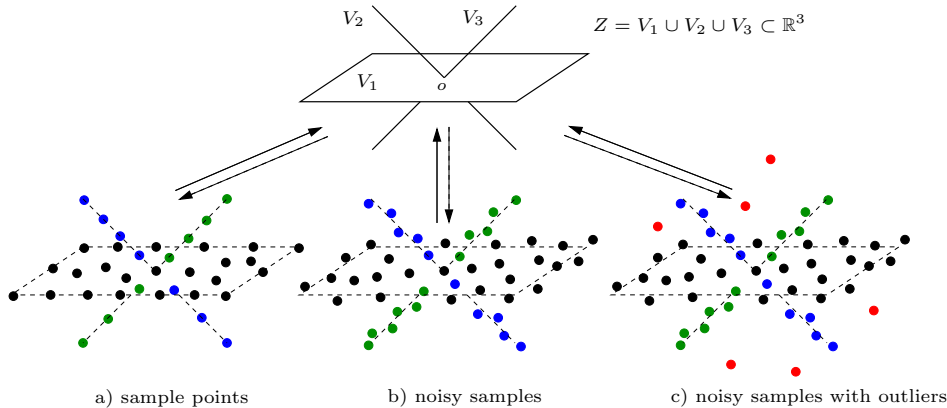
FIG. 1.1. *Inferring a hybrid linear model Z, consisting of one plane (V_1) and two lines (V_2, V_3), from a set of mixed data, which can be: a) noiseless samples from the plane and lines; b) noisy samples; c) noisy samples with outliers.*

and system identification: *How to segment a given set of data into multiple subsets and find the best model for each subset?* In different contexts, such a data set, as well as the associated model, has been called "mixed," or "multi-modal," or "multi-model," or "piecewise," or "heterogeneous," or "hybrid." For simplicity, in this paper, we refer to the data as "mixed" and the model as "hybrid." We are particularly interested in the *hybrid linear model*: one linear model for every homogeneous subset of the data. Figure 1.1 shows a simple example. The importance of hybrid linear models is multi-fold: 1. They are the natural generalizations to single linear models; 2. They are sufficiently expressive for representing or approximating arbitrary complex data structures; 3. The understanding of hybrid linear models has been significantly advanced in recent years and many efficient solutions have been developed. The goal of this paper is to give a comprehensive introduction to some of these new developments, and to put many sporadic results in the literature in a coherent and complete mathematical and computational framework.

A fundamental challenge in estimating such a hybrid model for mixed data is the "chicken-and-egg" problem. If the data were already segmented properly into homogeneous subsets, estimating a model for each subset would be easy. Or, if the hybrid model were already known, segmenting the data into multiple subsets would be straightforward. For instance, in Figure 1.1, if a correct segmentation is given, finding an optimal linear subspace for each subset of sample points has a well-established solution known as *principal component analysis* (PCA) [25]. Or, given the three linear subspaces, one can easily segment the samples to their closest subspaces, respectively. The problem becomes much more involved if neither the model nor the segmentation is known *a priori* and we have only the unsegmented sample points, which sometimes are also corrupted with noise and outliers (see Figure 1.1 b) and c)). So at the heart of modeling such mixed data is the question of how to resolve effectively the coupling between data segmentation and model estimation.

In statistical learning, mixed data are typically modeled as a set of independent samples $\{z_1, z_2, \ldots, z_N\} \subset \mathbb{R}^D$ drawn from a mixture of probabilistic distributions $\{p(z, \theta_j)\}_{j=1}^n$, which is typically a weighted sum $p(z, \Theta) = \sum_j \pi_j p(z, \theta_j)$ with $\sum_j \pi_j = 1$. Then the problem of segmenting mixed data is often converted to a statistical model-estimation problem. Depending on the purpose, the estimated model

parameters can take either the maximum-likelihood estimate, which maximizes the log-likelihood: $\max_{\Theta,\pi} \sum_i \log \left(\sum_j \pi_j p(\boldsymbol{z}_i, \theta_j) \right)$, or the minimax estimate, which optimizes the objective: $\min_{\Theta} \sum_i [\min_j(-\log p(\boldsymbol{z}_i, \theta_j))]$. However, even for simple distributions such as Gaussian distributions, there is no simple closed-form solution to the estimate. One needs to resort to iterative schemes to find the optimal estimate. For the maximum-likelihood estimate, one can view the event that a sample is drawn from the jth distribution as a hidden random variable and it has an expectation of π_j. Then the classical expectation-maximization (EM) algorithm [11,34] can be called upon to maximize the likelihood in a "hill-climbing" fashion. The algorithm iterates between estimating the membership of the samples with the model parameters fixed (the expectation step) and estimating the model with the membership of the samples fixed (the maximization step). The minimax estimate leads to an iterative algorithm, known as the K-means algorithm [16, 24, 30, 32] (or its variation for subspaces, K-Subspaces [20]), which in many aspects resembles the EM algorithm.[1] In a sense, both the EM algorithm and the K-means algorithm have reinforced the belief that the "chicken-and-egg" coupling between model estimation and data segmentation can only be dealt with through such an iteration between the two.

Iterative statistical methods have several drawbacks that limit their applicability in estimating hybrid models. First, the log-likelihood method typically has multiple extrema. If the algorithm is not properly initialized, the iterative process may converge to a local extremum that gives an invalid estimate of the model. In practice, to increase the chance of finding the global extremum, one often needs to run the algorithm multiple times with random initialization, which obviously reduces the efficiency of the algorithm. Second, the statistical formulation typically relies on explicit assumptions about the mixture distribution: the number of component distributions, the parametric models of the distributions, and the dimension or complexity of each model, etc. However, in many practical applications, such information is not readily available and needs to be inferred from the given data. Finally, statistical methods such as maximum likelihood are known to be less effective when dealing with situations in which the distributions are degenerate [51]. These situations arise very often for mixed data with a hybrid model.

Thus, there is a need for alternative methods for mixed data modeling that may remedy the limitations mentioned above. More particularly, we are interested in a non-iterative method not requiring initialization. Although it is not known whether a general solution exists for arbitrary hybrid models, many effective methods have been developed in the past few years for the class of hybrid linear models. The goal of this paper is to provide a comprehensive review of some of these methods. However, to make this review more rigorous and complete, we also take the liberty of filling in some gaps between the theory and the practice. Thus, although this paper is mainly expository, many results presented here are actually new.

1.1. Problem Statement. More precisely, this paper addresses the following problem.

PROBLEM 1.1. *Given a set of sufficiently dense sample points drawn from a union of n linear subspaces $V_1, V_2, \ldots, V_n$ of dimensions $d_1, d_2, \ldots, d_n$, respectively, in a D-dimensional space $\mathbb{F}^D$, where the base field $\mathbb{F}$ is typically $\mathbb{R}$ or $\mathbb{C}$, estimate a basis for each subspace, and segment all sample points into their respective subspaces.*

---

[1]The only difference is that in the expectation step, instead of estimate the probability of each sample belonging to each model, the sample is assigned to the most probable model.

We consider the problem under three assumptions with increasing practicality and difficulty:

Assumption 1: The samples are noiseless samples from the subspaces, see Figure 1.1 a);

Assumption 2: The samples are corrupted by (typically Gaussian) noise, see Figure 1.1 b);

Assumption 3: The samples are corrupted by both noise and outliers, see Figure 1.1 c).

In what follows we develop the solution under the above assumptions. We will also consider situations in which the number of subspaces or their dimensions is either known or unknown.

The technical conditions under which a set of sample points is considered to be "sufficiently dense" will become clear in the context. Furthermore, there is no loss of generality in assuming the affine subsets are linear subspaces. If the affine subspaces do not contain the origin, we can always increase the dimension of the ambient space by one and identify each affine space with the linear subspace that it spans.

In mathematics, a union of multiple subspaces is called a *subspace arrangement*. Subspace arrangements, and their topological complements, are very important classes of objects that have been studied in mathematics for centuries. The importance as well as the difficulty of studying subspace arrangements can hardly be exaggerated. Different aspects of their properties have been and are still being investigated and exploited in many mathematical fields, including algebraic geometry, algebraic topology, combinatorics and complexity theory, and graph and lattice theory, etc. See [6, 7, 35] for a general review.

In the context of modeling mixed data, subspace arrangements are of immediate interest because they are the natural generalizations of single subspaces—the linear models. As an underlying model for mixed data, subspace arrangements are sufficiently flexible and expressive: They may contain subspaces of different dimensions. They can approximate with arbitrary accuracy any nonlinear geometric or topological structures. In addition, as we will see shortly, a subspace arrangement as an algebraic sets can be effectively estimated and segmented from a set of samples in a non-iterative way.

1.2. Organization of This Paper. In this paper we review the solutions to Problem 1.1 under each of the three assumptions. As a result, the scope of subjects to be covered are rather broad: They range from theory to practice, from algebra to statistics, and from simulations to real-world applications. Nevertheless, we hope to convince the reader that these subjects are strongly related to each other and are crucial for investigators who want to gain a deep and complete understanding about the problem.

If the sample points are noiseless, the problem is mostly an algebraic problem. Section 2 reviews the basic algebraic properties of subspace arrangements. As an algebraic set, the set of polynomials that vanish on a subspace arrangement form an ideal and the subspace arrangement is uniquely determined by this ideal. We give a complete characterization of the dimension of each graded component of the ideal, also known as the Hilbert function. We show how the dimensions of the subspaces can be uniquely determined from the values of the Hilbert function. We further show how the vanishing ideal can be determined from a sufficiently dense (nevertheless finite) set of sample points on the arrangement; and how the subspaces can be subsequently de-
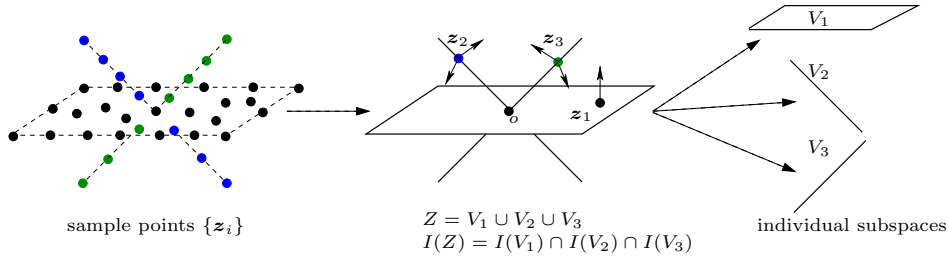
FIG. 2.1. *Inferring a subspace arrangement of three subspaces: $Z = V_1 \cup V_2 \cup V_3$ from a set of sample points $\{z_i\}$.*

duced from the vanishing polynomials. This process is known as *generalized principal component analysis* (GPCA).

In Section 3, we review some statistical techniques that allow us to estimate the vanishing polynomials and the subspaces from sample points that are corrupted by noise. When the number of subspaces and their dimensions are not known, we introduce some relevant model-selection criteria for subspace arrangements that strike a good balance between the complexity of the chosen model and the fidelity of the data (w.r.t. the model). In Section 4, we study the problem under the assumption that the given sample points are contaminated with outliers. We introduce certain robust statistical techniques that can detect or diminish the effect of outliers, especially for subspace arrangements. Finally, we demonstrate how these methods can be applied to several real-world applications in Section 5.[2]

2. Inference of Subspace Arrangements via Algebraic Techniques. Before we can introduce subspace arrangements as a useful class of models for data modeling and segmentation, we need to understand their properties as an important class of algebraic sets. In this section we review the necessary mathematical facts that allow us to infer a subspace arrangement from a finite number of samples and subsequently to decompose the arrangement into separate subspaces, as shown in Figure 2.1. The algebraic facts presented in this section serve as the theoretical foundation for an effective method to model and segment mixed data known as *generalized principal component analysis* (GPCA). In the sections that follow this one, we will show how this algebraic method should be modified when the samples are corrupted by noise or contaminated with outliers.

2.1. Basic Notation and Definitions. In this section, we assume that the reader has basic knowledge of abstract algebra that is covered in any graduate-level algebra course. Details may be found in most texts, for example [13, 29]. In what follows, the ambient space is a D-dimensional vector space over an infinite field $\mathbb{F}$ (which is usually either $\mathbb{R}$ or $\mathbb{C}$). We immediately identify our vector space with $\mathbb{F}^D$. If V is a d-dimensional subspace, then its **codimension** is $c \doteq D - d$.

DEFINITION 2.1 (Subspace Arrangement). *A **subspace arrangement** in $\mathbb{F}^D$ is a union*

$$\mathcal{A} \doteq V_1 \cup V_2 \cup \cdots \cup V_n \tag{2.1}$$

of n subspaces $V_1, V_2, \ldots, V_n$ of $\mathbb{F}^D$.

---

[2]The source code for all the algorithms, as well as more applications, are available on-line at http://perception.csl.uiuc.edu/gpca/Hidden/sample_code.

For a nonempty subset S of the index set $\{1, 2, \ldots, n\}$, we define the intersection

$$V_S \doteq \cap_{s \in S} V_s$$

with dimension $d_S \doteq \dim V_S$ and codimension $c_S \doteq D - d_S$.

DEFINITION 2.2 (Transversal Subspace Arrangement). *A subspace arrangement*

$$\mathcal{A} = V_1 \cup V_2 \cup \ldots \cup V_n$$

is called ***transversal*** *if*

$$c_S = \min \left(D, \sum_{i \in S} c_i \right) \quad \text{for all nonempty} \quad S \subseteq \{1, 2, \ldots, n\}.$$

That is, the dimensions of all intersections are as small as possible.

Notice that transversality is a weaker condition than the typical notion of general position. For instance, three coplanar lines through the origin are transversal in $\mathbb{R}^3$, but usually they are not regarded to be in general position. Transversality is an appropriate assumption for most real applications. Moderate data noise and machine round off should guarantee that the subspace structures of the data are transversal.

The ring of polynomial functions on our ambient space $\mathbb{F}^D$ is denoted by

$$\mathbb{F}^{[D]} \doteq \mathbb{F}[X_1, X_2, \ldots, X_D].$$

It is the ring of polynomials in the functions $\{X_1, X_2, \ldots, X_D\}$, where X_j is the function that assigns the jth coordinate to a point in $\mathbb{F}^D$. Any polynomial $f \in \mathbb{F}^{[D]}$ can be written as a unique sum

$$f = f_0 + f_1 + \cdots + f_T,$$

where the f_i are homogeneous polynomials of degree i. Let $\mathbb{F}_h^{[D]}$ denote the vector space of all polynomials homogeneous of degree h. Then there is a decomposition

$$\mathbb{F}^{[D]} = \mathbb{F} \oplus \mathbb{F}_1^{[D]} \oplus \mathbb{F}_2^{[D]} \oplus \cdots \tag{2.2}$$

of $\mathbb{F}^{[D]}$ into the direct sum of its homogeneous components. Clearly $\mathbb{F}_h^{[D]} \mathbb{F}_k^{[D]} \subseteq \mathbb{F}_{h+k}^{[D]}$.

Each homogeneous component $\mathbb{F}_h^{[D]}$ is a finite dimensional vector space over $\mathbb{F}$ of dimension

$$M_h^{[D]} \doteq \binom{h + D - 1}{D - 1}. \tag{2.3}$$

One can verify this by observing that the monomials

$$\{X_1^h, X_1^{h-1} X_2, X_1^{h-1} X_3, \ldots, X_D^h\}$$

form a basis of $\mathbb{F}_h^{[D]}$.

We end this subsection with one more definition.

DEFINITION 2.3 (Veronese Map). *The* ***Veronese Map*** *of order h is the map*

$$\nu_h : \mathbb{F}^D \to \mathbb{F}^{M_h^{[D]}}$$

given by

$$\nu_h \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{pmatrix} = \begin{pmatrix} x_1^h \\ x_1^{h-1} x_2 \\ \vdots \\ x_D^h \end{pmatrix}. \tag{2.4}$$

2.2. Vanishing Ideals and Hilbert Functions of Subspace Arrangements.
We will discuss the correspondence between ideals in the polynomial ring $\mathbb{F}^{[D]}$ and
subsets in $\mathbb{F}^D$.

DEFINITION 2.4 (Vanishing Ideal). *The **vanishing ideal** $I(W)$ of a subset $W \subseteq$
$\mathbb{F}^D$ is defined by*

$$I(W) \doteq \{ f \in \mathbb{F}^{[D]} : f(\boldsymbol{z}) = 0, \forall \boldsymbol{z} \in W \}.$$

One easily checks that $I(W)$ is indeed an ideal of the polynomial ring $\mathbb{F}^{[D]}$. Before
dealing with a general subspace arrangement, consider first the situation of a single
subspace V. The homogeneous component $\mathbb{F}_1^{[D]}$ is the vector space of linear functions
from $\mathbb{F}^D$ to $\mathbb{F}$. Denote by $V^\perp$ those linear functions on $\mathbb{F}^D$ that vanish on V. Any
linear function that vanishes on V can be written as:

$$f(\boldsymbol{X}) = b_1 X_1 + b_2 X_2 + \cdots + b_D X_D,$$

where $\boldsymbol{b} = (b_1, b_2, \ldots, b_D)^T \in \mathbb{F}^D$ is a vector that satisfies

$$b_1 x_1 + b_2 x_2 + \cdots + b_D x_D = 0 \quad \text{for all} \quad (x_1, x_2, \ldots, x_D)^T \in V. \qquad (2.5)$$

One can show that if the dimension of V is d, then $V^\perp$ has dimension $c = D - d$.
That is, $V^\perp$ is spanned by c linearly independent linear functions

$$V^\perp = \text{span}\{ g_1, g_2, \ldots, g_c \}, \qquad (2.6)$$

where each $g_i \in \mathbb{F}_1^{[D]}$.

All the ideals that we work with turn out to be homogeneous.

DEFINITION 2.5 (Homogeneous Ideal). *An ideal I in $\mathbb{F}^{[D]}$ is **homogeneous** if
the homogeneous components of elements in I are also in I.*

It is well-known, and easy to see, that an ideal is homogeneous if and only if it is
generated by homogeneous elements. The vanishing ideal $I(V)$ of a subspace $V \subseteq \mathbb{F}^D$
is obviously generated by the linear functions in $V^\perp$, in fact by a basis of $V^\perp$, and
hence is a homogeneous ideal generated by finitely many homogeneous elements.

It is easy to see that the vanishing ideal $I(\mathcal{A})$ of a subspace arrangement $\mathcal{A}$ is the
intersection of the vanishing ideals of the individual subspaces:

$$I(\mathcal{A}) = I(V_1 \cup V_2 \cup \cdots \cup V_n) = I(V_1) \cap I(V_2) \cap \cdots \cap I(V_n). \qquad (2.7)$$

Since each of the constituents is homogeneous, the ideal $I(\mathcal{A})$ itself is homogeneous
and hence

$$I(\mathcal{A}) = I_0 \oplus I_1 \oplus I_2 \oplus \cdots$$

where $I_h = I(\mathcal{A}) \cap \mathbb{F}_h^{[D]}$ is the homogeneous part of degree h (for small h this may
be the trivial vector space). Let m be the smallest nonnegative integer such that
$I_m \neq \{0\}$. Then $m \leq n$ and we can write

$$I(\mathcal{A}) = I_m \oplus I_{m+1} \oplus \cdots \oplus I_n \oplus I_{n+1} \oplus \cdots . \qquad (2.8)$$

Notice that polynomials that vanish on $\mathcal{A}$ may have degree strictly lower than
n, the number of subspaces in the arrangement. One example is a transversal ar-
rangement of two lines and one plane in $\mathbb{R}^3$. Since any two lines lie on a plane, this

arrangement can be embedded in a hyperplane arrangement of two planes, and there exist homogeneous polynomials of degree two that vanish on the arrangement.

