SIAM REVIEW
Vol. 50, No. 3, pp. 413–458

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

Yi Ma†
Allen Y. Yang‡
Harm Derksen§
Robert Fossum¶

Abstract. Recently many scientific and engineering applications have involved the challenging task of analyzing large amounts of unsorted high-dimensional data that have very complicated structures. From both geometric and statistical points of view, such unsorted data are considered mixed as different parts of the data have significantly different structures which cannot be described by a single model. In this paper we propose to use subspace arrangements—a union of multiple subspaces—for modeling mixed data: each subspace in the arrangement is used to model just a homogeneous subset of the data. Thus, multiple subspaces together can capture the heterogeneous structures within the data set. In this paper, we give a comprehensive introduction to a new approach for the estimation of subspace arrangements. This is known as generalized principal component analysis (GPCA). 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 by noise or contaminated with 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 several of those representative applications. This paper is intended to be expository in nature. However, in order that this may serve as a more complete reference for both theoreticians and practitioners, we take the liberty of filling in several gaps between the theory and the practice in the existing literature.

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

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

DOI. 10.1137/060655523

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 nature

*Received by the editors March 28, 2006; accepted for publication (in revised form) February 21, 2007; published electronically August 4, 2008. This work was partially supported by UIUC CSE Fellowship and by the grants NSF Career IIS-0347456, NSF CRS-EHS-0509151, NSF CCF-TF-0514955, ONR YIP N00014-05-1-0633, and NSF Career DMS-0349019.

http://www.siam.org/journals/sirev/50-3/65552.html

†Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 145 Coordinated Science Lab., 1308 W. Main St., Urbana, IL 61801 (yima@uiuc.edu).

‡Department of Electrical Engineering and Computer Science, University of California at Berkeley, 333 Cory Hall, Berkeley, CA 94720 (yang@eecs.berkeley.edu).

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

¶Department of Mathematics, University of Illinois at Urbana-Champaign, 1409 W. Green St., Urbana, IL 61801 (rmfossum@uiuc.edu).

of the data and the purpose of the analysis, the model can be either a probabilistic distribution (e.g., a Gaussian distribution or 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 or 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 so in 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 is unlikely to apply to other regions. It is therefore reasonable to use multiple models to describe different regions of the image.

The above example containing images reveals a challenging problem that permeates many research areas such as image processing, computer vision, pattern recognition, and system identification: *How do we 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," "multimodal," "multimodel," "piecewise," "heterogeneous," or "hybrid." For simplicity, in this paper, we refer to the data as "mixed" and the model as "hybrid." Important examples of mixed data that one often encounters nowadays include but are not limited to images, acoustic data, and gene expression data.

Here we are particularly interested in the class of *hybrid linear models*: one linear model for each homogeneous subset of the data. Figure 1.1 shows a simple example. The importance of hybrid linear models is manifold. First, they are the natural generalizations to single linear models; second, they are sufficiently expressive for representing or approximating arbitrary complex data structures; and third, the understanding of hybrid linear models has been significantly advanced in recent years and many efficient solutions have been developed (see [61, 13] and the references therein). Thus, the goal of this paper is to give a comprehensive introduction to some of these new developments in the study of modeling such mixed data with hybrid linear models, and to place 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 together with its parameters were 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) [30]; or, given the three linear subspaces, one can easily segment the samples to their respective closest subspaces. 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 are sometimes also corrupted by noise or outliers, as shown in Figures 1.1(b) and 1.1(c), respectively. 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 $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N\} \subset \mathbb{R}^D$ drawn from a mixture of probabilistic distributions $\{p(\boldsymbol{z}, \theta_j)\}_{j=1}^n$, which is typically a weighted sum $p(\boldsymbol{z}, \Theta) = \sum_j \pi_j p(\boldsymbol{z}, \theta_j)$ with $\sum_j \pi_j = 1$. Then the problem of segmenting mixed data is often converted to a sta-

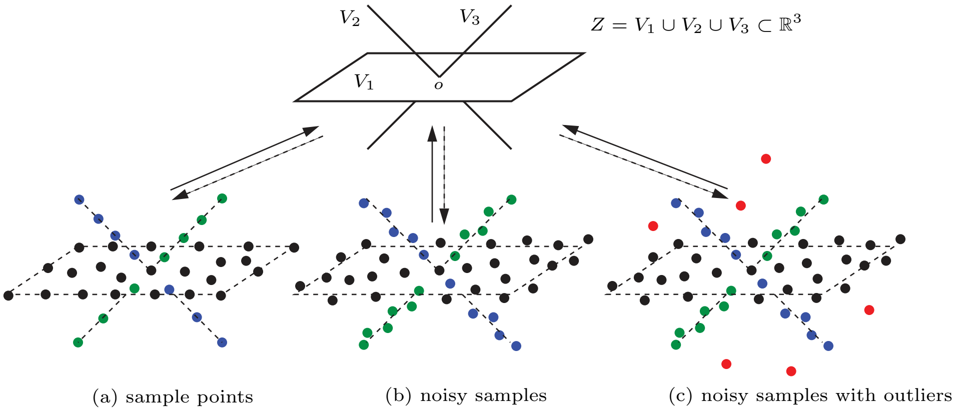

V_2
V_3
$Z = V_1 \cup V_2 \cup V_3 \subset \mathbb{R}^3$
V_1
o
(a) sample points
(b) noisy samples
(c) noisy samples with outliers

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.*

tistical model-estimation problem. Depending on the purpose, the estimated model parameters can take either the maximum-likelihood estimate, which maximizes the log-likelihood, $\max_{\Theta,\pi} \sum_i \log\big(\sum_j \pi_j p(\boldsymbol{z}_i, \theta_j)\big)$, 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 with an expectation of π_j. Then the classical expectation-maximization (EM) algorithm [12, 39] 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 [34, 21, 29, 37] (or its variation for subspaces, K-subspaces [25]), 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 be dealt with, with practical computational complexity, only 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

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

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 [58]. Unfortunately, these situations arise very often for mixed data that are drawn from a typical hybrid (linear) 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 noniterative method not requiring initialization. Although it is unlikely a general solution exists for arbitrary hybrid models, many effective methods have been developed in the past few years for the special 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.*

We consider the problem under three assumptions of 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 noise and contaminated by 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 dimension 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 subspaces to be linear, i.e., they all contain the origin. When a hybrid model consists of affine subspaces that do not contain the origin, we can always increase the dimension of the ambient space by one and identify each affine subspace with the linear subspace that it spans.[2]

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. See [7, 6, 41] for a general review.