Let us introduce an ideal related to the vanishing ideal $I(\mathcal{A})$, called the product ideal $J(\mathcal{A}) = I(V_1)I(V_2)\cdots I(V_n)$. That is, $J(\mathcal{A})$ is the ideal generated by the products $g_1 g_2 \cdots g_n$ where $g_j \in I(V_j)$ for each j. The ideal $J(\mathcal{A})$ is also homogeneous. So

$$J(\mathcal{A}) = J_n \oplus J_{n+1} \oplus \cdots. \tag{2.9}$$

It is clear the the first non-zero graded component of $J(\mathcal{A})$ is J_n and that

$$J_n = V_1^\perp V_2^\perp \cdots V_n^\perp = I_1(V_1)I_1(V_2)\cdots I_1(V_n). \tag{2.10}$$

DEFINITION 2.6 (Zero Set). *Given a set of polynomials $I \subseteq \mathbb{F}^{[D]}$, the **zero set** of I is defined to be*

$$Z(I) \doteq \{ \boldsymbol{z} \in \mathbb{F}^D : g(\boldsymbol{z}) = 0 \text{ for all } g \in I \} \subseteq \mathbb{F}^D.$$

LEMMA 2.7. *The subspace arrangement $\mathcal{A}$ is the zero set of the homogeneous component I_n and also the zero set of the homogeneous component J_n. That is,*

$$Z(I_n) = Z(J_n) = Z(I(\mathcal{A})) = Z(J(\mathcal{A})) = \mathcal{A}.$$

Proof. Since $J_n \subseteq I_n \subset I(\mathcal{A})$ and elements of $I(\mathcal{A})$ vanish on the set $\mathcal{A}$ by definition, we have

$$\mathcal{A} \subseteq Z(I(\mathcal{A})) \subseteq Z(I_n) \subseteq Z(J_n). \tag{2.11}$$

For the other direction, suppose $\boldsymbol{z} \notin \mathcal{A}$. Then $\boldsymbol{z} \notin V_i$ for all $i = 1, 2, \ldots, n$. Hence for each i, there exists a linear function $g_i \in V_i^\perp$ such that $g_i(\boldsymbol{z}) \neq 0$. Let $g = g_1 g_2 \cdots g_n$. Then $g(\boldsymbol{z}) \neq 0$. Obviously $g \in J_n$. It then follows that $\boldsymbol{z} \notin Z(J_n)$. Therefore $Z(J_n) \subseteq \mathcal{A}$. Using (2.11), we obtain

$$\mathcal{A} = Z(I(\mathcal{A})) = Z(I_n) = Z(J_n).$$

Also $Z(J(\mathcal{A})) = Z(J_n) = \mathcal{A}$ because $J(\mathcal{A})$ is generated by J_n. $\square$

A consequence of Lemma 2.7 is that that in order to recover an arrangement $\mathcal{A}$ of n subspaces, one needs only to know the set of polynomials of degree n that vanish on $\mathcal{A}$. As we will soon see: to estimate a subspace arrangement $\mathcal{A}$, it is very useful to know first the number of linearly independent polynomials of degree n that vanish on $\mathcal{A}$. This is related to the **Hilbert function** of the vanishing ideal $I(\mathcal{A})$.

DEFINITION 2.8 (Hilbert Function). *The **Hilbert Function** of a homogeneous ideal K is the function $h_K : \mathbb{N} \to \mathbb{N}$ defined by*

$$h_K(j) \doteq \dim(K_j), \tag{2.12}$$

where K_i is the ith homogeneous component of K and $\mathbb{N}$ denotes the nonnegative integers.[3]

---

[3]Be aware that in the literature, the Hilbert function is sometimes defined as the codimension of K_i in $\mathbb{F}_i^{[D]}$: $M_i^{[D]} - \dim(K_i)$.

The remainder of this section is devoted to providing a closed-form formula for the Hilbert function $h_I(i)$ of subspace arrangements that is valid for $i \geq n$. A more complete development is given in [12].

The ideal $J = J(\mathcal{A})$ has its associated Hilbert function $h_J(i) \doteq \dim(J_i)$. We claim there is a closed-form formula for $h_J(i)$. In addition when the arrangement is transversal, then $h_I(i) = h_J(i)$ for all $i \geq n$.

DEFINITION 2.9 (Hilbert Series). *The **Hilbert Series** of a homogeneous ideal K is defined to be*

$$\mathcal{H}(K, t) \doteq \sum_{i \in \mathbb{N}} h_K(i) t^i. \tag{2.13}$$

EXAMPLE 2.10. The Hilbert series of the polynomial ring $\mathbb{F}^{[D]}$ is

$$\mathcal{H}(\mathbb{F}^{[D]}, t) = \sum_{i \in \mathbb{Z}} \dim(\mathbb{F}_i^{[D]}) t^i = \sum_{i \in \mathbb{Z}} \binom{i+D-1}{D-1} t^i = \frac{1}{(1-t)^D}.$$

EXAMPLE 2.11. Suppose $I(V)$ is the vanishing ideal of a subspace V of dimension d in $\mathbb{F}^D$. Then $I(V)$ is generated by $c = D - d$ linear polynomials $g_1, g_2, \ldots, g_c$. The quotient ring $\mathbb{F}^{[D]}/I(V)$ can be identified with the ring of polynomial functions on $V \cong \mathbb{F}^d$, so $\mathbb{F}^{[D]}/I(V) \cong \mathbb{F}^{[d]}$. Hence,

$$\mathcal{H}(I(V), t) = \mathcal{H}(\mathbb{F}^{[D]}, t) - \mathcal{H}(\mathbb{F}^{[d]}, t) = \frac{1}{(1-t)^D} - \frac{1}{(1-t)^d} = \frac{1 - (1-t)^c}{(1-t)^D}.$$

A recursive formula for the Hilbert series of $J(\mathcal{A})$ was given in [12]. Surprisingly, this formula depends only on the codimensions of the intersections (c_S, $S \subseteq \{1, 2, \ldots, n\}$) and D, the dimension of the ambient vector space. This means that the Hilbert series $\mathcal{H}(J(\mathcal{A}), t)$ is a *combinatorial invariant* of the arrangement $\mathcal{A}$. Combinatorial invariants play an important role in the study of subspace arrangements and hyperplane arrangements. In general, the Hilbert series of $I(\mathcal{A})$ is *not* a combinatorial invariant. This means the series $\mathcal{H}(I(\mathcal{A}), t)$ depends more delicately on the geometry of the arrangement. For example, suppose that $\mathcal{A}$ is the union of three distinct lines (through the origin) in $\mathbb{F}^3$. Regardless whether the three lines are coplanar or not, we have

$$\mathcal{H}(J(\mathcal{A}), t) = \frac{7t^3 - 9t^4 + 3t^5}{(1-t)^3} = 7t^3 + 12t^4 + 18t^5 + \cdots$$

(for a derivation of this formula and the formulas below, see [12]). However, one has

$$\mathcal{H}(I(\mathcal{A}), t) = \frac{t + t^3 - t^4}{(1-t)^3} = t + 3t^2 + 7t^3 + 12t^4 + 18t^5 + \cdots$$

if the lines are coplanar, and

$$\mathcal{H}(I(\mathcal{A}), t) = \frac{3t^2 - 2t^3}{(1-t)^3} = 3t^2 + 7t^3 + 12t^4 + 18t^5 + \cdots$$

if the three lines are not coplanar. In these examples, the subspace arrangements are transversal. The situation is still quite nice for transversal arrangements. In that case, the Hilbert series of $J(\mathcal{A})$ has a particularly nice form, which we will describe now.

Suppose

$$\mathcal{A} = V_1 \cup V_2 \cup \cdots \cup V_n$$

is a subspace arrangement and that $(c_1, c_2, \ldots, c_n)$ is the vector of codimensions. Define the power series $F_\mathcal{A}(t)$ by

$$F_\mathcal{A}(t) \doteq \frac{\prod_{i=1}^n (1 - (1-t)^{c_i})}{(1-t)^D}. \tag{2.14}$$

We can decompose $F_\mathcal{A}(t)$ in a unique way as

$$F_\mathcal{A}(t) = P_\mathcal{A}(t) + G_\mathcal{A}(t)$$

where $P_\mathcal{A}(t)$ is a polynomial, and

$$G_\mathcal{A}(t) = \frac{Q_\mathcal{A}(t)}{(1-t)^D} = g(0) + g(1)t + g(2)t^2 + \cdots$$

such that $Q_\mathcal{A}(t)$ is a polynomial of degree $< D$.

THEOREM 2.12 ([12]). *Suppose $\mathcal{A} = V_1 \cup V_2 \cup \cdots \cup V_n$ is a transversal arrangement. Then*

$$h_I(i) = h_J(i) = g(i)$$

for $i \geq n$. In other words, $\mathcal{H}(I(\mathcal{A}), t) - G_\mathcal{A}(t)$ and $\mathcal{H}(J(\mathcal{A}), t) - G_\mathcal{A}(t)$ are polynomials of degree $< n$.

From (2.14) we deduce that

$$G_\mathcal{A}(t) = \sum_S (-1)^{|S|} \frac{1}{(1-t)^{D-c_S}}$$

where $c_S = \sum_{j \in S} c_j$ and the sum is over all $S \subseteq \{1, 2, \ldots, n\}$ for which $c_S < D$.

COROLLARY 2.13. *If $\mathcal{A} = V_1 \cup V_2 \cup \cdots \cup V_n$ is transversal, then for all $i \geq n$,*

$$h_I(i) = h_J(i) = g(i) = \sum_S (-1)^{|S|} \binom{i + D - 1 - c_S}{D - 1 - c_S}, \tag{2.15}$$

where $c_S = \sum_{j \in S} c_j$ and the sum is over all $S \subseteq \{1, 2, \ldots, n\}$ (including the empty set) for which $c_S < D$.

The formula in Corollary 2.13 is not particularly efficient to evaluate: the number of terms may depend exponentially on n. Directly evaluating $F_\mathcal{A}(t)$ and $G_\mathcal{A}(t)$ as quotients of expanded polynomials, and then evaluating the power series of $G_\mathcal{A}(t)$ is a more efficient way to determine the values $g(i)$, $i = n, n+1, n+2, \ldots$.

EXAMPLE 2.14.   Suppose that $\mathcal{A} = V_1 \cup V_2 \cup V_3$ is a transversal arrangement in $\mathbb{F}^4$. Let d_1, d_2, d_3 (respectively c_1, c_2, c_3) be the dimensions (resp. codimensions) of

V_1, V_2, V_3. We make a table of $h_I(n)$ for $n = 3, 4, 5$.

c_1, c_2, c_3	d_1, d_2, d_3	$h_I(3)$	$h_I(4)$	$h_I(5)$
$1, 1, 1$	$3, 3, 3$	1	4	10
$1, 1, 2$	$3, 3, 2$	2	7	16
$1, 1, 3$	$3, 3, 1$	3	9	19
$1, 2, 2$	$3, 3, 2$	4	12	25
$1, 2, 3$	$3, 2, 1$	6	15	29
$1, 3, 3$	$3, 1, 1$	8	18	33
$2, 2, 2$	$2, 2, 2$	8	20	38
$2, 2, 3$	$2, 2, 1$	11	24	43
$2, 3, 3$	$2, 1, 1$	14	28	48
$3, 3, 3$	$1, 1, 1$	17	32	53

Note that the codimensions c_1, c_2, c_3 are almost determined by $h_I(3)$. They are uniquely determined $h_I(3)$ and $h_I(4)$. Corollary 2.15 below is a general result that implies that c_1, c_2, c_3 are determined by $h_I(3), h_I(4), h_I(5)$ in this particular example.

COROLLARY 2.15. *Consider a transversal arrangement of n subspaces. The codimensions $c_1, \ldots, c_n$ (and hence the dimensions) of the subspaces are uniquely determined by the values of the Hilbert function $h_I(i)$ for $i = n, n + 1, \ldots, n + D - 1$.*

We will explain how $c_1, c_2, \ldots, c_n$ can be determined from the values

$$h_I(n), h_I(n+1), \ldots, h_I(n + D - 1).$$

Let us assume that $0 < c_1 \le c_2 \le \cdots \le c_n \le D$. First of all $g(i) = h_I(i)$ for $i = n, n+1, \ldots, n+D-1$. Corollary 2.13 shows us that g is a polynomial function of degree $\le D - 1$. This implies that the values $g(n), g(n+1), \ldots, g(n+D-1)$ uniquely determine g and hence $G_{\mathcal{A}}(t)$. In fact $G_{\mathcal{A}}(t) = Q_{\mathcal{A}}(t)/(1-t)^D$ can be determined as follows. Let us write $Q_{\mathcal{A}}(t) = q_0 + q_1 t + \cdots + q_{D-1} t^{D-1}$ were $q_0, q_1, \ldots, q_{D-1}$ are indeterminates. We expand the power series of $G_{\mathcal{A}}(t)$. Equating the coefficient of t^i to $g(i)$ for $i = n, n+1, \ldots, n+D-1$ gives a system of D linear equations from which $q_0, q_1, \ldots, q_{D-1}$ can be solved.

The polynomials $t^D G_{\mathcal{A}}(1-t)$ and

$$t^D F_{\mathcal{A}}(1-t) = \prod_{i=1}^{n}(1 - t^{c_i})$$

coincide in degree $< D$. From this we can determine the value of c_1, because c_1 is the smallest positive integer such that the coefficient of t^{c_1} in $t^D G_{\mathcal{A}}(1-t)$ is nonzero. The power series of

$$\frac{t^D G_{\mathcal{A}}(1-t)}{1 - t^{c_1}}$$

and the polynomial

$$\prod_{i=2}^{n}(1 - t^{c_i})$$

coincide in degree $< D$. From this follows that c_2 is the smallest positive integer such that the coefficient of t^{c_2} in the power series of

$$\frac{t^D G_{\mathcal{A}}(1-t)}{1 - t^{c_1}}$$

is nonzero. Continuing like this we see that c_i is the smallest positive integer such that the coefficient of t^{c_i} in the power series of

$$\frac{t^D G_{\mathcal{A}}(t)}{\prod_{j=1}^{i-1} (1 - t^{c_j})}$$

is nonzero. This way, we can determinate $c_1, c_2, \ldots, c_n$.

EXAMPLE 2.16. Suppose $\mathcal{A} = V_1 \cup V_2 \cup V_3 \cup V_4$ is a transversal arrangement in $\mathbb{F}^4$. We would like to find the codimensions c_1, c_2, c_3, c_4. Suppose that we know that

$$h_I(4) = 11, h_I(5) = 24, h_I(6) = 43, h_I(7) = 69.$$

Let us write

$$G_{\mathcal{A}}(t) = \frac{q_0 + q_1 t + q_2 t^2 + q_3 t^3}{(1-t)^4}.$$

We expand the power series of $G(t)$:

$$G_{\mathcal{A}}(t) = \cdots + (35q_0 + 20q_1 + 10q_2 + 4q_3)t^4 + (56q_0 + 35q_1 + 20q_2 + 10q_3)t^5 +$$

$$+ (84q_0 + 56q_1 + 35q_2 + 20q_3)t^6 + (120q_0 + 84q_1 + 56q_2 + 35q_3)t^7 + \cdots$$

This should be equal to

$$\cdots + 11t^4 + 24t^5 + 43t^6 + 69t^7 + \cdots.$$

We solve the system of equations

$$\begin{cases} 35q_0 & + & 20q_1 & + & 10q_2 & + & 4q_3 & = & 11 \\ 56q_0 & + & 35q_1 & + & 20q_2 & + & 10q_3 & = & 24 \\ 84q_0 & + & 56q_1 & + & 35q_2 & + & 20q_3 & = & 43 \\ 120q_0 & + & 84q_1 & + & 56q_2 & + & 35q_3 & = & 69. \end{cases}$$

The unique solution is

$$q_0 = -1, q_1 = 2, q_2 = 1, q_3 = -1.$$

This implies that

$$G_{\mathcal{A}}(t) = \frac{-1 + 2t + t^2 - t^3}{(1-t)^4}.$$

We compute

$$t^4 G_{\mathcal{A}}(1-t) = 1 - t - 2t^2 + t^3.$$

The first nonzero coefficient (apart from t^0) is the coefficient of t^1. Therefore, $c_1 = 1$. We compute

$$\frac{t^4 G_{\mathcal{A}}(1-t)}{1-t} = \frac{1 - t - 2t^2 + t^3}{1-t} = 1 - 2t^2 - t^3 + \cdots$$

The first nonzero coefficient is the coefficient of t^2. Therefore, $c_2 = 2$. (In fact the coefficient -2 implies that 2 subspaces have codimension 2.) We divide by $1 - t^2$:

$$\frac{t^4 G_{\mathcal{A}}(1-t)}{(1-t)(1-t^2)} = \frac{1 - 2t^2 - t^3 + \cdots}{1-t^2} = 1 - t^2 - t^3 + \cdots$$

Again, the coefficient of t^2 is nonzero, so $c_3 = 2$ as well. We divide by $1 - t^2$ again:

$$\frac{t^4 G_{\mathcal{A}}(1-t)}{(1-t)(1-t^2)^2} = \frac{1 - 2t^2 - t^3 + \cdots}{1-t^2} = 1 - t^3 + \cdots$$

The first nonzero coefficient is the coefficient of t^3. Therefore, $c_4 = 3$.

These results are very important for the development and improvement of the algorithm for estimating a subspace arrangement from a data set. This algorithm is known as generalized principal component analysis (GPCA):

1. The equality $h_I(i) = h_J(i)$ for $i \geq n$ implies that $I_i = J_i$ for $i \geq n$ and in particular $I_n = J_n$. That is, the homogeneous component I_n of the vanishing ideal of a transversal subspace arrangement is always generated by products of linear forms. (This is called *pl-generated* in [7].). This fact was used (but not established at the time) in the early development of the GPCA algorithm because the algorithm would be much easier to explain (to engineers) by using products of linear forms.

2. The values of the Hilbert function give a rich class of invariants for subspace arrangements. Knowing those values may greatly facilitate the task of finding the correct subspace arrangement model for a given set of (noisy) data. On the one hand, given a data set, if we know the number of subspaces and their dimensions, the value of the Hilbert function will tell us exactly how many linearly independent polynomials of a certain degree to use to fit the data set. This information becomes particularly important when the data are noisy and the number of fitting polynomials is difficult to be determined from the data themselves. On the other hand, if the dimensions (or number) of the subspaces are not given but we are able to obtain the set of vanishing polynomials (up to certain degree), then the dimensions (or number) of the subspaces can be uniquely determined from the values of the Hilbert function (even without segmenting the data first).

The reader needs to be aware that formula (2.15) for $h_I(i)$ is valid only for $i \geq n$. For $i < n$ there is no known closed-form formula for $h_I(i)$. One must resort to symbolic or numerical computation to find those values. Fortunately, for most practical applications that we have seen so far, it is typically good enough to know the values of $h_I(i)$ for fewer than 10 subspaces in an ambient space of dimension less than 15.[4]

---

[4]Source codes of both symbolic and numerical computation are available from the authors. We have also computed the table of values of $h_I(i)$ for up to six subspaces in $\mathbb{R}^{12}$.

2.3. Sampling of an Algebraic Set. Before we introduce an algebraic algorithm to estimate an arrangement of N subspaces, we want to prove that in theory one can always recover an algebraic set Z from a finite number of sample points. This is the goal of this section.

From an algebraic point of view, an algebraic set Z, such as a subspace arrangement, is rather different from a finite number of discrete sample points on Z. In fact, suppose $\mathfrak{m}_i$ is the set of polynomials that vanish on just one point z_i, then the set of polynomials that vanish on the finite set $F = \{z_1, z_2, \ldots, z_N\}$ is the intersection

$$I(F) = \mathfrak{m}_1 \cap \mathfrak{m}_2 \cap \cdots \cap \mathfrak{m}_N. \tag{2.16}$$

It should be noted that the quotient $\mathbb{F}^{[D]}/I(F)$ is always a finite dimensional vector space over $\mathbb{F}$, regardless of the number of points N.

We want to find the ideal $I(Z)$ by sampling a finite set F. In general, the ideal $I(Z)$ is always a proper subideal of $I(F)$, regardless of how many points one samples. However, the information about $I(Z)$ can still be retrieved from $I(F)$ as we show in the theorem below. A further bit of notation is required. For the graded ring $\mathbb{F}^{[D]}$, let

$$\mathbb{F}^{[D]}_{\leq n} = \mathbb{F} \oplus \mathbb{F}^{[D]}_1 \oplus \cdots \oplus \mathbb{F}^{[D]}_n. \tag{2.17}$$

It is important to note that this is a finite dimensional vector space.

THEOREM 2.17 (Sampling of an Algebraic Set). *Consider a nonempty set $Z \subseteq \mathbb{F}^D$ whose vanishing ideal $I(Z)$ is generated by polynomials in $\mathbb{F}^{[D]}_{\leq n}$. Then there is a finite sequence $F = \{z_1, z_2, \ldots, z_N\}$ such that $I(F) \cap \mathbb{F}^{[D]}_{\leq n}$ generates $I(Z)$.*

Proof. Let $I_n = I(Z) \cap \mathbb{F}^{[D]}_{\leq n}$. This vector space generates $I(Z)$. Let $\mathfrak{a}_0 = \mathbb{F}^{[D]} = I(\emptyset)$. Let $\mathfrak{b}_0 = \mathfrak{a}_0 \cap \mathbb{F}^{[D]}_{\leq n}$ and let $A_0 = (\mathfrak{b}_0)$, the ideal generated by the polynomials in $\mathfrak{a}_0$ of degree less than or equal to n. Since $1 \in \mathbb{F}^{[D]} \cap \mathbb{F}^{[D]}_{\leq n}$ is the generator of this ideal, we have $A_0 = \mathbb{F}^{[D]}$. Since $Z \neq \emptyset$, then $A_0 \neq I(Z)$. Set $N = 1$ and pick a point $z_1 \in Z$. Then $1(z_1) \neq 0$ (1 is the function that assigns 1 to every point of Z.). Let $\mathfrak{a}_1$ be the ideal that vanishes on $\{z_1\}$ and define $\mathfrak{b}_1 = \mathfrak{a}_1 \cap \mathbb{F}^{[D]}_{\leq n}$. Further let $A_1 = (\mathfrak{b}_1)$.[5] Since $I(Z) \subseteq \mathfrak{a}_1$, it follows that $I_n \subseteq \mathfrak{b}_1$. If $A_1 = I(Z)$, then we are done. Suppose then that $I(Z) \subset A_1$.

Let us do the induction at this point. Suppose we have found a finite sequence $F_N = \{z_1, z_2, \ldots, z_N\} \subset Z$ with

$$I(F_N) = \mathfrak{a}_N \tag{2.18}$$

$$\mathfrak{b}_N = \mathfrak{a}_N \cap \mathbb{F}^{[D]}_{\leq n} \tag{2.19}$$

$$A_N = (\mathfrak{b}_N) \tag{2.20}$$

$$\mathfrak{b}_0 \supset \mathfrak{b}_1 \supset \cdots \supset \mathfrak{b}_N \supseteq I_n. \tag{2.21}$$

It follows that $I_n \subseteq \mathfrak{b}_N$ and that $I(Z) \subseteq A_N$. If equality holds here, then we are done. If not, then there is a function $g \in \mathfrak{b}_N$ not in $I(Z)$ and an element $z_{N+1} \in Z$ for which $g(z_{N+1}) \neq 0$. Set $F_{N+1} = \{z_1, \ldots, z_N, z_{N+1}\}$. Then one gets $\mathfrak{a}_{N+1}, \mathfrak{b}_{N+1}, A_{N+1}$ as above with

$$\mathfrak{b}_0 \supset \mathfrak{b}_1 \supset \cdots \supset \mathfrak{b}_N \supset \mathfrak{b}_{N+1} \supseteq I_n. \tag{2.22}$$

---

[5] Here we are using the convention that (S) is the ideal generated by the set S. Recall also that the ring $\mathbb{F}^{[D]}$ is *noetherian* by the Hilbert basis theorem and so all ideals in the ring are finitely generated [13].

We obtain a descending chain of subspaces of the vector space $\mathbb{F}^{[D]}_{\leq n}$. This chain must stabilize, since the vector space is finite dimensional. Hence there is an N for which $\mathfrak{b}_N = I_n$ and we are done. $\square$

We point out that no clear bound on the total number N of points needed is given in the proof above. Nevertheless, from the proof of the theorem it is seen that the set of finite sequences of samples that satisfy the theorem is an open set. This is of great practical importance for it implies that the vanishing ideal can be determined from a randomly chosen sequence of samples with probability one.

EXAMPLE 2.18. [A Hyperplane in $\mathbb{R}^3$] Consider a plane $P = \{\boldsymbol{z} \in \mathbb{R}^3 : f(\boldsymbol{z}) = ax_1 + bx_2 + cx_3 = 0\}$. The polynomial , $f(\boldsymbol{z}) = ax_1 + bx_2 + cx_3$ will be the only (homogeneous) polynomial of degree 1 that fits any two points in general position in P. In terms of the language introduced above, the ideal $I(P) = \left(\mathfrak{a}_2 \cap \mathbb{R}^{[3]}_{\leq 1}\right)$.

2.4. Generalized Principal Component Analysis. In the previous subsections we considered the correspondence between a transversal subspace arrangement $\mathcal{A}$ and its vanishing ideal (Section 2.2). Then we showed how we are able, in principal, to recover the ideal from a large enough number of samples on the arrangement (Section 2.3). In this subsection, we introduce an efficient algebraic algorithm to retrieve a subspace arrangement and its individual subspaces from a given set of samples. This process is known as *generalized principal component analysis* (GPCA). In this section, we assume the samples to be noise-free. We will discuss samples corrupted by noise in Section 3 and samples contaminated by outliers in Section 4.

The first version of the algebraic GPCA algorithm was proposed in [53]. Several different variations have been proposed since then. All variants consist of three steps. First, a set of polynomials that vanish on the given data samples is retrieved. Second, the vectors normal to the subspaces are estimated from the derivatives of these polynomials. Third, the samples are segmented into their respective subspaces based on the normals. We give a brief description of each step.

2.4.1. Retrieving the Vanishing Polynomials. We are given a set of samples $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$ that we know lies in a subspace arrangement. Typically, we are dealing with real data sets. Thus unless otherwise stated, for the rest of the paper, we will assume the field $\mathbb{F}$ to be the real field $\mathbb{R}$. Suppose that we know the number n and the dimensions of the subspaces in the subspace arrangement $\mathcal{A} \subseteq \mathbb{R}^D$. We then know the number of linearly independent vanishing polynomials of degree n is equal to the value of the Hilbert Function of $I(\mathcal{A})$ at n. Suppose $m = h_I(n)$. We then embed the samples in $\mathbb{R}^{M_n^{[D]}}$ via the veronese map ν_n (see Definition 2.3) obtaining the matrix

$$L_n \doteq \left(\nu_n(\boldsymbol{z}_1), \nu_n(\boldsymbol{z}_2), \ldots, \nu_n(\boldsymbol{z}_N)\right) \quad \in \mathbb{R}^{M_n^{[D]} \times N}. \tag{2.23}$$

Obviously, if $q(\boldsymbol{X}) = \boldsymbol{c}^T \nu_n(\boldsymbol{X})$ is a polynomial that vanishes on $\mathcal{A}$, then we have $q(\boldsymbol{z}_i) = \boldsymbol{c}^T \nu_n(\boldsymbol{z}_i) = 0$ for all $i = 1, 2, \ldots, N$. Therefore the column of coefficients $\boldsymbol{c}$ must be in the (left) null space of L_n: $\boldsymbol{c}^T L_n = 0$. If the sample set is large enough, according to the algebraic sampling theorem 2.17, the dimension of the null space of L_n is exactly $m = h_I(n)$. Thus, a basis $\boldsymbol{C} = \left(\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_m\right)$ of the null space of L_n gives a basis of $I_n(\mathcal{A})$:

$$Q(\boldsymbol{X}) \doteq \left(q_1(\boldsymbol{X}), q_2(\boldsymbol{X}), \ldots, q_m(\boldsymbol{X})\right)^T, \tag{2.24}$$

where $q_i(\boldsymbol{X}) = \boldsymbol{c}_i^T \nu_n(\boldsymbol{X}), i = 1, 2, \ldots, m$.

The matrix $\boldsymbol{C}$ can be computed from the eigenvectors of the matrix

$$W \doteq \frac{1}{N} L_n L_n^T \in \mathbb{R}^{M_n^{[D]} \times M_n^{[D]}}$$

that correspond to its m eigenvalues with eigenvalue 0. In the case of small noise or numerical round-off errors, we can take the m eigenvectors associated with the m smallest eigenvalues. Numerically, this can be done via singular value decomposition (SVD) of L_n. Statistically, this the same as principal component analysis (PCA). In the next section, we will see how the estimate of $\boldsymbol{C}$ can be further improved when the samples are noisy.

This will be facilitated by using the Jacobian of the polynomials generating the vanishing ideal.

DEFINITION 2.19 (Jacobian Matrix). *The **Jacobian Matrix** of the polynomials* $Q(\boldsymbol{X}) = \big(q_1(\boldsymbol{X}), q_2(\boldsymbol{X}), \ldots, q_m(\boldsymbol{X})\big)^T$ *is the $m \times D$ matrix*

$$\mathcal{J}(Q)(\boldsymbol{X}) \doteq \begin{pmatrix} \frac{\partial q_1}{\partial X_1} & \cdots & \frac{\partial q_1}{\partial X_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial q_m}{\partial X_1} & \cdots & \frac{\partial q_m}{\partial X_D} \end{pmatrix} \quad \in \mathbb{R}^{m \times D}. \tag{2.25}$$

2.4.2. Retrieving the Normal Vectors and Bases of the Subspaces. Having found the vanishing polynomials $Q(\boldsymbol{X})$, we can, in principle, obtain the subspace arrangement $\mathcal{A}$ as their zero set. In practice, we are more interested in the individual subspaces of the arrangement rather than their union, particularly, we want to segment the data into their respective subspaces. Thus, the problem that arises is how to retrieve the subspaces from the vanishing polynomials. Fortunately, in addition to the polynomials generating the vanishing ideal, we also have sample points from their zero set. This turns out to simplify greatly the identification of the individual consituent subspaces in the arrangement.

Pick one sample $\boldsymbol{z}_i$ per subspace V_i (not in any of the other subspaces).[6] Evaluate the Jacobian matrix $\boldsymbol{z}_i$ and we obtain the $m \times D$ matrix $\mathcal{J}(Q)(\boldsymbol{z}_i)$.

It is easy to verify that the rows of $\mathcal{J}(Q)(\boldsymbol{z}_i)$ span the orthogonal complement $V_i^{\perp}$ of V_i. Figure 2.1 illustrates this concept with a simple example. Thus, a basis of V_i can be computed from the (right) null space of $\mathcal{J}(Q)(\boldsymbol{z}_i)$, say from the SVD of $\mathcal{J}(Q)(\boldsymbol{z}_i)$, in a manner similar to the computation of $\boldsymbol{C}$.

2.4.3. The Algebraic GPCA Algorithm and Its Variations. For future reference, we summarize the above process as an algorithm below.

Algorithm 1 applies to the very idealistic situation in which the samples have no noise and the number and dimensions of the subspaces are all known. If any of those conditions are changed, the algorithm needs to be modified accordingly.

For instance, we know that the lowest degree of the polynomials that vanish on the given data set can be strictly lower than the number of subspaces. If the number of subspaces is not known, the derivatives of these polynomials of the lowest degree lead to a super subspace arrangement $\mathcal{A}'$ that contains the original arrangement: $\mathcal{A} \subseteq \mathcal{A}'$. Thus, we can recursively apply GPCA to samples in each subspace of $\mathcal{A}'$. In principle, the process will stop when all the subspaces in the original arrangement are found. In

---

[6]The literature is full of many proposals for picking such a point when the samples are noisy. In the next section, we will provide a scheme that does not rely on the choice of the point.

Algorithm 1 (Generalized Principal Component Analysis).

Given a set of samples $\{z_1, z_2, \ldots, z_N\}$ from n linear subspaces of dimensions $d_1, d_2, \ldots, d_n$ in $\mathbb{R}^D$:

1: Construct the matrix $L_n = \big(\nu_n(z_1), \nu_n(z_2), \ldots, \nu_n(z_N)\big)$.
2: Compute the singular value decomposition (SVD) of L_n and let $\boldsymbol{C}$ be the singular vectors associated with the $m = h_I(n)$ smallest singular values.
3: Construct the polynomials $Q(\boldsymbol{X}) = \boldsymbol{C}^T \nu_n(\boldsymbol{X})$.
4: **for all** $1 \leq i \leq n$ **do**
5:    Pick one point z_i per subspace, and compute the Jacobian $\mathcal{J}(Q)(z_i)$.
6:    Compute a basis $B_i = \big(b_1, b_2, \ldots, b_{d_i}\big)$ of V_i from the right null space of $\mathcal{J}(Q)(z_i)$ via the singular value decomposition of $\mathcal{J}(Q)(z_i)$.
7:    Assign samples z_j that satisfy $B_i^T z_j = 0$ to the subspace V_i.
8: **end for**

---

the literature, this is known as Recursive GPCA. However, if the samples are noisy, the stopping criterion becomes much more elusive.

There are many more variations to the GPCA algorithm when the samples are corrupted by noise or contaminated with outliers. We will discuss some of the important variations in the next two sections.

3. Estimation of Subspace Arrangements from Noisy Samples. When the samples from a subspace arrangement are corrupted by noise, estimating the vanishing polynomials and subsequently retrieving the subspaces become a statistical problem. In this case, the embedded data matrix will be of full rank and the vanishing polynomials can no longer be retrieved directly from its null space. We discuss how to estimate the vanishing polynomials from noisy samples in Section 3.1, which is inspired by the work of [45] with special treatment given to homogeneous polynomials. Likewise, the derivatives of the vanishing polynomials at a noisy sample point no longer span the orthogonal complement to the underlying subspace. Thus, neither the dimension nor the basis of the subspace can be obtained directly from the derivatives. In Section 3.2, we show how to modify the algebraic GPCA algorithm with a multiple-hypotheses voting scheme to estimate the subspaces. This voting-based GPCA algorithm has been shown to outperform other extant variations. When neither the number of subspaces nor their dimensions are known, we introduce relevant model-selection criteria for choosing the optimal subspace arrangement for a given set of noisy samples in Section 3.3.

3.1. Estimation of Vanishing Polynomials. From the previous section, we know that GPCA is based on the concept that we are able to identify correctly a set of (linearly independent) polynomials $Q(\boldsymbol{X}) = \big(q_1(\boldsymbol{X}), q_2(\boldsymbol{X}), \ldots, q_m(\boldsymbol{X})\big)^T$, say of degree n, whose zero set is exactly the subspace arrangement

$$\mathcal{A} = V_1 \cup V_2 \cup \cdots \cup V_n = \{z \in \mathbb{R}^D : Q(z) = 0\}. \tag{3.1}$$

For noisy samples, the algebraic GPCA algorithm is modified by replacing the null space of the embedded data matrix L_n by the eigenspace associated to the smallest eigenvalues. In order for such a *least-square fitting*

$$\min_{\boldsymbol{c}} \|\nu_n(\boldsymbol{z})^T \boldsymbol{c}\|^2 \tag{3.2}$$

to be statistically optimal, one needs to assume that the embedded data vector $\nu_n(\boldsymbol{z})$ has an isotropic Gaussian distribution. In practice, it is often more natural and mean-

ingful to assume instead that the samples z_i themselves are corrupted by (isotropic) Gaussian noise. That is, we assume that for each sample point z_i,

$$z_i = \hat{z}_i(c) + n_i, \quad i = 1, 2, \ldots, N, \tag{3.3}$$

where $\hat{z}_i(c)$ is a point on the subspace arrangement determined by c and n_i is an independent isotropic Gaussian random noise added to $\hat{z}_i(c)$. If the arrangement is clearly indicated from the context, we also write $\hat{z}_i(c)$ as $\hat{z}$. It is easy to verify that with respect to this noise model, the embedded data vector $\nu_n(z_i)$ no longer has a Gaussian distribution and subsequently the least-square fitting no longer gives the optimal estimate of the vanishing polynomials. In fact, under the Gaussian noise model, the maximum-likelihood estimate minimizes the *mean square distance*:

$$\min_{c} \frac{1}{N} \sum_{i=1}^{N} \|z_i - \hat{z}_i(c)\|^2. \tag{3.4}$$

However, it is difficult to minimize (3.4), because the closest point $\hat{z}_i(c)$ to z_i is a complicated function of the polynomial coefficients c. To resolve this difficulty, in practice we often use the first order approximation of $z_i - \hat{z}_i$ as a replacement for the mean square distance. This leads to the Sampson distance that we now introduce.

3.1.1. Sampson Distance. We assume that the polynomials in $Q(X)$ are linearly independent. Given a point z close to the zero set of $Q(X)$, i.e. the subspace arrangement $\mathcal{A}$, we let $\hat{z}$ denote the point closest to z on $\mathcal{A}$. Using the Taylor series of $Q(X)$ expanded at z, the value of $Q(X)$ at $\hat{z}$ is then given by

$$Q(\hat{z}) = Q(z) + \mathcal{J}(Q)(z)(\hat{z} - z) + O(\|\hat{z} - z\|^2). \tag{3.5}$$

After ignoring the higher order terms and noting that $Q(\hat{z}) = 0$, we have

$$z - \hat{z} \approx \left(\mathcal{J}(Q)(z)^T \mathcal{J}(Q)(z)\right)^{\dagger} \mathcal{J}(Q)(z)^T Q(z) \quad \in \mathbb{R}^D, \tag{3.6}$$

where $\left(\mathcal{J}(Q)(z)^T \mathcal{J}(Q)(z)\right)^{\dagger}$ is the pseudo inverse of the matrix $\mathcal{J}(Q)(z)^T \mathcal{J}(Q)(z)$. Thus, the approximate square distance from z to $\mathcal{A}$ is given by

$$\|z - \hat{z}\|^2 \approx Q(z)^T \left(\mathcal{J}(Q)(z) \mathcal{J}(Q)(z)^T\right)^{\dagger} Q(z) \in \mathbb{R}. \tag{3.7}$$

The expression on the right-hand side is known as the *Sampson distance* [41]. Thus, the average Sampson distance

$$\frac{1}{N} \sum_{i=1}^{N} Q(z_i)^T \left(\mathcal{J}(Q)(z_i) \mathcal{J}(Q)(z_i)^T\right)^{\dagger} Q(z_i) \tag{3.8}$$

is an approximation of the mean square distance (3.4). Minimizing the Sampson distance typically leads to a good approximation to the maximum-likelihood estimate that minimizes the mean square distance.

There is however a certain redundancy in the expression of Sampson distance. If $\mathcal{A}$ is the zero set of $Q(X)$, it is also the zero set of the polynomials $\tilde{Q}(X) = MQ(X)$ for any non-singular matrix $M \in \mathbb{R}^{m \times m}$. It is easy to check that the Sampson distance (3.7) is invariant under the non-singular linear transformation M. Thus, the estimate of polynomials in Q that minimize the average Sampson distance (or the mean square

error) is not unique, at least not in terms of the coefficients of the polynomials in $Q(\boldsymbol{X})$.

One way to reduce the redundancy is to impose some constraints on the coefficients of the polynomials in $Q(\boldsymbol{X})$. Notice that

$$\mathcal{J}(\tilde{Q})(\boldsymbol{z}_i)\mathcal{J}(\tilde{Q})(\boldsymbol{z}_i)^T = M\mathcal{J}(Q)(\boldsymbol{z}_i)\mathcal{J}(Q)(\boldsymbol{z}_i)^T M^T$$

and if there is no polynomial of lower degree (than those in $Q(\boldsymbol{X})$) that vanishes on $\mathcal{A}$, the matrix

$$\frac{1}{N}\sum_{i=1}^{N}\mathcal{J}(Q)(\boldsymbol{z}_i)\mathcal{J}(Q)(\boldsymbol{z}_i)^T \in \mathbb{R}^{m\times m}$$

is a positive-definite symmetric matrix. Therefore, we can choose the matrix M such that the matrix below is the identity:

$$\frac{1}{N}\sum_{i=1}^{N}\mathcal{J}(Q)(\boldsymbol{z}_i)\mathcal{J}(Q)(\boldsymbol{z}_i)^T = I_{m\times m}. \tag{3.9}$$

Thus, the problem of minimizing the average Sampson distance now becomes a constrained nonlinear optimization problem:

$$\begin{aligned}Q^* &= \arg\min_P \frac{1}{N}\sum_{i=1}^{N}Q(\boldsymbol{z}_i)^T\big(\mathcal{J}(Q)(\boldsymbol{z}_i)\mathcal{J}(Q)(\boldsymbol{z}_i)^T\big)^\dagger Q(\boldsymbol{z}_i),\\ &\text{subject to} \quad \frac{1}{N}\sum_{i=1}^{N}\mathcal{J}(Q)(\boldsymbol{z}_i)\mathcal{J}(Q)(\boldsymbol{z}_i)^T = I_{m\times m}.\end{aligned} \tag{3.10}$$

Many nonlinear optimization algorithms can be employed here to minimize the above objective function via iterative gradient-descent techniques. However, in order for the iterative process to converge quickly to the global minimum, a good initialization is needed. Below we discuss one such method.