[2] As an example, if we are to estimate two affine line models in $\mathbb{R}^2$, we can lift the sample points from the two lines into $\mathbb{R}^3$ by adding a nonzero constant as the third coordinate. Then the problem is converted to estimating two linear planes in $\mathbb{R}^3$.

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 a class of models for describing mixed data, subspace arrangements are sufficiently flexible and expressive: they may contain subspaces of different dimensions, and they can approximate with arbitrary accuracy any nonlinear geometric or topological structures. In fact, subspace arrangements are implicitly assumed in another important area of statistical signal processing, *sparse representation* [40, 14, 16, 15, 17], because the set of all signals that allow a sparse solution w.r.t. a (possibly over-complete) basis precisely lies on multiple low-dimensional linear subspaces. However, in our study, we do not assume that the bases of the subspaces are given in advance and so they are part of the unknowns that need to be retrieved from the data. Nevertheless, as we will see shortly, a subspace arrangement, as an algebraic set, can be effectively estimated and segmented given a sufficient set of sample data.

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 is rather broad, ranging 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 one another 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 algebraic. Section 2 reviews the basic algebraic properties of subspace arrangements. As an algebraic set, the set of polynomials that vanish on a subspace arrangement forms 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 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 deduced from the vanishing polynomials. These results lead to a simple algebraic algorithm for estimating and segmenting a subspace arrangement from a set of sample points, 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, in section 5, we demonstrate how these methods can be applied to several real-world applications. The source codes for all the algorithms in this paper, as well as more applications, are available online at http://perception.csl.uiuc.edu/gpca.

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 to subsequently decompose the arrangement into separate subspaces, as shown in Figure 2.1. The

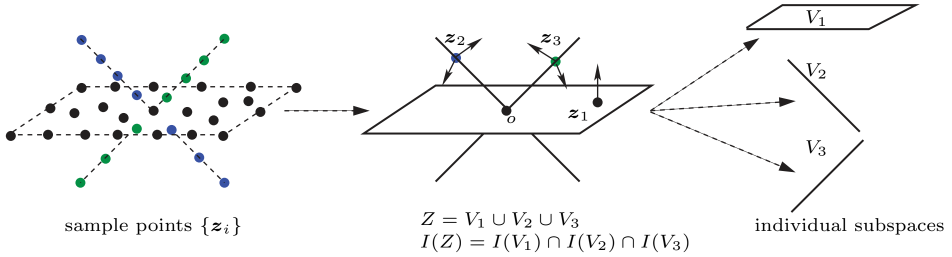


Fig. 2.1 *Inferring a subspace arrangement of three subspaces, $Z = V_1 \cup V_2 \cup V_3$, from a set of sample points $\{z_i\}$.*

algebraic facts presented in this section serve as the theoretical foundation for an effective method to model and segment mixed data known as GPCA. In the sections that follow, we will show how this algebraic method should be modified when the samples are corrupted by noise (section 3) or contaminated with outliers (section 4).

2.1. Basic Definitions and Algorithm.

In this section, we assume that the reader has basic knowledge of the abstract algebra that is covered in any graduate-level algebra course. Details may be found in most texts, for example, [33, 18]. 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 denoted by $c \doteq D - d$.

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 $V_1, V_2, \dots, V_n$ of $\mathbb{F}^D$.

For a nonempty subset S of the index set $\{1, 2, \dots, n\}$, 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 $\mathcal{A} = V_1 \cup V_2 \cup \cdots \cup V_n$ is called* transversal *if*

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

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 roundoff should guarantee that the subspace structures of the data are transversal. Thus, in this paper, unless stated otherwise, we always assume that a subspace arrangement is transversal.

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

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

This is the ring of polynomials in the functions $\boldsymbol{X} \doteq \{X_1, X_2, \ldots, X_D\}$, 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 homogeneous polynomials 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

$$\{X_1^{u_1} X_2^{u_2} \cdots X_D^{u_D} : u_1 + u_2 + \cdots + u_D = h\}$$

form a basis of $\mathbb{F}_h^{[D]}$. Given any point $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)^T \in \mathbb{F}^D$, the values of these monomials give the image of the point under the Veronese map.

Definition 2.3 (Veronese map). *The* Veronese map *of degree h is the map*

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

given by

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

An arbitrary homogeneous polynomial $q(\boldsymbol{X})$ of degree h in $\boldsymbol{X} = \{X_1, X_2, \ldots, X_D\}$ can be written as $q(\boldsymbol{X}) = \boldsymbol{c}^T v_h(\boldsymbol{X})$ for some vector $\boldsymbol{c} \in \mathbb{F}^{M_h^{[D]}}$ that collects all the coefficients associated with the monomials.