3.1.2. Generalized Eigenvector Fit. Notice that the linear transformations that preserve the identity (3.9) are unitary transformations, the group of which is denoted by $O(m) = \{R \in \mathbb{R}^{m\times m} : R^T R = I_{m\times m}\}$. Obviously, the least-square fitting error is invariant under unitary transformations: $\|RQ(\boldsymbol{z})\|^2 = \|Q(\boldsymbol{z})\|^2$. In addition, as the identity matrix $I_{m\times m}$ is the average of the matrices $\mathcal{J}(Q)(\boldsymbol{z}_i)\mathcal{J}(Q)(\boldsymbol{z}_i)^T$, we can use the identity matrix to approximate each $\mathcal{J}(Q)(\boldsymbol{z}_i)\mathcal{J}(Q)(\boldsymbol{z}_i)^T$. With this approximation, the Sampson distance (3.7) becomes the least-square fitting error:

$$Q(\boldsymbol{z})^T\big(\mathcal{J}(Q)(\boldsymbol{z})\mathcal{J}(Q)(\boldsymbol{z})^T\big)^\dagger Q(\boldsymbol{z}) \approx Q(\boldsymbol{z})^T Q(\boldsymbol{z}) = \|Q(\boldsymbol{z})\|^2. \tag{3.11}$$

This leads to the following constrained optimization problem:

$$\begin{aligned}Q^* &= \arg\min_Q \frac{1}{N}\sum_{i=1}^{N}\|Q(\boldsymbol{z}_i)\|^2,\\ &\text{subject to} \quad \frac{1}{N}\sum_{i=1}^{N}\mathcal{J}(Q)(\boldsymbol{z}_i)\mathcal{J}(Q)(\boldsymbol{z}_i)^T = I_{m\times m}.\end{aligned} \tag{3.12}$$

This problem has a simple linear algebraic solution. Without loss of generality, we assume that all the polynomials in $Q(\boldsymbol{X})$ are of degree n and there is no polynomial of degree strictly less than n that vanishes on the subspace arrangement $\mathcal{A}$ of interest. Homogeneous polynomials of degree n have the form:

$$q_i(\boldsymbol{X}) = \nu_n(\boldsymbol{X})^T\boldsymbol{c}_i, \quad i = 1, 2, \ldots, m. \tag{3.13}$$

Let $\boldsymbol{C} \doteq (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_m)$. Then we have $Q(\boldsymbol{X}) = \boldsymbol{C}^T \nu_n(\boldsymbol{X})$ and $\mathcal{J}(Q)(\boldsymbol{X}) = \boldsymbol{C}^T \nabla \nu_n(\boldsymbol{X})$. Define two matrices

$$\Sigma \doteq \frac{1}{N} \sum_{i=1}^N \nu_n(\boldsymbol{z}_i) \nu_n(\boldsymbol{z}_i)^T, \quad \Gamma \doteq \frac{1}{N} \sum_{i=1}^N \nabla \nu_n(\boldsymbol{z}_i) \nabla \nu_n(\boldsymbol{z}_i)^T. \tag{3.14}$$

Using these notations, we rewrite the above optimization problem (3.12) as

$$\boldsymbol{C}^* = \arg \min_{\boldsymbol{C}} \ \mathrm{Trace}(\boldsymbol{C}^T \Sigma \boldsymbol{C}), \quad \text{subject to} \quad \boldsymbol{C}^T \Gamma \boldsymbol{C} = I_{m \times m}. \tag{3.15}$$

In comparison, the naive least-square fitting (3.2) minimizes the same objective function but subject to a different constraint: $\boldsymbol{C}^T \boldsymbol{C} = I_{m \times m}$.

Using Lagrange multipliers and the necessary conditions for minima, one can show that the optimal solution $\boldsymbol{C}^*$ is such that its ith column $\boldsymbol{c}_i^*$ is the ith generalized eigenvector of the matrix pair (Σ, Γ):

$$\Sigma \boldsymbol{c}_i^* = \lambda_i \Gamma \boldsymbol{c}_i^*, \quad i = 1, 2, \ldots, m, \tag{3.16}$$

where $0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_m$ are the m smallest generalized eigenvalues of (Σ, Γ). Furthermore, as Γ is non-singular,[7] $\boldsymbol{c}_i^*$ is also the eigenvector associated with the ith smallest eigenvalue of the matrix $\Gamma^{-1}\Sigma$:

$$\Gamma^{-1}\Sigma \boldsymbol{c}_i^* = \lambda_i \boldsymbol{c}_i^*, \quad i = 1, 2, \ldots, m. \tag{3.17}$$

As the optimal solution to the problem (3.12), the polynomials $q_i(\boldsymbol{X}) = \nu_n(\boldsymbol{X})^T \boldsymbol{c}_i^*$ usually give a good initialization to the problem (3.10). It usually takes only a few more iterations for any reasonable gradient-descent method (such as the Levenberg-Marquardt) to converge to the (global) minimum.

The generalized eigenvector fit has yet another statistical explanation from the viewpoint of (Fisher) discriminant analysis. The matrix Σ can be viewed as a measure of the intra-class distance – the closer a point is to one of the subspaces, the smaller the (absolute) value of a fitting polynomial; and the matrix Γ can be viewed as a measure of the inter-class distance – the norm of the derivative at a point in a subspace is roughly proportional to its distance to other subspaces.[8] According to discriminant analysis, the optimal polynomial $q(\boldsymbol{X}) = \nu_n(\boldsymbol{X})^T \boldsymbol{c}^*$ for discriminating the subspaces minimizes the Rayleigh quotient:

$$\boldsymbol{c}^* = \arg \min_{\boldsymbol{c}} \frac{\boldsymbol{c}^T \Sigma \boldsymbol{c}}{\boldsymbol{c}^T \Gamma \boldsymbol{c}}. \tag{3.18}$$

It is then easy to show that the optimal solution $\boldsymbol{c}^*$ is exactly the generalized eigenvector of the matrix pair (Σ, Γ). Therefore, the fitting polynomials found via the generalized eigenvector fit are the ones that are in a sense optimal for segmenting the multiple subspaces.

3.1.3. Simulation Results. In this subsection, we demonstrate by simulation how the normalization by Γ may significantly improve the eigenvalue spectrum of Σ. That is, the generalized eigenvectors of (Σ, Γ) are less sensitive to the corruption of

---

[7]Otherwise there would be polynomial of degree less than n that fits the data, which contradicts our assumptions.

[8]This is easy to see from an arrangement of hyperplanes.

noise than the null space of Σ, which makes the estimation of the fitting polynomials a more well conditioned problem. To see this, let us consider a set of points drawn from two lines and one plane in $\mathbb{R}^3$ (see Figure 1.1) – 1000 points from the plane and 200 points from each line – with 5% Gaussian noise added.[9] As Figure 3.1 illustrates, the generalized eigenvalues of (Σ, Γ) provide a much sharper "knee point" than the eigenvalues of Σ. With the new spectrum, one can more easily estimate the correct number of polynomials that fit the data (in this case four polynomials).
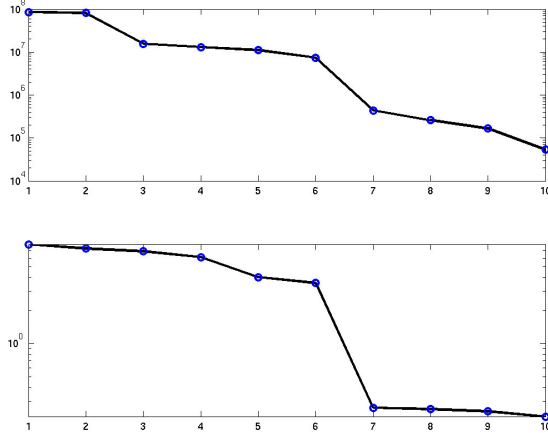

FIG. 3.1. *Top: Plot of the eigenvalues of the matrix Σ. Bottom: Plot of the eigenvalues of the matrix $\Gamma^{-1}\Sigma$.*

3.2. Estimation of Subspace Arrangements via a Voting Scheme. In the algebraic GPCA algorithm, the basis of each subspace is computed as the orthogonal complement to the derivatives of the fitting polynomials at a representative sample point. However, if the chosen point is noisy, it may cause a large error in the estimated basis and subsequently cause a large error in the segmentation. From a statistical point of view, more accurate estimates of the basis can be obtained only if we are able to compute an average of the derivatives at many points in the same subspace. However, a fundamental difficulty here is that we do not know which points belong to the same subspace. There is yet another issue. In the algebraic GPCA algorithm, the rank of the derivatives at each point is the codimension of the subspace to which it belongs. In the presence of noise, one can determine the rank from the singular values of the derivatives, i.e., using principal component analysis (PCA). However, the rank can be erroneous if the chosen point is noisy. It is also difficult to find a uniform threshold for PCA that works for points in different subspaces.

We assume below that the number of subspaces and their dimensions are known. We show how to improve the estimates of the basis of each subspace by using collectively the derivatives at all the sample points. The algorithm relies on a voting method on the feature space of subspace basis parameters, which was inspired by the classical Hough transform [2, 47]. An important difference here is that we do not quantize the feature space of basis vector parameters, since it is impossible to store

---

[9]The percentage is computed as the variance of the Gaussian relative to the diameter of the data set.

the whole quantized space in a computer when the dimensions of the subspaces are high.

3.2.1. Arrays of Bases and Counters. Suppose the subspace arrangement is a union of n subspaces: $\mathcal{A} = V_1 \cup V_2 \cup \cdots \cup V_n$. Let the dimensions of the subspaces be $d_1, d_2, \ldots, d_n$ and their codimensions be $c_1, c_2, \ldots, c_n$. Pick a sample point $\boldsymbol{z}_1$. The Jacobian of the fitting polynomials $Q(\boldsymbol{X})$ at $\boldsymbol{z}_1$ is $\mathcal{J}(Q)(\boldsymbol{z}_1)$. If there is no noise, then $\mathrm{rank}(\mathcal{J}(Q)(\boldsymbol{z}_1))$ will be exactly the codimension of the subspace to which $\boldsymbol{z}_1$ belongs. When the samples are noisy, it is very difficult to determine the codimension in this way. However, we know all the codimensions and the number of subspaces having the same codimension. The idea is to calculate a basis under each codimension assumption, and invoke a voting method to search for high consensus among all the basis candidates over all samples.

Without loss of generality, we assume that $c_1, c_2, \ldots, c_n$ have l distinct values $c'_1 < c'_2 < \ldots < c'_l$. As we do not know the true codimension at the sample point $\boldsymbol{z}_1$, we compute a set of candidate bases in column form:

$$B_i(\boldsymbol{z}_1) \in \mathbb{R}^{D \times c'_i}, \quad i = 1, 2, \ldots, l, \tag{3.19}$$

as $B_i(\boldsymbol{z}_1)$ collects the first c'_i principal components of $\mathcal{J}(Q)(\boldsymbol{z}_1)$. Thus, each $B_i(\boldsymbol{z}_1)$ is a $D \times c'_i$ orthogonal matrix.

To store the basis candidates $B_1(\boldsymbol{z}_j), B_2(\boldsymbol{z}_j), \ldots, B_l(\boldsymbol{z}_j)$ for all samples $j = 1, 2, \ldots, N$, we create l arrays of bases $U_1, U_2, \ldots, U_l$, where each U_i stores all candidate $D \times c'_i$ matrices. Correspondingly, we create l arrays of voting counters $u_1, u_2, \ldots, u_l$. Suppose $U_i(j) \in \mathbb{R}^{D \times c'_i}$ stores a candidate basis, then $u_i(j)$ is an integer that counts the number of sample points $\boldsymbol{z}_k$ with $B_i(\boldsymbol{z}_k) = U_i(j)$. Notice numerically any other $B_i(\boldsymbol{z}_k)$ cannot exactly be equal to $U_i(j)$. We introduce a similarity measure to compare two set of basis vectors in the next subsection.

3.2.2. Tally the Votes of the Subspaces. With the above definitions, we now outline an algorithm that will select a set of bases for the n subspaces that achieves the highest consensus on all the sample points. Suppose J_i is the size of the array U_i and hence u_i for all $i = 1, 2, \ldots, l$. Initially, all J_i's are equal to zero. For every sample point $\boldsymbol{z}_k$,

1. we compute a set of basis candidates $B_i(\boldsymbol{z}_k), i = 1, 2, \ldots, l$ as in (3.19);
2. for each $B_i(\boldsymbol{z}_k)$, we compare it with the bases already in the array U_i:
   (a) if $B_i(\boldsymbol{z}_k) = U_i(j)$ for some j, then increase the value of $u_i(j)$ by one;
   (b) if $B_i(\boldsymbol{z}_k)$ is different from any of the bases in U_i, then add $U_i(J_i + 1) = B_i(\boldsymbol{z}_k)$ as a new basis to U_i, and also add a new counter $u_i(J_i + 1)$ to u_i with the initial value $u_i(J_i + 1) = 1$. Set $J_i \leftarrow J_i + 1$.

In the end, the bases of the n subspaces are chosen to be the n bases in the arrays $\{U_1, U_2, \ldots, U_l\}$ that have the highest votes according to the corresponding counters in $\{u_1, u_2, \ldots, u_l\}$. For instance, suppose the codimensions of 4 subspaces are $1, 3, 3, 4$ in $\mathbb{R}^5$ and the distinct codimensions are $c'_1 = 1, c'_2 = 3$, and $c'_3 = 4$. Then after the bases are evaluated at all the samples, we select one basis candidate from U_1 and one from U_3 with the largest numbers in u_1 and u_3, respectively, and two basis candidates from U_2 with the largest two numbers in u_2.

In the above scheme, in order to compare $B_i(\boldsymbol{z}_k)$ with bases in U_i when the data are noisy, we need to set an error tolerance. This tolerance, denoted by τ, can be

a small subspace angle chosen by the user.[10] Thus, if the subspace angle difference between $B_i(\boldsymbol{z}_k)$ and $U_i(j)$

$$\langle B_i(\boldsymbol{z}_k), U_i(j)\rangle \tag{3.20}$$

is less than τ, we set the new value of $U_i(j)$ to be the average of its votes with $B_i(\boldsymbol{z}_k)$ added in:

$$U_i(j) \;\leftarrow\; \frac{1}{u_i(j)+1}\big(u_i(j)U_i(j) + B_i(\boldsymbol{z}_k)\big), \tag{3.21}$$

and increase the value of the counter $u_i(j)$ by one. Notice that the weighted sum may no longer be an orthogonal matrix. If so, apply the Gram-Schmidt process to make $U_i(j)$ an orthogonal matrix again.

In case when subspaces have different dimensions, for the same sample point $\boldsymbol{z}$, it is possible more than one candidate basis $B_{i_1}(\boldsymbol{z}), B_{i_2}(\boldsymbol{z}), \ldots, B_{i_j}(\boldsymbol{z})$ belongs to more than one counter that has the highest vote. Our analysis shows that this ambiguity is mainly caused by the fact that samples on a high-dimensional subspace may form a low-dimensional subspace, e.g., sample points on a 2-D plane along the same direction may result in high consensus as a line. To resolve this ambiguity, we choose highest votes starting with the smallest codimension c_1'. When a sample is grouped to a subspace of smaller codimension, its votes in other counters will be removed and the associated bases $U_i(j)$ recalculated. We summarize the overall process as Algorithm 2.

Algorithm 2 (Generalized Principal Component Analysis with Voting).

Given a set of samples $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$ in $\mathbb{R}^D$ and a parameter for angle tolerance τ, fit n linear subspaces with codimensions $c_1, c_2, \ldots, c_n$:

1: Suppose there are l distinct codimensions, ordered as $c_1' < c_2' < \cdots < c_l'$. Allocate $u_1, u_2, \ldots, u_l$ to be l stacks of counters and $U_1, U_2, \ldots, U_l$ be l stacks of candidate bases.
2: Estimate the set of fitting polynomials $Q(\boldsymbol{X})$, and compute their derivatives $\mathcal{J}(Q)(\boldsymbol{X})$ for all $\boldsymbol{z}_k$.
3: **for all** sample $\boldsymbol{z}_k$ **do**
4:     **for all** $1 \le i \le l$ **do**
5:         Assume $\boldsymbol{z}_k$ is drawn from a subspace of codimension c_i'. Find the first c_i' principal vectors of $\mathcal{J}(P)(\boldsymbol{z}_k)$ and stack them into the matrix $B_i(\boldsymbol{z}_k) \in \mathbb{R}^{D \times c_i'}$.
6:         If $\langle B_i(\boldsymbol{z}_k), U_i(j)\rangle < \tau$ for some j, increase $u_i(j)$ by one and reset $U_i(j)$ to be the weighted sum in (3.21). Otherwise, create a new candidate basis in U_i and a new counter in u_i with initial value one.
7:     **end for**
8: **end for**
9: **for all** $1 \le i \le l$ **do**
10:     Choose the highest vote(s) in u_i with their corresponding basis/bases in U_i.
11:     Assign the samples to their closest subspaces, and remove their votes in other counters and bases of higher codimensions.
12: **end for**
13: Segment the remaining samples that are not in the stacks of the highest votes based on the estimated bases.

There are important features about the above voting scheme that are quite different from the well-known statistical learning methods K-Subspaces [20] and EM [34] for estimating subspace arrangements. The K-Subspaces and EM algorithms iteratively update one basis for each subspace; while the voting scheme essentially keeps multiple candidate bases per subspace through the process. Thus, the voting algorithm does not have the same difficulty with local minima as K-subspaces and EM do.

There are other voting or random sampling methods developed in statistics and machine learning, such as the *least median estimate* (LME) and *random sampling consensus* (RANSAC). These methods are similar in nature as they compute multiple candidate models from multiple down-sampled subsets of the data and then choose the one which achieves the highest consensus (for RANSAC) or smallest median error (for LME). The data that do not conform to the model are regarded as outliers. We will discuss these methods in the context of dealing with outliers in Section 4.

3.2.3. Simulation Results. We provide a comparison of various algorithms for the estimation and segmentation of subspace arrangements that we have mentioned so far. They include: The EM algorithm, the K-Subspaces algorithm, the algebraic GPCA algorithm, the GPCA algorithm with voting, as well as some combination of them.

We randomly generate some subspace arrangements of some pre-chosen dimensions. For instance, $(2, 2, 1)$ indicates an arrangement of three subspaces of dimensions $2, 2, 1$, respectively. We then randomly draw a set of samples from them. The samples are corrupted with Gaussian noises. Here we choose the level of noise to be 4%. The error is measured in terms of the percentage of sample points that are wrongfully grouped.[11] All cases are averaged over 100 trials. The performance of all the algorithms are compared in Table 3.1. The reader can download the MATLAB codes from our website.

TABLE 3.1
The percentage of sample points mis-grouped by different algorithms. The number of subspaces and their dimensions are given to all algorithms. The EM and K-Subspaces algorithms are randomly initialized. "GPCA-Voting+K-Subspaces" means the K-Subspaces method initialized with the GPCA-Voting algorithm. The sample number for each subspace is 200 times its dimension.

Methods	$(2, 2, 1) \in \mathbb{R}^3$	$(2, 2, 2) \in \mathbb{R}^3$	$(4, 2, 2, 1) \in \mathbb{R}^5$	$(4, 4, 4, 4) \in \mathbb{R}^5$
EM	29%	11%	53%	20%
K-Subspaces	27%	12%	57%	25%
Algebraic GPCA	10.3%	10.6%	39.8%	25.3%
GPCA-Voting	6.4%	9.2%	5.7%	17%
GPCA-Voting + K-Subspaces	5.4%	8.6%	5.7%	11 %

As we can see from this table, the voting scheme significantly improves the performance of the (naive) algebraic GPCA algorithm. The performance is further improved by running K-subspaces as a post-processing step, but not much.

3.3. Model Selection Criteria for Subspace Arrangements. The methods that we have discussed so far for the estimation of subspace arrangements (e.g., EM,

---

[11] Notice that even with prior knowledge of the subspaces, due to the samples drawn at subspace intersections and sample noises, the segmentation error cannot be zero.

K-Subspaces, GPCA) assume that the number of subspaces and their dimensions are known. If they are *not* given, the problem of fitting multiple subspaces to a set of samples becomes much more elusive. For instance, sample points drawn from two lines and one plane in $\mathbb{R}^3$ can also be fit by two planes, one of which is spanned by the two lines. In Section 2, we have suggested that in this case one can apply the algebraic GPCA algorithm in a *recursive* fashion to identify all the subspaces and their dimensions.

However, when there is noise in the given data, the purely algebraic GPCA algorithm may fail to return a meaningful solution. In fact, up till now, we have been purposely avoiding a fundamental difficulty in our problem: it is inherently *ambiguous* in fitting multiple subspaces for any given data set when the number of subspaces and their dimensions are not given *a priori*. When the data are noisy or nonlinear, it is unlikely that any model can fit the data perfectly except for the pathological cases: 1. All points are viewed as in a D-dimensional subspace – the ambient space; 2. Every point is viewed as on an individual one-dimensional subspace through the origin. In general, the more subspaces we use to *overfit* a data set, the higher accuracy we may achieve. Thus, a fundamental question we address in this section is: Among the class of subspace arrangements, what is the "optimal" model that fits a given data set? From a practical point of view, we also need to know under what conditions the optimal model exists and is unique, and more importantly, how to compute it efficiently.

3.3.1. Model Selection Criteria for Subspaces. Many general-purpose model-selection criteria have been developed in the statistics community and the algorithmic complexity community for general classes of models. These criteria include

- Akaike Information Criterion (AIC) [1] (also known as the C_p statistic [33]) and Geometric AIC (G-AIC) [26].
- Bayesian Information Criterion (BIC) (also known as the Schwartz criterion, see [19] and references therein).
- Minimum Description Length (MDL) [37] and Minimum Message Length (MML) [55].

Although these criteria are originally motivated and derived from different points of view (or in different contexts), they all share a common characteristic: The optimal model should be the one that strikes a good *balance* between the model complexity (typically depends on the dimension of the parameter space) and the data fidelity to the chosen model (e.g., measured as the sum of squared errors assuming a Gaussian noise model). In fact, some of the criteria are essentially equivalent despite their different origins: Roughly speaking, BIC is equivalent to MDL; and AIC is equivalent to the C_p statistic. It is not the intention to give a detailed review of all the model selection criteria in this paper. In the following, we use AIC to illustrate some of the key ideas behind model selection.

Given N independent sample points $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$ drawn from a distribution denoted by $p(\boldsymbol{z}, \theta_0)$, the maximum-likelihood estimate $\hat{\theta}_N$ of the parameter θ is the one that maximizes the log-likelihood function $L(\theta, \boldsymbol{Z}) = \sum_{i=1}^{N} \log p(\boldsymbol{z}_i, \theta)$. From an information-theoretic point of view:

$$E[-\log p(\boldsymbol{z}, \hat{\theta}_N)] = \int \big(-\log p(\boldsymbol{z}, \hat{\theta}_N)\big) p(\boldsymbol{z}, \theta_0) \ d\boldsymbol{z} \qquad (3.22)$$

corresponds to the expected code length that we use the optimal coding scheme of $p(\boldsymbol{z}, \hat{\theta}_N)$ for a random variable with actual distribution $p(\boldsymbol{z}, \theta_0)$. Thus, for model

selection, it is desirable to choose the model that minimizes the above expected log-likelihood loss.

The *Akaike information criterion* (AIC) relies on an approximation to the above expected log-likelihood loss that holds asymptotically as $N \to \infty$:

$$\text{AIC} \doteq -\frac{2}{N} L(\hat{\theta}_N, \boldsymbol{Z}) + 2\frac{d}{N} \approx 2E[-\log p(\boldsymbol{z}, \hat{\theta}_N)], \qquad (3.23)$$

where d is the number of free parameters for the class of models of interest. For an (isotropic) Gaussian noise model with variance σ^2, we have

$$L(\hat{\theta}_N, \boldsymbol{Z}) = -\frac{1}{2\sigma^2} \sum_{i=1}^{N} \|\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\|^2,$$

where $\hat{\boldsymbol{z}}_i$ is the best estimate of $\boldsymbol{z}_i$ given the model $p(\boldsymbol{z}, \hat{\theta}_N)$. Thus, if σ^2 is known (or approximated by the empirical sample variance), minimizing AIC is equivalent to minimizing the C_p statistic: $C_p = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\|^2 + 2\frac{d}{N}\sigma^2$, where the first term is obviously the mean squared error (a measure of the data fidelity) and the second term depends linearly on the dimension of the parameter space (a measure of the complexity of the model).

Now consider multiple classes of models whose parameter spaces are of different dimensions. Denote the dimension of model class m by $d(m)$. Then AIC selects the model class m^* that minimizes the following objective function:

$$\text{AIC}(m) = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\|^2 + 2\frac{d(m)}{N}\sigma^2. \qquad (3.24)$$

Although motivated by a different reason, the Bayesian information criterion (BIC) results in a formula similar to AIC except that the factor 2 in front of the second term in AIC is replaced by $\log(N)$ in BIC. Because normally $\log(N) \gg 2$, BIC penalizes complex models much more than AIC does. Thus, BIC tends to choose simpler models. In general, no model selection criterion is always better than others under all circumstances; the best criterion depends on the purpose of the model. From our experience, AIC tends to provide more satisfactory results for estimation of subspaces. That makes it more favorable in the context of PCA and GPCA.

We now discuss how to apply the above criterion to the problem of principal component analysis (PCA) where we try to fit a subspace V of an unknown dimension d in $\mathbb{R}^D$ to a given set of data points $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\} \subset \mathbb{R}^D$. Denote the projection of each data point $\boldsymbol{z}_i \in \boldsymbol{Z}$ onto the subspace by $\hat{\boldsymbol{z}}_i$ and let $\hat{\boldsymbol{Z}} = \{\hat{\boldsymbol{z}}_1, \hat{\boldsymbol{z}}_2, \ldots, \hat{\boldsymbol{z}}_N\}$. Then, the sum of squared errors is $\|\boldsymbol{Z} - \hat{\boldsymbol{Z}}\|^2 = \sum_{i=1}^{N} \|\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\|^2$.

The Grassmannian variety of dimension d subspaces of $\mathbb{R}^D$ has dimension $(D-d)d$. Therefore the Akaike information criterion (AIC) minimizes

$$\text{AIC}(d) \doteq \frac{1}{N} \|\boldsymbol{Z} - \hat{\boldsymbol{Z}}\|^2 + 2\frac{(Dd - d^2)}{N}\sigma^2 \qquad (3.25)$$

for our model with parameter space of dimension $Dd - d^2$ and Gaussian noise with variance σ^2, More recently, a geometric version of the Akaike information criterion has been proposed by [26] which minimizes

$$\text{GAIC}(d) \doteq \frac{1}{N} \|\boldsymbol{Z} - \hat{\boldsymbol{Z}}\|^2 + 2\frac{(Dd - d^2 + Nd)}{N}\sigma^2, \qquad (3.26)$$

where the extra term Nd accounts for the number of coordinates needed to represent (the closest projection of) the given N data points in the estimated d-dimensional subspace. From the information-theoretic point of view, the additional Nd coordinates are necessary if we are interested in encoding not only the model but also the given data. This is often the case when we use PCA or GPCA for purposes such as data compression and dimension reduction.

3.3.2. Effective Dimension of Samples on a Subspace Arrangement. If we were to apply any of the model-selection criteria (or their concepts) to subspace arrangements, at least two needs must be addressed::

1. We need to know how to measure the model complexity of arrangements of subspaces (possibly of different dimensions).
2. We need to know how to balance properly the model complexity and the modeling error for subspace arrangements, since the choice of a subspace arrangement involves both continuous parameters (the subspace bases) and discrete parameters (the number of subspaces and their dimensions),

Although model selection for subspace arrangements in its full generality is still an open problem at this point, in the next two subsections, we introduce a few specific approaches to attempt to solve the problem of model selection from slightly different aspects. We hope the basic concepts introduced in this subsection and the next may help the reader to appreciate better the subtlety and difficulty of the problem.

DEFINITION 3.1 (Effective Dimension). *Given an arrangement of n subspaces $\mathcal{A} \doteq \cup_{j=1}^{n} V_j$ in $\mathbb{R}^D$ of dimension $d_j < D$, and N_j sample points $\boldsymbol{Z}_j$ drawn from each subspace V_j, the* effective dimension *of the entire set of $N = \sum_{j=1}^{n} N_j$ sample points, $\boldsymbol{Z} = \cup_{j=1}^{n} \boldsymbol{Z}_j$, is*

$$\text{ED}(\boldsymbol{Z}, \mathcal{A}) \doteq \frac{1}{N} \Big(\sum_{j=1}^{n} d_j (D - d_j) + \sum_{j=1}^{n} N_j d_j \Big). \tag{3.27}$$

We contend that $\text{ED}(\boldsymbol{Z}, \mathcal{A})$ is the "average" number of (unquantized) real numbers one needs to assign to $\boldsymbol{Z}$ per sample point in order to specify the configurations of the n subspaces and the relative locations of the sample points in the subspaces.[12] In the first term of equation (3.27), $d_j(D-d_j)$ is the total number of Grassmann coordinates needed to specify a d_j-dimensional subspace V_j in $\mathbb{R}^D$; in the second term of (3.27), $N_j d_j$ is the total number of real numbers needed to specify the d_j coordinates of the N_j sample points in the subspace V_j. In general, if there are more than one subspace in $\mathcal{A}$, $\text{ED}(\boldsymbol{Z}, \mathcal{A})$ can be a rational number, instead of an integer for the conventional dimension.

Notice that in the above definition, the effective dimension of $\boldsymbol{Z}$ depends on the subspace arrangement $\mathcal{A}$. The reason is that in general there are many subspace arrangements that can fit the same data set $\boldsymbol{Z}$, as we have discussed in the beginning of this section. Therefore, we define the *minimum effective dimension* of a given sample set $\boldsymbol{Z}$ to be the minimum one among all possible subspace arrangements that can fit the data set:[13]

$$\text{MED}(\boldsymbol{Z}) \doteq \min_{\mathcal{A}: \boldsymbol{Z} \subset \mathcal{A}} \text{ED}(\boldsymbol{Z}, \mathcal{A}). \tag{3.28}$$

---

[12]We here choose real numbers as the basic "units" for measuring complexity in a similar fashion to binary numbers, "bits," traditionally used in algorithmic complexity or coding theory.

[13]The space of all subspace arrangements (with a bounded number of subspaces) is topologically compact and closed, hence the minimum effective dimension is always achievable and hence well-defined.

EXAMPLE 3.2. [Samples from One Plane and Two Lines] As shown in Figure 1.1, suppose that we have a set of samples drawn from one plane and two lines in $\mathbb{R}^3$. Obviously, the points in the two lines can also be viewed as lying in the plane that is spanned by the two lines. However, that interpretation would result in an increase of the effective dimension since one would need two coordinates to specify a point in a plane, as opposed to one in a line. For instance, suppose there are fifteen points in each line; and thirty points in the plane. When we use two planes to represent the data, the effective dimension is: $\frac{1}{60}(2 \times 2 \times 3 - 2 \times 2^2 + 60 \times 2) = 2.07$; when we use one plane and two lines, the effective dimension is reduced to: $\frac{1}{60}(2 \times 2 \times 3 - 2^2 - 2 \times 1 + 30 \times 1 + 30 \times 2) = 1.6$. In general, if the number of points N is arbitrarily large (say approaching to infinity), depending on the distribution of the points on the lines or the plane, the effective dimension will be within 1 and 2, the true dimensions of the subspaces.

As suggested by the above example, the subspace arrangement model that leads to the minimum effective dimension normally corresponds to a "natural" and hence "efficient" representation of the data in the sense that it achieves the best dimension reduction among all possible subspace arrangements.

3.3.3. Minimum Effective Dimension of Noisy Samples. In practice, real data are corrupted with noise, hence we normally do not expect a model to fit the data perfectly. The conventional wisdom is to strike a good balance between the complexity of the chosen model and the data fidelity. As all model-selection criteria exercise the same rationale, we here adopt the geometric-AIC (GAIC) criterion (3.26)[14] and it leads to the following objective for selecting the optimal subspace arrangement model:

$$\mathcal{A}^* = \arg\min_{\mathcal{A}:\hat{\boldsymbol{Z}}\subset\mathcal{A}} \left\{ \frac{1}{N}\|\boldsymbol{Z} - \hat{\boldsymbol{Z}}\|^2 + 2\sigma^2 \mathrm{ED}(\hat{\boldsymbol{Z}}, \mathcal{A}) \right\}, \qquad (3.29)$$

where σ^2 is the variance of the Gaussian noise model (3.3). However, this optimization problem can be very difficult to solve: The variance σ^2 might not be known *a priori* and we need to search for the global minimum in the configuration space of all subspace arrangements, which is not a smooth manifold and has very complicated topological and geometric structures. The resulting computation can be prohibitive.

To alleviate some of the difficulties, in practice, we may instead minimize the effective dimension subject to a maximum allowable error tolerance. That is, among all the subspace arrangements that fit the data within a given error bound, we choose the one with the smallest effective dimension. To this end, we define the minimum effective dimension *subject to an error tolerance τ* as:

$$\mathrm{MED}(\boldsymbol{Z}, \tau) \;\doteq\; \min_{\mathcal{A}:\; \|\boldsymbol{Z}-\hat{\boldsymbol{Z}}\|_\infty \leq \tau} \mathrm{ED}(\hat{\boldsymbol{Z}}, \mathcal{A}), \qquad (3.30)$$

where $\hat{\boldsymbol{Z}}$ is the projection of $\boldsymbol{Z}$ onto the subspaces in $\mathcal{A}$ and the error norm $\|\cdot\|_\infty$ indicates the maximum norm: $\|\boldsymbol{Z} - \hat{\boldsymbol{Z}}\|_\infty = \max_{1\leq i\leq N} \|\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\|$. Based on the above definition, the minimum effective dimension of a data set now becomes a notion that depends on the error tolerance. In the extreme, if the error tolerance is arbitrarily large, the "optimal" subspace arrangement for any data set can simply be the (zero-dimensional) origin; if the error tolerance is zero instead, for data with random noise,

---

[14]We here adopt the GAIC criterion only to illustrate the basic ideas. In practice, depending on the nature of the problem and its purpose, it is possible that other model selection criteria may be more appropriate and effective.

each sample point needs to be treated as a one-dimensional subspace in $\mathbb{R}^D$ of its own and that brings the minimum effective dimension up close to D.

In many applications, the notion of maximum allowable error tolerance is particularly relevant. For instance, in image representation and compression, the task is often to find a linear or hybrid linear model to fit the imagery data subject to a given *peak signal to noise ratio* (PSNR), where the noise becomes the different between the original image and the approximate one. The resulting effective dimension directly corresponds to the number of coefficients needed to store the resulting representation. The smaller the effective dimension is, the more compact or compressed is the final representation. In Section 5.2, we will see exactly how the minimum effective dimension principle is applied to image representation. The same principle can be applied to any situation in which one tries to fit a piecewise linear model to a data set whose structure is nonlinear or hybrid.

Unlike the geometric AIC (3.29), the MED objective (3.30) is relatively easy to achieve. For instance, the recursive version of the GPCA algorithm discussed in Section 2 can be easily modified to minimize the effective dimension subject to an error tolerance: We allow the recursion to proceed only if the effective dimension would decrease while the resulting subspaces still fit the data with the given error bound.

3.3.4. Simulation Results. Figure 3.2 demonstrates the result of such a recursive GPCA algorithm segmenting synthetic data drawn from two lines (100 points each) and one plane (400 points) in $\mathbb{R}^3$ corrupted with 5% Gaussian noise. Given a reasonable error tolerance, the algorithm stops after two levels of recursion (Figure 3.2 left). Note that the pink line is a "ghost" line at the (virtual) intersection of the original plane and the plane spanned by the two lines.[15] Figure 3.2 right is the plot of the MED of the same data set subject to different levels of the error tolerance. As we see, the effective dimension decreases monotonically with the increase of the error tolerance.
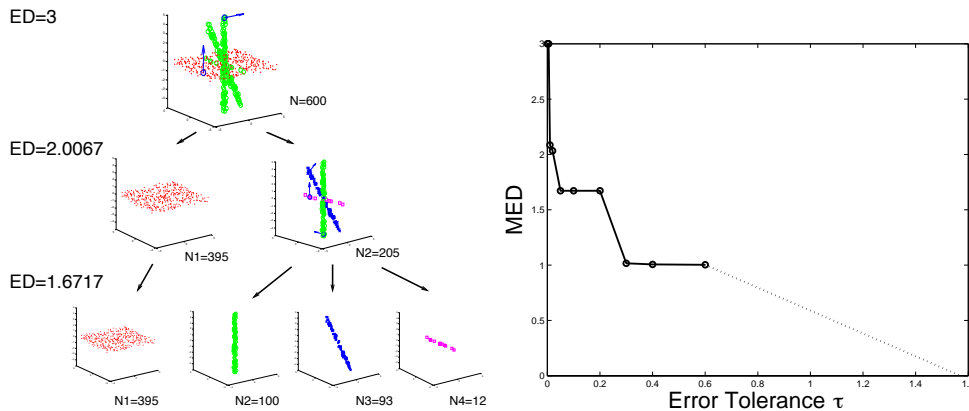

FIG. 3.2. *Left: sample points drawn from two lines and a plane in $\mathbb{R}^3$ with 5% Gaussian noise, are segmented recursively by the GPCA algorithm with an error tolerance $\tau = 0.05$; Right: plot of the MED versus the error tolerance.*

---

[15]Points on the intersection of the two planes get assigned arbitrarily to either plane depending on the random noise. If needed, the points on the ghost line can be merged with the plane by some simple post-processing.
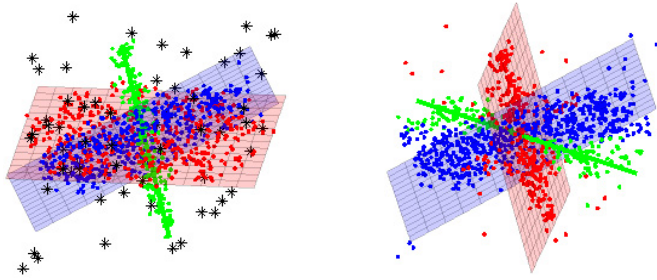
FIG. 4.1. *Results of GPCA (with voting) for samples drawn from two planes and one line in $\mathbb{R}^3$, with 6% Gaussian noise as well as 6% outliers drawn from a uniform distribution (marked as black asterisks "∗"). Left: the ground truth. Right: estimated subspaces and segmentation result.*

4. Estimation of Subspace Arrangements with Outliers. In many practical situations the sample points can be contaminated by some atypical samples known as "outliers" in addition to the noise that we have discussed above. The application of any of the GPCA algorithms to valid samples contaminated with such outliers can lead to disastrous results. Both the estimated subspaces and the segmentation can be far from the ground truth, as illustrated by the example in Figure 4.1. Thus, in this section we introduce some relevant *robust statistical techniques* that can detect or diminish the effect of outliers in estimating subspace arrangements.