Definition 2.4 (Jacobian matrix). *The* Jacobian matrix *of a collection of 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*

$$\mathcal{J}(Q)(\boldsymbol{X}) \doteq \begin{pmatrix} \frac{\partial q_1}{\partial X_1} & \cdots & \frac{\partial q_1}{\partial X_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial q_m}{\partial X_1} & \cdots & \frac{\partial q_m}{\partial X_D} \end{pmatrix} \in \mathbb{R}^{m \times D}. \tag{2.5}$$

With the above definitions, we are now ready to present a basic algebraic algorithm in Algorithm 1, called GPCA, which provides a simple solution to Problem 1.1 under Assumption 1.

From section 2.3 to 2.4, we provide a rigorous justification for and a detailed analysis of this algorithm. Steps 1–3 of the algorithm aim to find the set of polynomials

Algorithm 1. *GPCA.*

```
Given a set of samples $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N\}$ from a (transversal) arrangement of $n$ linear subspaces with dimensions $d_1, d_2, \dots, d_n$ in $\mathbb{R}^D$:
1: Construct the matrix $L_n = \big(\nu_n(\boldsymbol{z}_1), \nu_n(\boldsymbol{z}_2), \dots, \nu_n(\boldsymbol{z}_N)\big)$.
2: Compute the singular value decomposition (SVD) of $L_n$ and let $\boldsymbol{C}$ be the matrix whose columns are the singular vectors associated with all the zero singular values.
3: Construct the polynomials $Q(\boldsymbol{X}) = \boldsymbol{C}^T \nu_n(\boldsymbol{X})$.
4: for all $1 \le i \le n$ do
5:    Pick one point $\boldsymbol{z}_i$ per subspace $V_i$ and compute the Jacobian $\mathcal{J}(Q)(\boldsymbol{z}_i)$.
6:    Compute a basis $B_i = \big(\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_{d_i}\big)$ of $V_i$ from the right null space of $\mathcal{J}(Q)(\boldsymbol{z}_i)$ via the singular value decomposition of $\mathcal{J}(Q)(\boldsymbol{z}_i)$.
7:    Assign samples $\boldsymbol{z}_j$ that satisfy $B_i^T \boldsymbol{z}_j = 0$ to the subspace $V_i$.
8: end for
```

$Q(\boldsymbol{X})$ (of degree n) that vanish on the subspace arrangement of interest. This is possible for two reasons to be elaborated on in section 2.2. First, the subspace arrangement as an algebraic set is uniquely determined by its vanishing polynomials (according to Lemma 2.8); second, the vanishing polynomials can be determined from a finite set of sample points on the subspaces (according to Theorem 2.9). Notice in step 2 of the algorithm that the coefficient vectors of the polynomials $Q(\boldsymbol{X})$ are computed as the singular vectors in the null space of matrix L_n, which can be sensitive to noise. Thus, it is of great practical importance to know the number of linearly independent vanishing polynomials of degree n, since we can take columns of $\boldsymbol{C}$ to be the singular vectors associated with the same number of smallest (not necessarily zero) singular values. The number is given by the Hilbert function of the subspace arrangement, for which we will give a closed formula in section 2.3 (see Corollary 2.16). As we will see in section 2.4, through the relationships between the vanishing ideal and the product ideal of a subspace arrangement revealed in the study of their Hilbert functions, one can easily show that the derivatives of the vanishing polynomials span the orthogonal complement to each subspace, on which steps 5–7 of the algorithm rely.

2.2. Vanishing Polynomials of Subspace Arrangements.

We will discuss the correspondence between ideals in the polynomial ring $\mathbb{F}^{[D]}$ and subsets in $\mathbb{F}^D$.

Definition 2.5 (vanishing ideal). *The* vanishing ideal $I(W)$ *of a subset* $W \subseteq \mathbb{F}^D$ *is defined by*

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

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

$$b_1 x_1 + b_2 x_2 + \cdots + b_D x_D = 0 \quad \text{for all} \quad (x_1, x_2, \dots, x_D)^T \in V. \tag{2.6}$$

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

$$V^{\perp} = \operatorname{span}\{g_1, g_2, \dots, g_c\}, \tag{2.7}$$

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

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

DEFINITION 2.6 (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 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,

$$I(\mathcal{A}) = I(V_1 \cup V_2 \cup \cdots \cup V_n) = I(V_1) \cap I(V_2) \cap \cdots \cap I(V_n). \tag{2.8}$$

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

$$I(\mathcal{A}) = I_m \oplus I_{m+1} \oplus \cdots \oplus I_n \oplus I_{n+1} \oplus \cdots. \tag{2.9}$$

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 arrangement 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

$$J(\mathcal{A}) = J_n \oplus J_{n+1} \oplus \cdots. \tag{2.10}$$

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

$$J_n = V_1^{\perp} V_2^{\perp} \cdots V_n^{\perp} = I_1(V_1) I_1(V_2) \cdots I_1(V_n). \tag{2.11}$$

DEFINITION 2.7 (zero set). *Given a set of polynomials $I \subseteq \mathbb{F}^{[D]}$, the* zero set *of I is defined to be*

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

LEMMA 2.8. *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

$$\mathcal{A} \subseteq Z(I(\mathcal{A})) \subseteq Z(I_n) \subseteq Z(J_n). \tag{2.12}$$

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 $g(\boldsymbol{z}) \neq 0$ and obviously $g \in J_n$. It then follows that $\boldsymbol{z} \notin Z(J_n)$. Therefore $Z(J_n) \subseteq \mathcal{A}$. Using (2.12), 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. □

A consequence of Lemma 2.8 is 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 $\mathcal{A}$. A subset in $\mathbb{F}^D$ is called an *algebraic set* if it is the zero set of its vanishing ideal. In other words, W is an algebraic set if and only if $W = Z(I(W))$. In this sense, a subspace arrangement is an algebraic set. There is a one-to-one correspondence between a subspace arrangement and its vanishing ideal.