Despite an extended history of interest and study, there is unfortunately no universally accepted definition of "outlier".[16] Most definitions (or tests) are based on one of the following three guidelines:

1. Outliers form a set of *small-probability* samples with respect to the distribution in question. The given data set is therefore an atypical set if such samples constitute a significant portion of the data.
2. Outliers form a set of samples that are *not consistent* with (the model inferred from) the remainder of the data. A measure of inconsistency is normally the error residue of the sample in question with respect to the model.
3. Outliers form a set of samples that have relatively *large influence* on the estimated model parameters. A measure of influence is normally the difference between the model estimated with and without the sample in question.

Despite their dissimilarity, these guidelines result in essentially equivalent criteria for testing for outliers. In different contexts, one of the guidelines may become more natural or convenient to use than the others. In our context, as our goal is to obtain the vanishing polynomials of the subspace arrangement, we assume that the "outliers", together with the valid samples, cannot be fit well by any polynomials.[17]

In the robust statistics literature, there have been extensive studies about outlier detection and rejection [18, 25, 43, 50]. Most of them are conducted with the assump-

---

[16]Earliest documented discussions among astronomers about outliers or "erroneous observations" date back to mid 18th century. See [3, 4, 23] for a more thorough exposition of the studies of outliers in statistics.

[17]In situations when a data set contains samples drawn from some nonlinear algebraic sets, e.g., a quadratic surface, it is then no longer appropriate to view such data as outliers. One way to resolve the problem is to view the linear and nonlinear structures together as an algebraic set and develop solutions to this new class of hybrid models. The interested reader may refer to [36] for more detailed discussion along this direction.

tion that the valid samples points, i.e., the *inliers*, are drawn from a conventional statistical (or geometric) model. In our case we need to examine how to apply the basic principals of robust statistics to subspace arrangements to determine which of the existing techniques is the most relevant and efficient. In this section, we will discuss first the simpler situation in which the percentage of outliers is known (Section 4.1). Two methods, namely, the influence function and robust covariance estimator, both can be adopted to robustify the GPCA algorithm. When the percentage of outliers is not given, we propose a criterion to conveniently estimate the percentage in Section 4.2. Finally, in Section 4.3, we discuss several other common robust statistical techniques such as the least median estimate (LME) [38] and random sampling consensus (RANSAC) [15]. These techniques have been widely used in the areas of computer vision, image processin, and pattern recognition. However, their own drawbacks have limited them from being useful for subspace arrangements.

4.1. Robust Estimation of Vanishing Polynomials.

4.1.1. Influence Functions. When we try to estimate the parameter θ of the distribution $p(z, \theta)$ from a set of samples $\{z_1, z_2, \ldots, z_N\}$, every sample z_i might have uneven effect on the estimated parameter $\hat{\theta}$. The samples that have relatively large effect are called *influential samples* and they can be regarded as outliers [8, 10, 18].

To measure the influence of a particular sample z_i, we may compare the difference between the parameter $\hat{\theta}$ estimated from all the N samples and the parameter $\hat{\theta}^{(i)}$ estimated from all but the ith sample. We consider the maximum-likelihood estimate as an example:

$$\hat{\theta} = \arg\max_{\theta} \sum_{j=1}^{N} \log p(z_j, \theta), \quad \hat{\theta}^{(i)} = \arg\max_{\theta} \sum_{j \neq i} \log p(z_j, \theta), \tag{4.1}$$

and the influence of z_i on the estimation of θ can be measured by the difference

$$I(z_i; \theta) \doteq \|\hat{\theta} - \hat{\theta}^{(i)}\|. \tag{4.2}$$

The function $I(z_i; \theta)$ is also called the *sample influence function* in the literature of robust statistics.

If a set of sample points $\{z_1, z_2, \ldots, z_N\}$ is drawn from a subspace arrangement $\mathcal{A} = \cup_{i=1}^{n} V_i \subset \mathbb{R}^D$, then GPCA relies on obtaining the set of polynomials $Q(\boldsymbol{X}) = \{q_1(\boldsymbol{X}), q_2(\boldsymbol{X}), \ldots, q_m(\boldsymbol{X})\}$ of degree n that vanish on the subspace arrangement. As we have discussed in Section 3, the coefficients $\boldsymbol{C} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_m)$ of the polynomials $\{q_i(\boldsymbol{X}) = \nu_n(\boldsymbol{X})^T \boldsymbol{c}_i\}$ are estimated from the eigenvectors associated with the smallest eigenvalues of the matrix Σ for least-square fitting or the generalized eigenvectors of the matrix $\Gamma^{-1}\Sigma$ (see (3.17)). Regardless of the case, we denote the estimate as $\hat{\boldsymbol{C}}$.

The outliers affect the final results mainly by influencing the eigenvectors $\hat{\boldsymbol{C}}$, and subsequently lead to erroneous estimate of the coefficients of the vanishing polynomials. Therefore, to eliminate the effect of outliers, we seek a *robust* method to estimate the eigenvectors in such a manner that they would be insensitive to the outliers, or to reject the outliers before the eigenvectors are estimated. Such a robust modification applies to all versions of GPCA introduced earlier.

Notice that for our problem, we are not interested in the individual vectors in $\hat{\boldsymbol{C}}$, but rather the eigensubspace spanned by the eigenvectors: $\hat{S} = \mathrm{span}(\hat{\boldsymbol{C}})$. Therefore, the influence of the sample z_i on the estimate of the eigensubspace can be measured

by:

$$I(\boldsymbol{z}_i; S) = \langle \hat{S}, \hat{S}^{(i)} \rangle, \tag{4.3}$$

where $\langle \cdot, \cdot \rangle$ denotes the subspace angle between two subspaces [5], and $\hat{S}^{(i)}$ is the eigensubspace estimated with the ith sample omitted. All samples then can be sorted by their influence values, and the ones with the highest values will be rejected as "outliers" and will not be used for the estimation of the eigensubspace (or the vanishing polynomials).

Equation (4.3) is a precise expression in describing the influence of a sample on the estimation of the vanishing polynomials $Q(\boldsymbol{X})$. However, the complexity of the resulting algorithm is rather high. Suppose we have N samples, then we need to perform PCA $N + 1$ times in order to evaluate the influence values for the N samples. In light of this drawback, some first order approximations of the influence values were developed at roughly the same period as the sample influence function was proposed [8,10], when the computational resource was scarcer than it is today. In robust statistics, formulae that approximate a sample influence function are referred to as *theoretical influence functions*.

While it is rather difficult to approximate the influence of each sample on the estimated subspace S, it is relatively easier to approximate the sample's influence on the (coefficient) vectors $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_m\}$ as the eigenvectors of the sample covariance matrix

$$\Sigma \doteq \frac{1}{N-1} \sum_{i=1}^{N} \nu_n(\boldsymbol{z}_i) \nu_n(\boldsymbol{z}_i)^T. \tag{4.4}$$

The basic idea here is to assume that each $\boldsymbol{c}_j$ is a random vector with a cumulative distribution function (c.d.f) F. The distribution can be perturbed by a change of the weighting $\epsilon \in [0,1]$ of the ith sample:

$$F_i(\epsilon) = (1 - \epsilon)F + \epsilon \delta_i, \tag{4.5}$$

where δ_i indicates the c.d.f. of a random variable that takes the value $\boldsymbol{z}_i$ with probability one. When F becomes $F_i(\epsilon)$, let $\boldsymbol{c}_j(\epsilon)$ be the new estimate of $\boldsymbol{c}_j$ after the change. Now we can define a theoretical influence function $I(\boldsymbol{z}_i; \boldsymbol{c}_j)$ of the ith sample on $\boldsymbol{c}_j$ as the first-order approximation of the above sample influence:

$$\boldsymbol{c}_j(\epsilon) - \boldsymbol{c}_j = I(\boldsymbol{z}_i; \boldsymbol{c}_j)\epsilon + \text{h.o.t.}(\epsilon). \tag{4.6}$$

As derived in [10], the theoretical influence function $I(\boldsymbol{z}_i; \boldsymbol{c}_j) \doteq \lim_{\epsilon \to 0} \frac{\boldsymbol{c}_j(\epsilon) - \boldsymbol{c}_j}{\epsilon}$ is given by

$$I(\boldsymbol{z}_i; \boldsymbol{c}_j) = -z_j \sum_{h \neq j} z_h \boldsymbol{c}_h (\lambda_h - \lambda_j)^{-1} \quad \in \mathbb{R}^{M_n^{[D]}}, \tag{4.7}$$

where $\{\lambda_j, \boldsymbol{c}_j\}$ are the eigenvalues and eigenvectors of the sample covariance matrix Σ and z_h is the hth principal component (PC) of the sample $\boldsymbol{z}_i$, i.e., the coordinate value with respect to the hth eigenvector $\boldsymbol{c}_h$ of the covariance matrix Σ. A further discussion of this solution can be found in [25].

Notice that in order to compute the theoretical influence function (4.7), one needs only to compute once the sample covariance matrix Σ and its eigenvalues and eigenvectors. Thus, computationally, it is much more efficient than the sample influence function.

4.1.2. Robust Covariance Estimators. As we noticed in the estimation of the vanishing polynomials, if we view the vectors $\nu_n(\boldsymbol{z}_i)$ as the random samples, the problem becomes how to estimate robustly the covariance matrix of the random vector $\boldsymbol{u} = \nu_n(\boldsymbol{z})$. It is shown in [14] that, if both the valid samples and the outliers are of zero-mean Gaussian distribution and the covariance matrix of the outliers is a scaled version of that of the valid samples, then the *Mahalanobis* distance:

$$d_i = \boldsymbol{u}_i^T \Sigma^{-1} \boldsymbol{u}_i \tag{4.8}$$

based on the empirical sample covariance $\Sigma = \frac{1}{N-1} \sum_{i=1}^N \boldsymbol{u}_i \boldsymbol{u}_i^T$ is a sufficient statistic for the optimal test that maximizes the probability of correct decision about the outliers (in the class of tests that are invariant under linear transformations). Thus, one can use d_i as a measure to down-weight or discard outlying samples while trying to estimate the correct sample covariance Σ.

Depending on the choice of the down-weighting schemes, many robust covariance estimators have been developed in the literature. Among them, two methods have been widely adopted, namely, the *M-estimator* [23] and *multivariate trimming* (MVT) [17] . A major constraint for robust covariance estimators is the maximal percentage of outliers in a data set that an algorithm can effectively handle. This percentage is called the *breakdown point* [18, 23]. Roughly speaking, for the M-estimator it is inversely proportional to the dimension of the samples, and it usually becomes prohibitive when the data dimension is higher than 20. For MVT, it is equal to the percentage of samples trimmed from the data set, which can be very high. The convergence rate of MVT is also the fastest among all methods of this kind. In the case of subspace arrangements, the dimension of $\boldsymbol{u} = \nu_n(\boldsymbol{z})$, i.e. $M_n^{[D]}$, is normally very high. Thus, the M-estimator becomes impractical and MVT becomes the method of choice.

The MVT method proceeds as follows. As the random vector $\nu_n(\boldsymbol{z})$ is not necessarily zero mean, we first obtain a robust estimate of the mean $\bar{\boldsymbol{u}}$ of the samples $\{\boldsymbol{u}_i = \nu_n(\boldsymbol{z}_i)\}$ (using techniques such as in [17]). We then need to specify a trimming parameter α, which is essentially equivalent to the outlier percentage. To initialize the covariance matrix Σ_0, all samples are sorted by their Euclidean distance $\|\boldsymbol{u}_i - \bar{\boldsymbol{u}}\|$, and Σ_0 is calculated as:

$$\Sigma_0 = \frac{1}{|U| - 1} \sum_{h \in U} (\boldsymbol{u}_h - \bar{\boldsymbol{u}})(\boldsymbol{u}_h - \bar{\boldsymbol{u}})^T, \tag{4.9}$$

where U is the subset of first $100(1 - \alpha)\%$ samples with the smallest distance. In the kth iteration, the Mahalanobis distance of each sample, $(\boldsymbol{u}_i - \bar{\boldsymbol{u}})^T \Sigma_{k-1}^{-1} (\boldsymbol{u}_i - \bar{\boldsymbol{u}})$, is calculated, and Σ_k is again calculated using the set of first $100(1 - \alpha)\%$ samples with the smallest Mahalanobis distance. The iteration terminates when the difference between Σ_{k-1} and Σ_k is small enough.

To proceed with the rest of the GPCA algorithm, we treat the trimmed samples in the final iteration as the outliers, and estimate $Q(\boldsymbol{X})$ from the last m eigenvectors of the resulting covariance matrix.

4.1.3. Performance Comparison. As we have mentioned before, in this subsection, we assume the outlier percentage is known, and the robust methods reject the same amount of samples from a data set. We test and compare the performance of the three robust methods discussed above. Two synthetic data sets are used for the test. Data set one: three subspaces in $\mathbb{R}^3$ of dimensions $2, 2, 1$, with $400, 400, 200$ sample points drawn from the subspaces, respectively. Data set two: four subspaces

in $\mathbb{R}^5$ of dimensions $4, 2, 2, 1$, with $600, 400, 400, 300$ sample points. For each data set, we first add 6% Gaussian noise to all the valid samples, then generate an additional set of outliers with percentages ranging from 0% to 48% of the total sample number, and the rejection rate is set to be the same as the true outlier percentage. At each percentage level, the simulation is repeated for 100 times.

Notice that because of randomness, some outliers may be very close to the subspaces and are not rejected as such; some samples might be rejected as outliers because of large noise. Thus, the segmentation error becomes a less accurate index for the performance of the algorithms. Nevertheless, the average subspace angle difference between the *a priori* model and the estimated one provides a reasonable measure of the change of the subspace structures. Figure 4.2 shows the average subspace angle difference (in degree). Table 4.1 summarizes the average running time of the MAT-LAB codes for one trial on a dual 2.7GHz G5 Macintosh workstation. The reader can download the MATLAB codes from our website.

(a) $(2, 2, 1)$ in $\mathbb{R}^3$ (b) $(4, 2, 2, 1)$ in $\mathbb{R}^5$
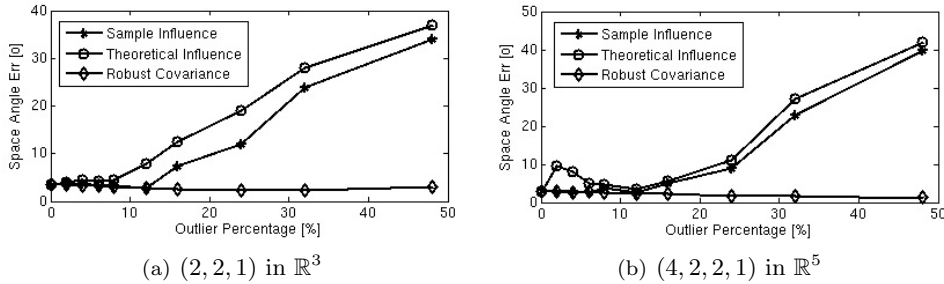
FIG. 4.2. *Average space angle error of GPCA with vanishing polynomials estimated by the three robust methods.*

TABLE 4.1
Average time for solving $(2, 2, 1)$ in $\mathbb{R}^3$ with sample numbers $(400, 400, 200)$.

Outlier Percentage	0%	4%	8%	16%	24%	32%	48%
Data Size	1000	1042	1087	1190	1316	1470	1923
Sample Influence	5.4s	2.5m	2.8m	3.7m	5m	7.8m	18m
Theoretical Influence	5.4s	9s	9.2s	9.3s	9.3s	9.6s	10.8s
MVT	5.4s	5.4s	5.5s	5.6s	5.7s	5.7s	5.8s

To summarize, given a sample rejection rate that is close to the true outlier percentage, the robust covariance estimator turns out to be the most accurate and fastest method for subspace arrangements. In fact, the MVT algorithm usually only takes less than 10 iterations to converge regardless of the outlier percentage, and the estimate does not deteriorate when the outlier percentage increases. On the other hand, the sample influence method and the theoretical influence method give reasonable results only when the outliers are less than 15%. The performances of the two methods are very close, but the sample influence method is significantly slower than the theoretical influence method.

4.2. Estimating the Outlier Percentage. The above techniques did not completely solve the outlier issue, since usually we do not know the outlier percentage, and hence the rejection rate, for a given data set. In this subsection, we propose a

way to estimate the outlier percentage. The percentage will be so determined that the GPCA algorithm returns a "good enough" subspace arrangement model from the remaining sample points. The main idea is to conduct the outlier rejection process multiple times under different rejection rates, and verify the "goodness" of the resulting models. The MVT covariance estimator becomes the natural choice because, besides its speed and accuracy comparing to the two influence function methods, it has the outlier rejection rate built in the algorithm.

We first illustrate the basic ideas with an example. We randomly draw a set of sample points from three subspaces of dimensions $(2, 2, 1)$ in $\mathbb{R}^3$ with sample sizes $(200, 200, 100)$ and add 6% Gaussian noise. Then, the data are contaminated by 16% uniformly distributed outliers. We use MVT to trim out various percentages of samples ranging from 0% to 54%, and compute the maximal residual of the remaining samples with respect to the subspace arrangement given by the GPCA-Voting algorithm. Figure 4.3 shows the plot of the maximal residual versus the rejection rate. The maximal sample residual reaches a plateau right after 7% rejection rate, and the residual decreases when the rejection rate increases. Figure 4.4 shows the segmentation results at rejection rate 7% and 38%, respectively.
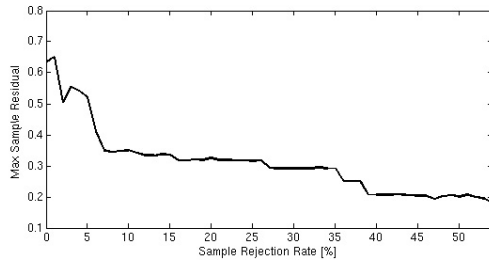
Fig. 4.3. *Maximal sample residuals with various rejection rates on subspace arrangement models estimated using the MVT covariance estimator. The data set is contaminated by* 16% *uniformly distributed outliers.*
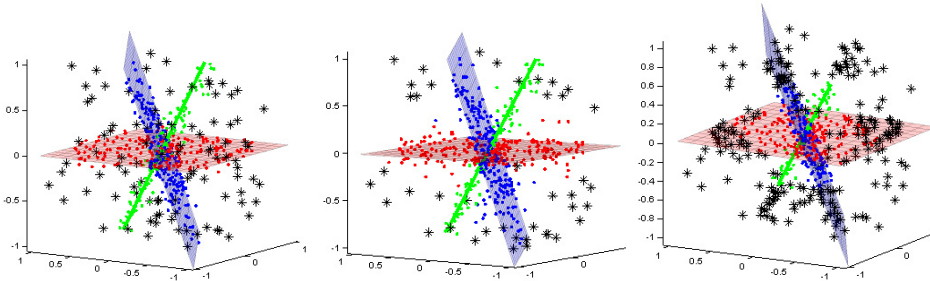

Fig. 4.4. *Subspace segmentation results. Left: a priori data (outliers are marked as black asterisks "∗"). Middle: estimation result with* 7% *samples rejected. Right: estimation result with* 38% *samples rejected.*

In the experiment, although the 7% rejection rate is far less than the *a priori* 16% outlier percentage, the remaining outliers in the sample set are nevertheless close to the subspaces (in terms of their residuals w.r.t. the estimated subspace arrangement), and the resulting subspace arrangement is close to the ground truth. We also see that MVT is moderately stable when the rejection rate is higher than the actual percentage

of outliers. In this case, when the rejection rate is 38%, MVT also trims out inlying samples that have relatively larger noise, which results in even a smaller maximal residual as shown in Figure 4.3.

Therefore, one does not have to reject the *a priori* outlier percentage in order to obtain a good estimate of the arrangement. An estimate of the percentage will be the rejection rate that results in small sample residuals of the remaining sample points. These observations suggest two possible approaches for determining the rejection rate from a plot of the maximal sample residual:

1. The rejection rate can be determined by finding the first "knee point," or equivalently the first "plateau," in the residual plot (in the above example, at 7%).
2. The rejection rate can be determined by a pre-specified maximal residual threshold.