Be aware that our task here is to recover the subspace arrangement $\mathcal{A}$ from only a finite number of samples $F = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$. From an algebraic point of view, an algebraic set Z, such as a subspace arrangement, is rather different from any 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 $\boldsymbol{z}_i$; then the set of polynomials that vanish on the finite set $F = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$ is the intersection

$$I(F) = \mathfrak{m}_1 \cap \mathfrak{m}_2 \cap \cdots \cap \mathfrak{m}_N. \tag{2.13}$$

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.

Thus, we need to find the vanishing ideal $I(Z)$ from a finite sample set $F \subset Z$. 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

$$\mathbb{F}^{[D]}_{\leq n} = \mathbb{F} \oplus \mathbb{F}^{[D]}_1 \oplus \cdots \oplus \mathbb{F}^{[D]}_n. \tag{2.14}$$

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

THEOREM 2.9 (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 $F = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$ 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 $\boldsymbol{z}_1 \in Z$. Then $1(\boldsymbol{z}_1) \neq 0$ (1 is the function that assigns 1 to every point of Z). Let $\mathfrak{a}_1$ be the ideal that vanishes on $\{\boldsymbol{z}_1\}$ and define $\mathfrak{b}_1 = \mathfrak{a}_1 \cap \mathbb{F}^{[D]}_{\leq n}$. Further, let $A_1 = (\mathfrak{b}_1)$.[3]

[3] 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 [18].

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 $F_N = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N\} \subset Z$ with

$$I(F_N) = \mathfrak{a}_N, \tag{2.15}$$

$$\mathfrak{b}_N = \mathfrak{a}_N \cap \mathbb{F}^{[D]}_{\leq n}, \tag{2.16}$$

$$A_N = (\mathfrak{b}_N), \tag{2.17}$$

$$\mathfrak{b}_0 \supset \mathfrak{b}_1 \supset \cdots \supset \mathfrak{b}_N \supseteq I_n. \tag{2.18}$$

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 $\boldsymbol{z}_{N+1} \in Z$ for which $g(\boldsymbol{z}_{N+1}) \neq 0$. Set $F_{N+1} = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_N, \boldsymbol{z}_{N+1}\}$. Then one gets $\mathfrak{a}_{N+1}, \mathfrak{b}_{N+1}, A_{N+1}$ as above with

$$\mathfrak{b}_0 \supset \mathfrak{b}_1 \supset \cdots \supset \mathfrak{b}_N \supset \mathfrak{b}_{N+1} \supseteq I_n. \tag{2.19}$$

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. □

Example 2.10 (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 $I(P) = \left(\mathfrak{a}_2 \cap \mathbb{R}^{[3]}_{\leq 1}\right)$.

We point out that no precise lower 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. Thus in principle, with probability 1, the vanishing ideal can be determined from a randomly chosen and sufficiently large set of samples. Moreover, if we know the dimension of I_n and decide to estimate coefficients of the vanishing polynomials linearly, then the smallest number of samples needed is the codimension of I_n in $\mathbb{F}^{[D]}_n$. The number of linearly independent polynomials in I_n is associated with the Hilbert function of the ideal, for which we now derive a closed-form formula.

2.3. Hilbert Functions of Subspace Arrangements.

As we alluded to earlier, in the GPCA algorithm, to more stably estimate the vanishing polynomials (of degree n) of a subspace arrangement $\mathcal{A}$, it is useful to know how many of the polynomials are linearly independent. This is related to the Hilbert function of the vanishing ideal $I(\mathcal{A})$.

DEFINITION 2.11 (Hilbert function). *The* Hilbert function *of a homogeneous ideal K is the function $h_K : \mathbb{N} \to \mathbb{N}$ defined by*

$$h_K(j) \doteq \dim(K_j), \tag{2.20}$$

where K_i is the ith homogeneous component of K and $\mathbb{N}$ denotes the nonnegative integers.[4]

The remainder of this section is devoted to providing a closed-form formula for the Hilbert function $h_I(i)$ of the vanishing ideal $I(\mathcal{A})$ of a subspace arrangement $\mathcal{A}$

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

that is valid for $i \geq n$. The basic idea is to show that the product ideal $J = J(\mathcal{A})$ of $\mathcal{A}$ has a closed-form formula for its Hilbert function $h_J(i) \doteq \dim(J_i)$. Then one can show that when the arrangement is transversal, one has $h_I(i) = h_J(i)$ for all $i \geq n$. A more complete development is given in [13].

Definition 2.12 (Hilbert series). *The* Hilbert series *of a homogeneous ideal K is defined to be*

$$\mathcal{H}(K,t) \doteq \sum_{i \in \mathbb{N}} h_K(i) t^i. \tag{2.21}$$

Example 2.13. The Hilbert series of the polynomial ring $\mathbb{F}^{[D]}$ is

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

Example 2.14. 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 [13]. 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 that 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 of 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 [13]). 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 all transversal. For transversal arrangements, the Hilbert series of the product ideal $J(\mathcal{A})$ has a particularly nice form, which we 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

$$F_{\mathcal{A}}(t) \doteq \frac{\prod_{i=1}^{n}(1-(1-t)^{c_i})}{(1-t)^D}. \tag{2.22}$$

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 2.15 (see [13]). *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 (2.22) 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 2.16. *If $\mathcal{A} = V_1 \cup V_2 \cup \cdots \cup V_n$ is transversal, then for all $i \geq n$,*

$$h_I(i) = h_J(i) = g(i) = \sum_{S} (-1)^{|S|} \binom{i+D-1-c_S}{D-1-c_S}, \tag{2.23}$$

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 reader needs to be aware that formula (2.23) for $h_I(i)$ is valid only for $i \geq n$. The formula in Corollary 2.16 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 $g(i)$, $i = n, n+1, n+2, \ldots$.