In practice, one may choose to use either approach based on the nature of the application. However, for the first approach, it is commonly agreed in the pattern recognition literature that a method that finds knee points and plateaus in a plot may not be robust if the data are noisy, since they are both related to the first-order derivatives of the residual curve. For example, in Figure 4.3, the rejection rate 3% is arguably a knee point too. In addition, a well-shaped plateau may not exist in the residual plot at all if the *a priori* outlier percentage is small. Two such examples will be shown in Figure 5.2.

In this paper, we propose to determine the outlier percentage as the smallest one such that the maximal sample residual is smaller than a given residual threshold for several consecutive rejection rates (i.e., the residual "stabilizes.") The residual threshold can be seen as the variance of the noise of the valid data. It plays a similar role as the error tolerance when we determine the minimum effective dimension of a noisy sample set (see Section 3.3.3). The choice of a residual threshold also helps us to conduct a fair comparison with other robust statistical techniques, in particular the RANSAC algorithm, later in this section. Algorithm 3 gives an outline of the resulting algorithm, which we refer to as Robust Generalized Principal Component Analysis (RGPCA).

To demonstrate the performance of the algorithm, we conduct three simulated experiments: 1. Three subspaces with dimensions $(2, 2, 1)$ in $\mathbb{R}^3$ and sample sizes $(200, 200, 100)$. 2. Four subspaces with dimensions $(4, 2, 2, 1)$ in $\mathbb{R}^5$ and sample sizes $(400, 200, 200, 100)$. 3. Three subspaces with dimensions $(5, 5, 5)$ in $\mathbb{R}^6$ and sample sizes $(600, 600, 600)$. The maximal data magnitude is 1, and it is corrupted with 6% Gaussian noise and uniformly distributed outliers of a series of percentages between 0% to 48%. The experiment is repeated 100 times at each percentage. In the experiment, the residual threshold is fixed at 0.05. Figure 4.5 shows the results of the average angle error. Table 4.2 shows the average time of the algorithm on a dual 2.7G Hz Macintosh workstation.

TABLE 4.2
Space angle errors and average time for estimating the three subspace arrangements with 20% uniform distributed outliers via RGPCA.

Arrangement	$(2, 2, 1)$ in $\mathbb{R}^3$	$(4, 2, 2, 1)$ in $\mathbb{R}^5$	$(5, 5, 5)$ in $\mathbb{R}^6$
Inlying Sample Size	$(200, 200, 100)$	$(400, 200, 200, 100)$	$(600, 600, 600)$
Angle Error (degree)	1.8	1.7	8
Time Lapse	46s	23m	8m

Algorithm 3 (Robust Generalized Principal Component Analysis).

Given a set of samples $\{z_1, z_2, \ldots, z_N\}$ in $\mathbb{R}^D$, a parameter for angle tolerance τ, and a parameter for residual threshold σ, fit n linear subspaces with codimensions $c_1, c_2, \ldots, c_n$:

1: Set a maximal possible outlier percentage $M\%$.
2: Normalize the data such that the max vector magnitude is 1.
3: **for all** rejection rate $0 \leq r \leq M$ **do**
4:     Remove $r\%$ samples using MVT.
5:     Call GPCA-Voting (Algorithm 2) on the remaining sample points with parameters τ
       and $c_1, c_2, \ldots, c_n$.
6:     Calculate the maximal residual $\sigma_{\max}$ w.r.t. the estimated subspace arrangement.
7:     **if** $\sigma_{\max} < \sigma$ for 3 consecutive rejection rates **then**
8:         Break, and obtain the final estimate.
9:     **end if**
10: **end for**
11: **if** $\sigma_{\max} > \sigma$ **then**
12:     ERROR: the given σ is too small.
13: **else**
14:     Assign samples as inliers if their residuals w.r.t. the model are less than σ.
15: **end if**

(a) $(2, 2, 1)$ in $\mathbb{R}^3$ (b) $(4, 2, 2, 1)$ in $\mathbb{R}^5$ (c) $(5, 5, 5,)$ in $\mathbb{R}^6$
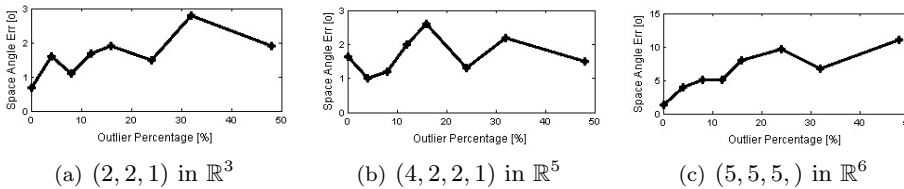
Fig. 4.5. *Average space angle errors (in degree) of arrangements estimated by RGPCA.*

In summary, assuming that the variance of the noise of inliers are somewhat known, the RGPCA algorithm can successfully recover a subspace arrangement from the data set contaminated by a large amount of outliers and noise. We will apply the algorithm to solve some real-world problems in computer vision in Section 5.1.

4.3. Other Robust Statistical Techniques. The above robust techniques have one thing in common: To begin with, they all somewhat rely on an estimate of the model from all the samples. This to some extent puts a limit on the amount of outliers that these techniques can deal with. Depending on the nature of the data and the actual implementation of the algorithm, these techniques, particularly robust covariance estimators, can only handle up to 50% outliers [39, 43]. There is yet another category of robust statistical techniques that are based on *random sampling*, including but not limited to *the least median estimate* (LME) [38], *random sampling consensus* (RANSAC) [15], and *the Hough transform* [2] developed in the computer vision literature. They typically start with certain estimates of the model from randomly drawn subsets of the whole sample set and then select the best one in terms of the resulting residual or consensus for the remaining samples. In principle, these techniques can deal with larger than or equal to 50% outliers.

In the context of subspace segmentation, there are at least three possible ways to estimate an arrangement model using the random sampling scheme. In the following,

we briefly discuss their implementations and difficulties in being applied to subspace arrangements.

1. *Estimating Vanishing Polynomials.* Similar to the previous methods, a random sampling algorithm can be applied to the estimation of a set of polynomials that consistently vanish on a subset of the sample data. In this approach, the number of random samplings becomes prohibitive when the model dimension is high. For instance, in the context of GPCA, assume the dimension of the vector $\nu_n(z)$ is 70 (which corresponds to the case of 4 subspaces in $\mathbb{R}^5$). To estimate a hyperplane in $\mathbb{R}^{70}$ with 20% outliers, in order to have at least one subset of 69 inliers with probability 0.95, on needs to sub-sample over 14 millions subsets, not to mention that we still need to use GPCA to calculate the subspace bases using the resulting polynomials. This drawback makes it impractical to apply random sampling techniques to the estimation of the vanishing polynomials.

2. *Estimating One Subspace at a Time.* One can also consider to apply the random sampling techniques to find one subspace at a time. The number sub-samplings becomes relatively reasonable in this case. For instance, for 4 hyperplanes in $\mathbb{R}^5$ with 20% outliers, suppose the samples are somewhat evenly distributed among the 4 hyperplanes. Then, with respect to each hyperplane, the outliers are actually 80%. In order to get at least one subset of 4 inliers with probability 0.95, we need to sub-sample about 1,900 subsets. However, when an arrangement contains subspaces of different dimensions, this approach leads to many complications that severely limit its performance: 1. If one tries to find higher-dimensional subspaces first, the model will overfit lower-dimensional subspaces or combinations of multiple subspaces, and they are more likely to rank high in the consensus test. 2. If one tries to estimate lower-dimensional subspaces first, subsets from higher-dimensional subspaces or even intersections of these subspaces may likely win out first in the consensus test.

3. *Estimating Mixture Models.* For a mixture model such as a subspace arrangement, another natural sampling scheme is to sub-sample a set of inliers drawn on all subspaces. For instance, to estimate an arrangement model of three subspaces of dimensions $(5, 5, 5)$ in $\mathbb{R}^6$, we sub-sample 15 samples each time, and evenly partition the set into three subsets and estimate three individual subspace models. Although an arrangement model is directly obtained from this approach, unfortunately, the number of samplings is still high even for a relatively small number of subspaces. To estimate the above arrangement model with 600 inliers drawn from each subspace and 20% outliers, one needs to sub-sample over 1.2 billions subsets to have one subset of 15 inliers with probability of 95%. To estimate the arrangement with four subspaces of dimensions $(4, 2, 2, 1)$ and sample sizes $(400, 200, 200, 100)$, respectively, one also needs to sub-sample more than 2 million times with 95% confidence. We have implemented this method in MATLAB, and the results of three simulations are shown in Table 4.3. The numbers of sub-samplings shown in the table are the smallest ones to achieve reasonable segmentation results. Although they are much smaller than the theoretical ones with 95% confidence, the algorithm is significantly slower than RGPCA.

In the computer vision literature, RANSAC has shown good performance for some special subspace arrangements. For instance, if all subspaces are of the same

TABLE 4.3
Space angle errors and average time for estimating the three subspace arrangements with 20 percentage outliers via RANSAC. With similar segmentation performance given in Table 4.2, RANSAC is much more time-consuming.

Arrangement	$(2, 2, 1)$ in $\mathbb{R}^3$	$(4, 2, 2, 1)$ in $\mathbb{R}^5$	$(5, 5, 5)$ in $\mathbb{R}^6$
Inlying Sample Size	$(200, 200, 100)$	$(400, 200, 200, 100)$	$(600, 600, 600)$
Number of Samplings	5,000	1,000,000	10,000,000
Angle Error (degree)	1.5	3.5	10.2
Time Lapse	27s	5 hours	> 2 days

dimension, the second approach has been used to iteratively recover one subspace model at a time [48]. If the subspace dimensions have different dimensions, a *Monte Carlo* scheme can be applied to speed up the third approach for the estimation of a hybrid model [42, 49]. However, none of the special cases seems to generalize well to others. At the time this paper is written, we are not yet aware of any implementation of robust techniques whose performance is competitive against the GPCA/RGPCA methods discussed earlier, even in the range of less than 50% outliers for relatively small number of subspaces (say 4) in a relatively low-dimensional ambient space (say 6). The authors would appreciate any information about any unified implementation of these techniques that works well for general subspace arrangements.

5. Applications. There have been many successful applications of the algebraic GPCA algorithm and its statistical variations in a wide range of research areas, including computer vision, image processing, pattern recognition, and system identification. In this section, we present a couple of representative examples that demonstrate the basic reasons why subspaces arrangements may become the model of choice in many real-world problems.

Roughly speaking, there are two categories of applications in which subspace arrangements have proved useful. In the first category, the given mixed data are known to have a piecewise-linear structure. That is, the data can be partitioned into different subsets such that each subset is drawn from a linear subspace model. Then we can apply GPCA to *extract* such hybrid linear structures. This is the case with motion segmentation in computer vision (see Section 5.1). In the second category, the exact structure of the given data is more complex, but known to be somewhat heterogeneous or even nonlinear. Then we can apply GPCA to find a hybrid linear model that can *approximate* the data. The resulting model provides a compact (lossy) representation of the data as well as a partition of the data into homogeneous subsets. This is the case with sparse image representation in image processing (see Section 5.2).

5.1. Motion Segmentation in Computer Vision. The observed scene of a video sequence typically consists of multiple objects moving independently against the background. Suppose multiple feature points are detected on the objects and the background. These could be either corner points or other local texture patterns that are invariant to camera motions. An important problem in computer vision is how to group these feature points that belong to different moving objects. More precisely, denote by $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N\} \subset \mathbb{R}^3$ a set of points in the three-dimensional scene that are attached either to the moving objects or to the background. Suppose the video sequence contains F frames of images. The image of every $\boldsymbol{X}_j$ in the ith image frame is denoted by $\boldsymbol{z}_{ij} \in \mathbb{R}^2$, a point in the two-dimensional image plane. Then the problem is how to group the images $\boldsymbol{z}_{ij}$ so that for each subset, their corresponding

$\boldsymbol{X}_j$'s belong to the same moving object or the background in the 3-D scene.

Of course, the problem depends on how the 3-D points $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots \boldsymbol{X}_N$ are projected onto the image plane (i.e., the camera model) and what class of motions we consider for $\boldsymbol{X}_j$ or for $\boldsymbol{z}_{ij}$ (i.e., the 3-D or 2-D motion models). Nevertheless, it has been shown that the motion segmentation problem can be converted to a subspace-segmentation problem for most motion models that have been considered in computer vision [52]. Thus, the GPCA algorithm in this paper provides a unified solution to all the possible cases. We present below one of those cases that has some practical importance.

For feature points on one object, the projection can be modeled as an affine camera model[18] from $\mathbb{R}^3$ to $\mathbb{R}^2$:

$$\boldsymbol{z}_{ij} = A_i \boldsymbol{X}_j + b_i \in \mathbb{R}^2 \quad \text{for all } i = 1, 2, \ldots, F, \tag{5.1}$$

where $A_i \in \mathbb{R}^{2 \times 3}$ and $b_i \in \mathbb{R}^2$ are the affine camera parameters for the ith frame. If we stack all the image measurements into a $2F \times N$ matrix W, we obtain

$$W \doteq \begin{bmatrix} \boldsymbol{z}_{11} \cdots \boldsymbol{z}_{1N} \\ \vdots \qquad \vdots \\ \boldsymbol{z}_{F1} \cdots \boldsymbol{z}_{FN} \end{bmatrix}_{2F \times N} = \begin{bmatrix} A_1 & b_1 \\ \vdots & \vdots \\ A_F & b_F \end{bmatrix}_{2F \times 4} \begin{bmatrix} \boldsymbol{X}_1 \cdots \boldsymbol{X}_N \\ 1 \ \cdots \ 1 \end{bmatrix}_{4 \times N}. \tag{5.2}$$

Notice that the product of the two matrices on the right hand side of the equation should result in a matrix of maximum rank 4. It follows that $\text{rank}(W) \leq 4$, hence the 2-D trajectories of the image points across multiple frames, i.e. the columns of W, live in a subspace of $\mathbb{R}^{2F}$ of dimension less than 5.

For multiple moving objects, it can be shown under mild conditions that the trajectories of their image points span different subspaces in $\mathbb{R}^{2F}$. Thus, if we view the columns of W as the sample points, then these sample points belong to multiple subspaces (of dimension less than 5) in $\mathbb{R}^{2F}$. In the computer vision literature, many algorithms have been developed to solve the problem of segmenting the points to their respective subspaces, see [9, 27, 44] and references therein.

We first give the experimental results of GPCA-Voting (Algorithm 2) on two outdoor sequences taken by a moving camera tracking a car moving in front of a parking lot and a building (sequences A and B), and one indoor sequence taken by a moving camera tracking a person moving his head (sequence C), as shown in Figure 5.1. The data for these sequences are borrowed from [28], which consist of *outlier-free* point correspondences in multiple views and are available at the website: `http://www.suri.it.okayama-u.ac.jp/data.html`.

We apply GPCA on the three sequences. Given N feature points in F consecutive frames, we first stack all points into $2F$-dimensional vectors

$$\boldsymbol{z}_j = [\boldsymbol{z}_{1j}^T, \boldsymbol{z}_{2j}^T, \ldots, \boldsymbol{z}_{Fj}^T]^T \in \mathbb{R}^{2F}, \quad j = 1, 2, \ldots, N, \tag{5.3}$$

and project the sample points to a 5-dimensional space by PCA. Then we use Algorithm 2 to segment two hyperplanes of dimension 4 in the 5-dimensional space. For all three cases, the angle tolerance is fixed at 0.3 rad. GPCA-Voting gives a percentage of correct classification of 100.0% for all three sequences as shown in Table 5.1. The table also shows results reported in [28] from other existing *multiframe* algorithms for motion segmentation.

---

[18]A more precise model for conventional cameras is a perspective projection. However, when the objects have a small depth variation relative to their distance to the camera, an affine projection is a good approximation.
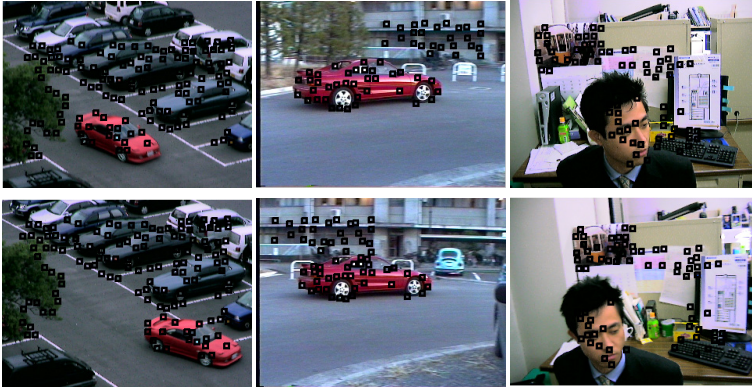
FIG. 5.1. *The first and last frames of sequences A (left), B (middle) and C (right) with point correspondences superimposed.*

TABLE 5.1
Classification rates given by various subspace segmentation algorithms for sequences A, B, C.

Sequence	A	B	C
Number of points	136	63	73
Number of frames	30	17	100
Costeira-Kanade	60.3%	71.3%	58.8%
Ichimura	92.6%	80.1%	68.3%
Kanatani: subspace separation	59.3%	99.5%	98.9%
Kanatani: affine subspace separation	81.8%	99.7%	67.5%
Kanatani: multi-stage optimization	100.0%	100.0%	100.0%
GPCA-Voting	100.0%	100.0%	100.0%

Next, we demonstrate the performance of RGPCA (Algorithm 3) on sequence A and C with original tracking outliers added in, as shown in Figure 5.2. The data are borrowed from [44], which are also available at the website: `http://www.suri.it.okayama-u.ac.jp/e-program-separate.html`. The reported outlier percentages in [44] were 1.4% and 30%, respectively. We use Algorithm 3 to segment two hyperplanes of dimension 4 in both sequences. For both cases, the angle tolerance is fixed at 0.3 rad, and the boundary tolerance is fixed at 0.065. The segmentation results are shown in Figure 5.3. The RGPCA algorithm achieves perfect segmentation with the rejection rate at 0% and 18% for sequence A and C, respectively, which outperforms the results reported in [44].

In sequence A, the camera is far away from the scene, so the projection relation is well described by the affine camera model (5.1), which also results in very small sample residuals in Figure 5.2. The spike in the plot of maximal sample residuals indicates the transition phase when all features on the car are trimmed out by the algorithm.

In sequence C, because the camera is close to the foreground object, the affine camera model does not approximate well the actual camera projection. Furthermore, the motion of the man's upper body is non-rigid, which leads to outliers on the face and shoulders. These observations are consistent with the result of the maximal sample residuals shown in Figure 5.2. We notice that the second plateau in the residual plot
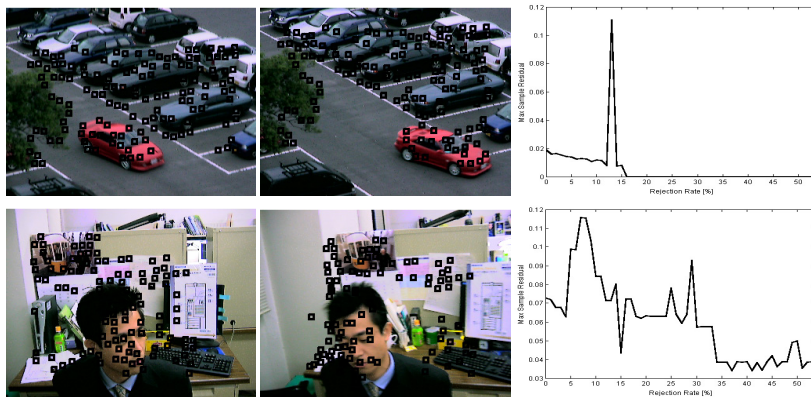
FIG. 5.2. *The first (left) and last (middle) frames of sequences A and C with the original tracking outliers. The right column shows the maximal residual values of the two sequences with various rejection rates using the MVT algorithm. Sequence A contains 140 feature points, and sequence B contains 107 feature points.*
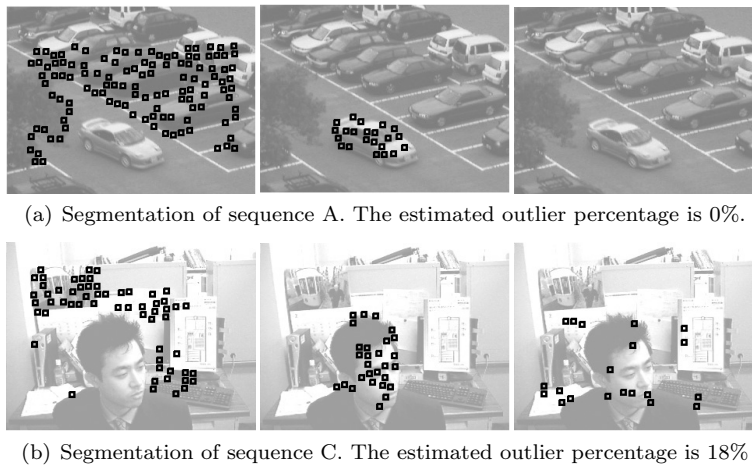

(a) Segmentation of sequence A. The estimated outlier percentage is 0%.