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.[5]

Example 2.17. 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 (respectively, codimensions)

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

of V_1, V_2, V_3. We can construct the following 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 from the above table, the codimensions c_1, c_2, c_3 are almost determined by $h_I(3)$. They are uniquely determined by $h_I(3)$ and $h_I(4)$. Corollary 2.18 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 2.18. *Consider a transversal arrangement of n subspaces. The codimensions $c_1, \dots, c_n$ (and hence the dimensions) of the subspaces are uniquely determined by the values of the Hilbert function $h_I(i)$ for $i = n, n+1, \dots, n+D-1$.*

These results are very important for the development and improvement of the GPCA algorithm for estimating and segmenting a subspace arrangement given a set of sample data.

First, 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 one hand, given a data set, if we know the number of subspaces and their dimensions (which can be the case for many practical problems), the value of the Hilbert function from (2.23) 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 determine from the rank of the matrix L_n (in Algorithm 1). 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 according to Corollary 2.18, the dimensions (or number) of the subspaces can be uniquely determined from the values of the Hilbert function (even without segmenting the data first).

Second, 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 [60] because the algorithm would be much easier to explain (to engineers) by using products of linear forms. In the next section, we will see that this property makes it extremely easy to show that the derivatives of the vanishing polynomials span the entire orthogonal complement of each subspace.

2.4. Computational Issues. In the previous subsections we considered the correspondence between a transversal subspace arrangement $\mathcal{A}$ and its vanishing ideal. We also showed how we are able, in principle, to recover the ideal from a large enough number of samples on the arrangement (Theorem 2.9). In this subsection, based on the facts established so far, we discuss a few computational issues associated with the algebraic GPCA algorithm given in section 2.1. 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 [60]. Several different variations have been proposed since then. All variants consist of three main 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 main step.

2.4.1. Retrieving the Vanishing Polynomials. We are given a set of samples $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N\}$ 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

$$L_n \doteq \big(\nu_n(\boldsymbol{z}_1), \nu_n(\boldsymbol{z}_2), \dots, \nu_n(\boldsymbol{z}_N)\big) \quad \in \mathbb{R}^{M_n^{[D]} \times N}. \tag{2.24}$$

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 $i = 1, 2, \dots, N$. 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 Theorem 2.9, the dimension of the null space of L_n is exactly $m = h_I(n)$. Thus, a basis $\boldsymbol{C} = \big(\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_m\big)$ of the null space of L_n gives a basis of $I_n(\mathcal{A})$,

$$Q(\boldsymbol{X}) \doteq \big(q_1(\boldsymbol{X}), q_2(\boldsymbol{X}), \dots, q_m(\boldsymbol{X})\big)^T, \tag{2.25}$$

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

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 roundoff errors, we can take the eigenvectors associated with the $m = h_I(n)$ smallest eigenvalues. Numerically, this can be done via singular value decomposition (SVD) of L_n, which statistically corresponds to principal component analysis (PCA). In section 3, we will see how the estimate of $\boldsymbol{C}$ can be further improved when the samples are noisy.

Computationally, computing the eigenvectors of W is the most expensive step of the entire GPCA algorithm. With the best numerical implementation of SVD, the complexity of the GPCA algorithm is typically *quadratic* in the size of the data N or the dimension of the Veronese map $M_n^{[D]}$. Since $M_n^{[D]}$ grows exponentially in both D and n, on a typical PC, due to memory limits, the GPCA algorithm can only handle $n \leq 10$ subspaces of dimension $D \leq 15$.

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 most practical problems where a subspace arrangement is of interest, we are more interested in the individual subspaces of the arrangement rather than the 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. However, it is computationally prohibitive to directly decompose the vanishing polynomials to obtain the subspaces as "factors" of the polynomials via algebraic means.[6] 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 constituent subspaces in the arrangement.

Let $q(\boldsymbol{X})$ be any polynomial of degree n that vanishes on the arrangement $\mathcal{A}$. Then, according to Theorem 2.15, $q(\boldsymbol{X}) \in J_n(\mathcal{A})$. In other words, it can be written as the sum of products of linear forms,

$$(\boldsymbol{b}_1\boldsymbol{X}) \cdot (\boldsymbol{b}_2\boldsymbol{X}) \cdots (\boldsymbol{b}_n\boldsymbol{X}), \tag{2.26}$$

where $\boldsymbol{b}_i$ is a vector orthogonal to the subspace V_i. Pick one sample $\boldsymbol{z}_i$ per subspace V_i (not in any of the other subspaces).[7] As $\boldsymbol{b}_i\boldsymbol{z}_i = 0$, the gradient of each product of linear forms evaluated at $\boldsymbol{z}_i$ is

$$\boldsymbol{b}_i \cdot (\boldsymbol{b}_1\boldsymbol{z}_i) \cdots (\boldsymbol{b}_{i-1}\boldsymbol{z}_i) \cdot (\boldsymbol{b}_{i+1}\boldsymbol{z}_i) \cdots (\boldsymbol{b}_n\boldsymbol{z}_i).$$

Thus, the gradient of the polynomial $q(\boldsymbol{X})$ evaluated at $\boldsymbol{z}_i$ is spanned by $\boldsymbol{b}_i \in V_i^{\perp}$. On the other hand, any polynomial of the above product form is in $J_n(\mathcal{A}) = I_n(\mathcal{A})$. That is, it can be written as a linear combination of the polynomials in $Q(\boldsymbol{X})$.

Therefore, the rows of the Jacobian matrix $\mathcal{J}(Q)(\boldsymbol{z}_i)$ evaluated at $\boldsymbol{z}_i$ span the entire 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 $\boldsymbol{C}$ from L_n.

2.4.3. Variations of the Basic GPCA Algorithm.

The basic GPCA 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 is 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 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.

[6] This is a problem that does not yet have a polynomial-complexity algorithm.

[7] 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.

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 [52] with a 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-hypothesis 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 $Q(\boldsymbol{X}) = \big(q_1(\boldsymbol{X}), q_2(\boldsymbol{X}), \dots, q_m(\boldsymbol{X})\big)^T$, say, of degree n, whose zero set is exactly the subspace arrangement

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

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 a Gaussian distribution. In practice, it is often more natural and meaningful to assume instead that the samples $\boldsymbol{z}_i$ themselves are corrupted by (isotropic) Gaussian noise. That is, we assume that, for each sample point $\boldsymbol{z}_i$,

$$\boldsymbol{z}_i = \hat{\boldsymbol{z}}_i(\boldsymbol{c}) + \boldsymbol{n}_i, \quad i = 1, 2, \dots, N, \tag{3.3}$$

where $\hat{\boldsymbol{z}}_i(\boldsymbol{c})$ is a point on the subspace arrangement determined by $\boldsymbol{c}$ and $\boldsymbol{n}_i$ is an independent isotropic Gaussian random noise added to $\hat{\boldsymbol{z}}_i(\boldsymbol{c})$. If the arrangement is clearly indicated from the context, we also write $\hat{\boldsymbol{z}}_i(\boldsymbol{c})$ as $\hat{\boldsymbol{z}}$. It is easy to verify that, with respect to this noise model, the embedded data vector $\nu_n(\boldsymbol{z}_i)$ 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*:

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

However, it is difficult to minimize (3.4) because the closest point $\hat{\boldsymbol{z}}_i(\boldsymbol{c})$ to $\boldsymbol{z}_i$ is a complicated function of the polynomial coefficients $\boldsymbol{c}$. To resolve this difficulty, in practice we often use the first order approximation of $\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i$ 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 $Q(\boldsymbol{X})$ are linearly independent. Given a point $\boldsymbol{z}$ close to the zero set of $Q(\boldsymbol{X})$, i.e., the subspace arrangement $\mathcal{A}$, we let $\hat{\boldsymbol{z}}$ denote the point closest to $\boldsymbol{z}$ on $\mathcal{A}$. Using the Taylor series of $Q(\boldsymbol{X})$ expanded at $\boldsymbol{z}$, the value of $Q(\boldsymbol{X})$ at $\hat{\boldsymbol{z}}$ is then given by

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

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

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

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

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

The expression on the right-hand side is known as the *Sampson distance* [47]. Thus, the average Sampson distance

$$\frac{1}{N}\sum_{i=1}^{N} Q(\boldsymbol{z}_i)^T \big(\mathcal{J}(Q)(\boldsymbol{z}_i)\mathcal{J}(Q)(\boldsymbol{z}_i)^T\big)^\dagger Q(\boldsymbol{z}_i) \tag{3.8}$$

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 $Q(\boldsymbol{X})$, it is also the zero set of the polynomials $\tilde{Q}(\boldsymbol{X}) = MQ(\boldsymbol{X})$ for any nonsingular matrix $M \in \mathbb{R}^{m\times m}$. It is easy to check that the Sampson distance (3.7) is invariant under the nonsingular 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 following matrix 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 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):

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

where $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ are the m smallest generalized eigenvalues of (Σ, Γ). Furthermore, as Γ is nonsingular,[8] $\boldsymbol{c}_i^*$ is also the eigenvector associated with the ith smallest eigenvalue of the matrix $\Gamma^{-1}\Sigma$:

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

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 intraclass 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 interclass distance—the norm of the derivative at a point in a subspace is roughly proportional to its distance to other subspaces.[9] 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 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.[10] 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).

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

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

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

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

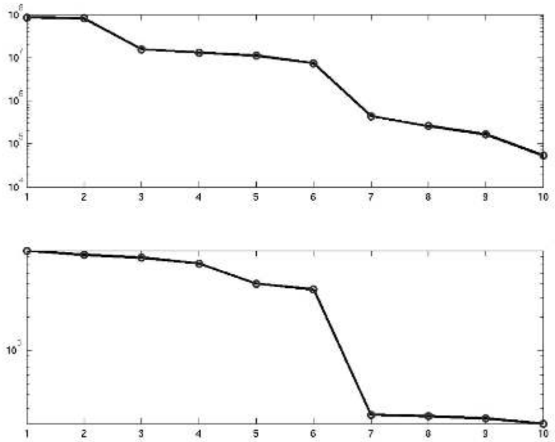


Fig. 3.1 *Top: plot of the eigenvalues of the matrix* Σ. *Bottom: plot of the eigenvalues of the matrix* $\Gamma^{-1}\Sigma$.

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 indicates 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 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.

In the following, we review a variation of algebraic GPCA that improves the estimation of the subspace bases in the presence of high data noise. This method was inspired by the classical Hough transform [2, 54], which collectively considers the derivatives at *all* the sample points. In this scheme, these sample points cast votes on the feature space of subspace basis parameters. More technical details can be found in [63].

3.2.1. GPCA with Voting.

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 $c_1, c_2, \ldots, c_n$. Without loss of generality, we assume that $c_1, c_2, \ldots, c_n$ have l distinct values $c'_1 < c'_2 < \cdots < c'_l$.

Pick a sample point $\boldsymbol{z}_1$. The Jacobian of the fitting polynomials $Q(\boldsymbol{X})$ at $\boldsymbol{z}_1$ is $\mathcal{J}(Q)(\boldsymbol{z}_1)$. 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 that numerically $B_i(\boldsymbol{z}_k)$ cannot be exactly equal to $U_i(j)$. 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.[11] 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, $B_i(\boldsymbol{z}_k)$ then belongs to the candidate basis $U_i(j)$ in the record.

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 1;
   (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.

We summarize the overall process as Algorithm 2, which is called GPCA-voting.