(b) Segmentation of sequence C. The estimated outlier percentage is 18%

FIG. 5.3. *Segmentation results of sequences A and C. Left: group 1. Middle: group 2. Right: outliers.*

indicates that a good segmentation can be achieved at the 30% rejection rate, which conforms to the percentage given in [44]. To make the comparison complete, we show the segmentation result of RGPCA with 30% rejection rate using MVT in Figure 5.4. Although more samples are trimmed as outliers, the algorithm still gives good segmentation on the inlying samples.

5.2. Hybrid Linear Representation of Images. An important problem in image processing is to find efficient and sparse representations of images (rather than the original bitmaps). Such representations are often the first step for many subsequent processes of the images: compression, classification, retrieval, and synthesis, etc. A popular and still dominant approach to represent images is to transform the images via certain linear transformations so that the energy of the image will be concentrated in the coefficients of a sparse set of bases.
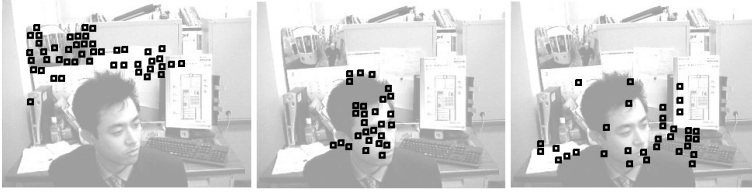
Fig. 5.4. *Segmentation results of C with* 30% *rejection rate. Left: group 1. Middle: group 2. Right: outliers.*

A linear transformation can be either *pre-fixed* for all images (such as the discrete cosine transform used for the JPEG standard and the wavelet transform for the JPEG2000 standard) or *adaptive* for each image (such as the Karhunen-Loève transform that is equivalent to PCA). However, natural images typically exhibit multimodal statistics as they usually contain many heterogeneous regions with significantly different geometric or statistical characteristics, loosely known as "textures." Figure 5.5 shows a typical example. Such heterogeneous data can be better-represented using a mixture of linear models, one for each a homogeneous subset. Figure 5.6 illustrates the basic idea.
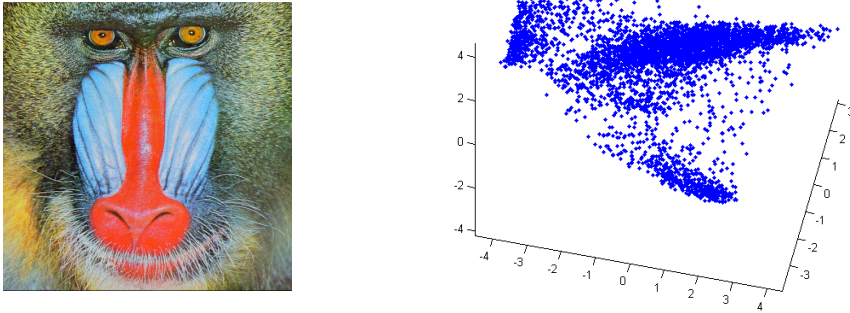
Fig. 5.5. *Left: The baboon image. Right: The coordinates of each dot are the first three principal components of a 2×2 color block of the image (stacked into a vector in $\mathbb{R}^{12}$). There is a visible multi-modal structure in the data.*
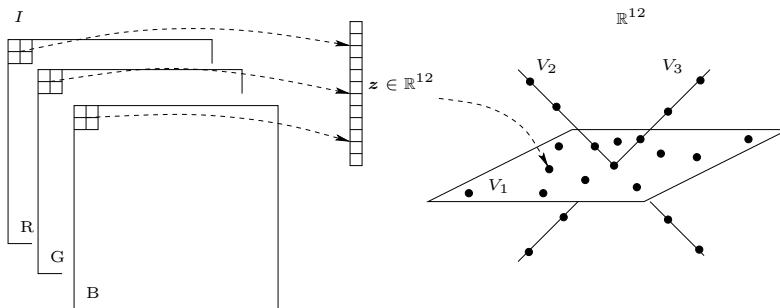

Fig. 5.6. *In a hybrid linear model, vectors $\{z_i\}$ (obtained by stacking the image blocks) are assumed to reside in multiple subspaces which may have different dimensions.*

Obviously, the same assumptions can be made for any transformed image, say

a sub-sampled version of the image and its residuals. Figure 5.7 left shows a three-level representation of the baboon image in terms of a (twice) sub-sampled version and its residuals at two higher level. Figure 5.7 right shows the segmentation of the sub-sampled image and its residuals according to the subspaces of their associated hybrid linear models. Using a slight variation of the GPCA algorithm, the number and dimensions of the subspaces of each hybrid linear model are found automatically in such a way that they minimize the effective dimension of the imagery data subject to a given error threshold. For more details about the algorithmic implementation, the reader may refer to [21]. The source code of the algorithm can be found at: `http://perception.csl.uiuc.edu/gpca`.
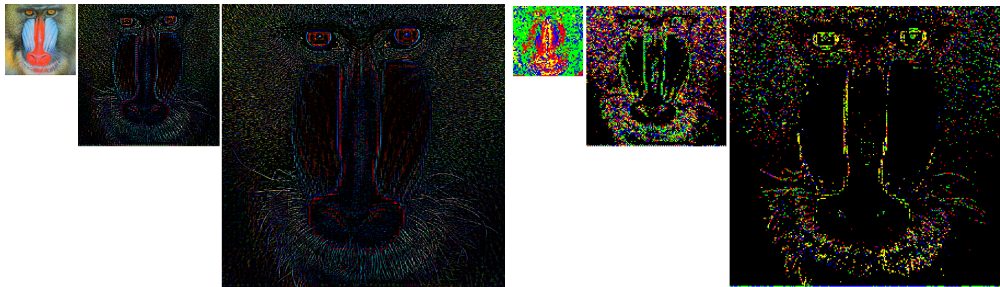

FIG. 5.7. *A multi-scale representation of the Baboon image. Left: Twice sub-sampled image and its residuals at two higher levels. Right: The segmentation of vectors (2×2 blocks) by a hybrid linear model at each level – different subspaces are denoted by different colors. The black regions correspond to data vectors whose energy is below a given error threshold.*

Typically, such a multi-scale scheme can achieve a more compact representation because it extracts low-frequency parts of the image first.[19] Figure 5.8 gives a comparison of the efficiency of different lossy image representation schemes for the baboon image: the discrete cosine transform (DCT), the Karhunen-Loéve transform (KLT), hybrid linear model (without sub-sampling), the level-3 biorthogonal 4.4 wavelets (used in JPEG2000), and the level-3 multi-scale hybrid linear model.[20]

Potentially, there might be many other ways of applying the (multi-scale) hybrid linear model to images that could achieve even better performance. In fact, a higher PSNR curve can be achieved for the baboon image if we apply the multi-scale hybrid linear model in the wavelet domain (see [21]).

5.3. Other Applications. Subspace arrangements have been proven to be pertinent for many other problems that arise in image processing, computer vision, pattern recognition, and system identification. Besides the two applications mentioned above, we list a few more examples and references in the following:

1. *Identification of Hybrid Linear Systems.* It is known from system theory that the input-output data of a linear dynamical system lie on a subspace. Thus, for a hybrid linear system that may switch among multiple linear systems, its input-output data lie a union of multiple subspaces. The problem of identifying the component systems (as well as the switching as a function of time) is essentially a problem of subspace arrangement estimation. The GPCA algorithm has been successfully applied to the identification of hybrid

---

[19]The energy of typical natural images concentrates more in low frequencies.

[20]The experimental results given here are attributed to Wei Hong of the University of Illinois at Urbana-Champaign.
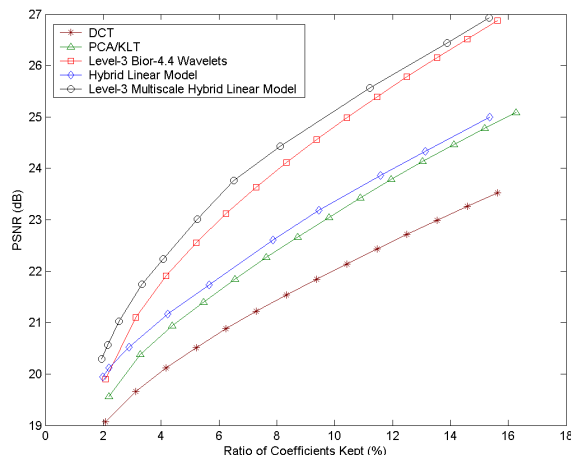
FIG. 5.8. *Comparison of several lossy image representation schemes for the baboon image. Vertical axis: here the signal is the original image and the noise is the difference between the original image and its approximation given by different representation schemes; Horizontal axis: percentage of the ratio of coefficients kept.*

    linear systems such as the hybrid auto-regressive exogeneous (ARX) model [31] and the hybrid auto-regressive moving average (ARMA) model [22].

2. *Classification of Face Images.* It is known that the frontal images of a person's face under different lighting conditions form a low-dimensional subspace [20]. Thus, the problem of clustering face images that belong to different persons can also be characterized as a problem of subspace arrangement estimation. The GPCA algorithm has shown quite success in solving this problem [54]. This approach can be generalized to the classification of other types of images (e.g., hand-written digits).

3. *Segmentation of Video Sequences.* One problem in computer vision is how to partition a long video sequence into multiple short segments such that each segment corresponds to a different scene or event. By viewing each image frame as a sample point, we can fit a piecewise-linear model to the video sequence. Image frames that belong to the same linear piece are naturally grouped together according to their appearance. GPCA has started to become a popular algorithm for video segmentation, both in the spatial and temporal domain [22, 54].

6. Conclusions and Perspectives. This paper introduces a set of new mathematical models – subspace arrangements – for the analysis of multivariate mixed data. Based on the algebraic and statistical properties of subspace arrangements, a set of new computational tools have been developed for the modeling and segmenting mixed data. One important feature of these tools is that they take a "top-down" approach to the estimation of multiple subspaces. That is, the overall algebraic structure of the data set is found first and then the geometric information of the individual subspaces and segmentation of the data are subsequently retrieved. This runs somewhat contrary to the conventional approach taken by existing data clustering methods in statistical learning, such as the EM and the K-means. As a consequence, the result-

ing algorithms, GPCA and its variations, require no initialization and can be used in combination with EM and K-means.

Our exposition also conveys an important message: The confluence of algebra, statistics, and computation is crucial for a complete and thorough understanding of the modeling of mixed data; it is often the source of the inspiration for many of the new algorithms. Within this framework, we will not be surprised that even more effective and general algorithms will be found in the near future.

These new algorithms have shown to be particularly effective in the modeling and segmenting of imagery data, including but not limited to conventional images, videos, biological images, as well as hyper-spectral images. The initial success of these tools in the identification of hybrid systems also suggests that there is good potential in extending them into the dynamical domain.

In many scientific studies, the structure of the data can be modeled as a low-dimensional (nonlinear) manifold embedded in a high-dimensional space. Many algorithms have been proposed to identify such a manifold [40, 46]. GPCA provides yet another class of tools that allow us to obtain a piece-wise linear approximation of the manifold (subject to an error threshold). Important geometric or topical properties, e.g. dimension(s), components, of the manifold can be extracted from such an approximation.

REFERENCES

[1] H. AKAIKE, *A new look at the statistical model selection*, IEEE Transactions on Automatic Control, 16 (1977), pp. 716–723.

[2] D. BALLARD, *Generalizing the Hough transform to detect arbitrary patterns*, Pattern Recognition, 13 (1981), pp. 111–122.

[3] V. BARNETT AND T. LEWIS, *Outliers in Statistical Data*, John Wiley & Sons, second ed., 1983.

[4] P. J. BICKEL, *Another look at robustness: A review of reviews and some new developments*, Scand. J. Statist., 3 (1976), pp. 145–168.

[5] A. BJORCK AND G. GOLUB, *Numerical methods for computing angles between linear subspaces*, Mathematics of Computation, 27 (1973), pp. 579–594.

[6] A. BJÖRNER, *Subspace arrangements*, in First European Congress of Mathematics, Vol. I (Paris, 1992), vol. 119 of Progr. Math., Birkhäuser, Basel, 1994, pp. 321–370.

[7] A. BJÖRNER, I. PEEVA, AND J. SIDMAN, *Subspace arrangements defined by products of linear forms*, Journal of the London Math. Society, (2005), pp. 273–288.

[8] N. CAMPBELL, *The influence function as an aid in outlier detection in discriminant analysis*, Applied Statistics, 27 (1978), pp. 251–258.

[9] J. COSTEIRA AND T. KANADE, *A multibody factorization method of independently moving objects*, International Journal on Computer Vision, 29 (1998), pp. 159–179.

[10] F. CRITCHLEY, *Influence in principal components analysis*, Biometrika, 72 (1985), pp. 627–636.

[11] A. DEMPSTER, N. LAIRD, AND D. RUBIN, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society, 39(B) (1977), pp. 1–38.

[12] HARM DERKSEN, *Hilbert series of subspace arrangements*, 2005 (preprint, math arXiv, math.AC/0510584).

[13] DAVID EISENBUD, *Commutative Algebra with a View Toward Algebraic Geometry*, Springer-Verlag, New York, 1995.

[14] T.S. FERGUSON, *On the rejection of outliers*, in Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, 1961.

[15] M. Fischler and R. Bolles, *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM, 24 (1981), pp. 381–85.

[16] E. Forgy, *Cluster analysis of multivariate data: efficiency vs. interpretability of classifications (abstract)*, Biometrics, 21 (1965), pp. 768–769.

[17] R. Gnanadesikan and J. Kettenring, *Robust estimates, residuals, and outlier detection with multiresponse data*, Biometrics, 28 (1972), pp. 81–124.

[18] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel, *Robust statistics: the approach based on influence functions*, John Wiley & Sons, 1986.

[19] M.H. Hansen and B. Yu, *Model selection and the principle of minimum description length*, Journal of American Statistical Association, 96 (2001), pp. 746–774.

[20] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman, *Clustering appearances of objects under varying illumination conditions*, in Proceedings of International Conference on Computer Vision and Pattern Recognition, 2003.

[21] W. Hong, J. Wright, K. Huang, and Y. Ma, *A multi-scale hybrid linear model for lossy image representation*, in Proceedings of IEEE International Conference on Computer Vision, 2005.

[22] K. Huang, A. Wagner, and Y. Ma, *Identification of hybrid linear time-invariant systems via subspace embedding and segmentation*, in IEEE Conference on Decision and Control, 2004.

[23] P. Huber, *Robust statistics*, John Wiley & Sons, 1981.

[24] R. Jancey, *Multidimensional group analysis*, Austral. J. Botany, 14 (1966), pp. 127–130.

[25] I. Jollife, *Principal Component Analysis*, Springer-Verlag, 2nd ed., 2002.

[26] K. Kanatani, *Model selection for geometric inference*, in Proceedings of Asian Conference on Computer Vision, 2002.

[27] ———, *Motion segmentation by subspace separation: Model selection and reliability evaluation*, International Journal of Image and Graphics, 2 (2003), pp. 179–197.

[28] K. Kanatani and Y. Sugaya, *Multi-stage optimization for multi-body motion segmentation*, in Proceedings of the Australia-Japan Advanced Workshop on Computer Vision, 2003.

[29] S. Lang, *Algebra*, Springer-Verlag, 2002.

[30] S. Lloyd, *Least squares quantization in PCM*, IEEE Transactions on Information Theory, 28 (1982), pp. 129–137.

[31] Y. Ma and R. Vidal, *A closed form solution to the identification of hybrid ARX models via the identification of algebraic varieties*, in International Conference on Hybrid Systems Computation and Control, 2005.

[32] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[33] C.L. Mallows, *Some comments on C_p*, Technometrics, 15 (1973), pp. 661–675.

[34] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, John Wiley & Sons, 1997.

[35] P. Orlik, *Introduction to Arrangements*, vol. 72 of conference board of the mathematical sciences regional conference series in math., American Mathematics Society, 1989.

[36] S. Rao, A. Y. Yang, A. Wagner, and Y. Ma, *Segmentation of hybrid motions via hybrid quadratic surface analysis*, in Proceedings of IEEE International Conference on Computer Vision, 2005.

[37] J.J. Rissanen, *Modeling by shortest data description*, Automatica, 14 (1978), pp. 465–471.

[38] P. Rousseeuw, *Least median of squares regression*, Journal of American Statistics Association, 79 (1984), pp. 871–880.

[39] P. Rousseeuw and A. Leroy, *Robust regression and outlier detection*, John Wiley & Sons, 1987.

[40] S. Roweis and L. K. Saul, *Nonlinear Dimensionality Reduction by Locally Linear Embedding*, SCIENCE, 290 (2000), pp. 2323–2326.

[41] P. D. Sampson, *Fitting conic section to "very scattered" data: An iterative refinement of the bookstein algorithm*, Computer Vision, Graphics and Image Processing, 18 (1982), pp. 97–108.

[42] K. Schindler and D. Suter, *Two-view multibody structure-and-motion with outliers*, in Proceedings of International Conference on Computer Vision and Pattern Recognition, 2005.

[43] C. Stewart, *Robust parameter estimation in computer vision*, SIAM Review, 41 (1999), pp. 513–537.

[44] Y. Sugaya and K. Kanatani, *Outlier removal for motion tracking by subspace separation*, IEICE Transactions on Information and Systems, E86-D (2003), pp. 1095–1102.

[45] G. Taubin, *Estimation of planar curves, surfaces, and nonplanar space curves defined by*

implicit equations with applications to edge and range image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (1991), pp. 1115–1138.

[46] J. Tenenbaum, Vin de Silva, and J. C. Langford, *A Global Geometric Framework for Nonlinear Demensionality Reduction*, SCIENCE, 290 (2000), pp. 2319–2323.

[47] W. Tong, C. Tang, and G. Medioni, *Simultaneous two-view epipolar geometry estimation and motion segmentation by 4D tensor voting*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), pp. 1167–1184.

[48] P. Torr, *Geometric motion segmentation and model selection*, Phil. Trans. Royal Society of London, 356 (1998), pp. 1321–1340.

[49] P. Torr and C. Davidson, *IMPSAC: synthesis of importance sampling and random sample consensus*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25 (2003), pp. 354–364.

[50] P. Torr and D. Murray, *The development and comparison of robust methods for estimating the fundamental matrix*, International Journal on Computer Vision, 24 (1997), pp. 271–300.

[51] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, N.Y., 1995.

[52] R. Vidal and Y. Ma, *A unified algebraic approach to 2-D and 3-D motion segmentation*, in Proceedings of European Conference on Computer Vision, 2004.

[53] R. Vidal, Y. Ma, and J. Piazzi, *A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing Polynomials*, in Proceedings of International Conference on Computer Vision and Pattern Recognition, 2004, pp. 510–517.

[54] R. Vidal, Y. Ma, and S. Sastry, *Generalized principal component analysis (GPCA)*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27 (2005), pp. 1–15.

[55] C.S. Wallace and D.M. Boulton, *An information measure for classification*, The Computer Journal, 11 (1968), pp. 185–194.