There are important features of the above voting scheme that are quite different from the well-known statistical learning methods K-subspaces [25] and EM [39] 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 sample 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 that 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.

3.2.2. 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, GPCA-voting, as well as some combination of them.

We randomly generate subspace arrangements of some prechosen dimensions. For instance, $(2, 2, 1)$ indicates an arrangement of three subspaces of dimensions $2, 2, 1$,

[11] Please refer to [5] for numerical implementations of computing subspace angles. In MATLAB, the built-in command is `subspace`.

Algorithm 2. *GPCA-voting.*

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

```
Suppose there are $l$ distinct codimensions, ordered as $c'_1 < c'_2 < \cdots < c'_l$. Allocate
  $u_1, u_2, \dots, u_l$ to be $l$ stacks of counters and $U_1, U_2, \dots, U_l$ to be $l$ stacks of candidate bases.
Estimate the set of fitting polynomials $Q(\boldsymbol{X})$, and compute their derivatives $\mathcal{J}(Q)(\boldsymbol{X})$
  for all $\boldsymbol{z}_k$.
for all samples $\boldsymbol{z}_k$ do
    for all $1 \leq i \leq l$ do
        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 $B_i(\boldsymbol{z}_k) \in \mathbb{R}^{D \times c'_i}$.
        If $\langle B_i(\boldsymbol{z}_k), U_i(j) \rangle < \tau$ for some $j$, increase $u_i(j)$ by one and reweight $U_i(j)$ by adding
          $B_i(\boldsymbol{z}_k)$. Otherwise, create a new candidate basis in $U_i$ and a new counter in $u_i$ with
          initial value one.
    end for
end for
for all $1 \leq i \leq l$ do
    Choose the highest vote(s) in $u_i$ with their corresponding basis/bases in $U_i$.
    Assign the samples to their closest subspaces, and remove their votes in other counters
      and bases of higher codimensions.
end for
Segment the remaining samples that are not in the stacks of the highest votes based on
  the estimated bases.
```

Table 3.1 *The percentage of sample points misgrouped 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%

respectively. We then randomly draw a set of samples from them. The samples are corrupted with Gaussian noise. 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.[12] All cases are averaged over 100 trials. The performance of all the algorithms is compared in Table 3.1. The reader can download the MATLAB codes from our website.

As we can see from this table, the voting scheme improves the performance of the algebraic GPCA algorithm. In particular, significant improvements are achieved where the subspaces have different dimensions, e.g., the $(4, 2, 2, 1)$ case. The performance is slightly further improved by combining GPCA-voting with the iterative K-subspace process, which uses the result from GPCA-voting to initialize the iteration.

[12] 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.

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, K-subspaces, and 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 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 following 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 the following:

- Akaike information criterion (AIC) [1] (also known as the C_p statistic [38]) and geometric AIC (G-AIC) [31].
- Bayesian information criterion (BIC) (also known as the Schwartz criterion; see [24] and references therein).
- Minimum description length (MDL) [43] and minimum message length (MML) [62].

Although these criteria were 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 (which 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 our 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 $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N\}$ 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,

$$E[-\log p(\boldsymbol{z}, \hat{\theta}_N)] = \int \big(-\log p(\boldsymbol{z}, \hat{\theta}_N)\big) p(\boldsymbol{z}, \theta_0)\ d\boldsymbol{z} \tag{3.21}$$

corresponds to the expected code length that we use in 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 expected log-likelihood loss above.

AIC relies on an approximation to the expected log-likelihood loss above that holds asymptotically as $N \to \infty$:

$$\text{AIC} \doteq -\frac{2}{N} L(\hat{\theta}_N, \boldsymbol{Z}) + 2\frac{d}{N} \ \approx 2E[-\log p(\boldsymbol{z}, \hat{\theta}_N)], \tag{3.22}$$

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:

$$\text{AIC}(m) = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\|^2 + 2\frac{d(m)}{N}\sigma^2. \tag{3.23}$$

Although motivated by a different reason, 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 all 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 the 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 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 AIC minimizes

$$\text{AIC}(d) \doteq \frac{1}{N}\|\boldsymbol{Z} - \hat{\boldsymbol{Z}}\|^2 + 2\frac{(Dd - d^2)}{N}\sigma^2 \tag{3.24}$$

for our model with parameter space of dimension $Dd - d^2$ and Gaussian noise with variance σ^2. More recently, a geometric version of AIC was proposed by [31], which minimizes

$$\text{GAIC}(d) \doteq \frac{1}{N}\|\boldsymbol{Z} - \hat{\boldsymbol{Z}}\|^2 + 2\frac{(Dd - d^2 + Nd)}{N}\sigma^2, \tag{3.25}$$

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 and the next subsection 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*

$$\mathrm{ED}(\boldsymbol{Z}, \mathcal{A}) \doteq \frac{1}{N}\left(\sum_{j=1}^{n} d_j(D - d_j) + \sum_{j=1}^{n} N_j d_j\right). \tag{3.26}$$

We contend that $\mathrm{ED}(\boldsymbol{Z}, \mathcal{A})$ 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.[13] In the first term of (3.26), $d_j(D - d_j)$ 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 (3.26), $N_j d_j$ 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 is more than one subspace in $\mathcal{A}$, $\mathrm{ED}(\boldsymbol{Z}, \mathcal{A})$ 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 discussed in the beginning of this section. Therefore, we define the *minimum effective dimension* (MED) of a given sample set $\boldsymbol{Z}$ to be the minimum among all possible subspace arrangements that can fit the data set[14]

$$\mathrm{MED}(\boldsymbol{Z}) \doteq \min_{\mathcal{A}:\boldsymbol{Z}\subset\mathcal{A}} \mathrm{ED}(\boldsymbol{Z}, \mathcal{A}). \tag{3.27}$$

[13] We choose here 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.

[14] The space of all subspace arrangements (with a bounded number of subspaces) is topologically compact and closed, hence the MED 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 in 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 $\frac{1}{60}(2\times2\times3-2\times2^2+60\times2) = 2.07$; when we use one plane and two lines, the effective dimension is reduced to $\frac{1}{60}(2\times2\times3-2^2-2\times1+30\times1+30\times2) = 1.6$. In general, if the number of points N is arbitrarily large (say, approaching infinity), depending on the distribution of the points on the lines or the plane, the effective dimension will be between 1 and 2, the true dimensions of the subspaces.

As suggested by the above example, the subspace arrangement model that leads to the MED 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. MED 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 G-AIC criterion (3.25),[15] which leads to the following objective for selecting the optimal subspace arrangement model:

$$\mathcal{A}^* = \arg\min_{\mathcal{A}:\hat{\boldsymbol{Z}}\subset\mathcal{A}} \left\{\frac{1}{N}\|\boldsymbol{Z}-\hat{\boldsymbol{Z}}\|^2 + 2\sigma^2\mathrm{ED}(\hat{\boldsymbol{Z}},\mathcal{A})\right\}, \tag{3.28}$$

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 MED *subject to an error tolerance* τ as

$$\mathrm{MED}(\boldsymbol{Z},\tau) \doteq \min_{\mathcal{A}:\ \|\boldsymbol{Z}-\hat{\boldsymbol{Z}}\|_\infty\le\tau} \mathrm{ED}(\hat{\boldsymbol{Z}},\mathcal{A}), \tag{3.29}$$

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: $\|\boldsymbol{Z}-\hat{\boldsymbol{Z}}\|_\infty = \max_{1\le i\le N}\|\boldsymbol{z}_i-\hat{\boldsymbol{z}}_i\|$. Based on the above definition, the MED 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, each sample point needs to be treated as a one-dimensional subspace in $\mathbb{R}^D$ of its own that brings the MED up close to D.

[15] We here adopt the G-AIC 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.

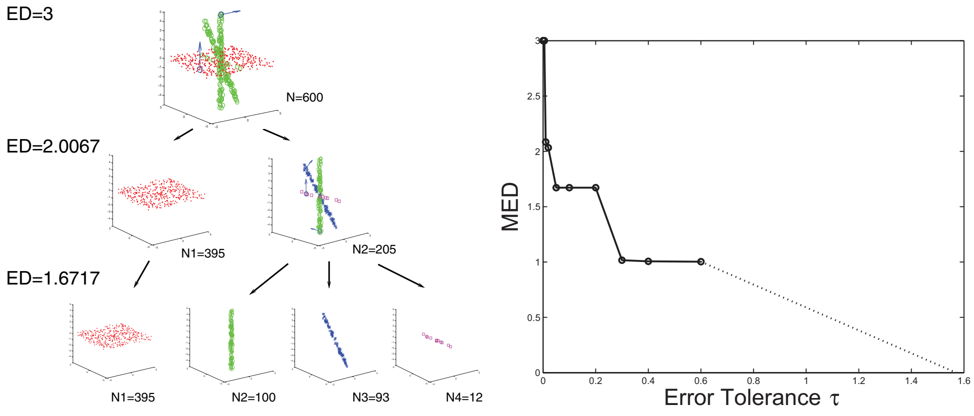


Fig. 3.2 *Left: sample points drawn from two lines and a plane in* $\mathbb{R}^3$ *with* 5% *Gaussian noise are segmented recursively by the GPCA algorithm with an error tolerance* $\tau = 0.05$*. Right: plot of the MED versus the error tolerance.*

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 difference 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, the more compact or compressed is the final representation. In section 5.2, we will see exactly how the MED 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 G-AIC (3.28), the MED objective (3.29) 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 decreases 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 (left side of Figure 3.2). 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.[16] The right side of Figure 3.2 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.

4. Estimation of Subspace Arrangements with Outliers.

In many practical situations the sample points can be contaminated by some atypical samples known

[16] 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 postprocessing.

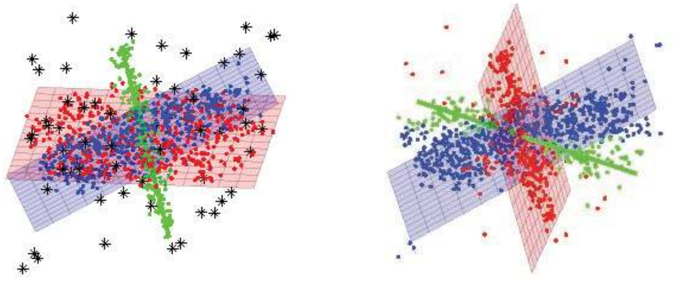


Fig. 4.1 *Results of GPCA (with voting) for samples drawn from two planes and one line in $\mathbb{R}^3$, with 6% Gaussian noise as well as 6% outliers drawn from a uniform distribution (marked as black asterisks "$*$"). Left: the ground truth. Right: estimated subspaces and segmentation result.*

as "outliers" in addition to the noise that we have discussed above. The application of any of the GPCA algorithms to a data set 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."[17] 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 [28, 9]. 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 have relatively *large influence* on the estimated model parameters [8, 11, 23]. A measure of influence is normally the difference between the model estimated with and without the sample in question.
3. Outliers form a set of samples that are *not consistent* with (the model inferred from) the remainder of the data [2, 44, 20]. A measure of inconsistency is normally the error residue of the sample in question with respect to the model.

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 of the polynomials.[18]

In the robust statistics literature, there have been extensive studies about outlier detection and rejection [23, 30, 57, 49]. Most of them are conducted with the assumption that the valid samples points, i.e., the *inliers*, are drawn from a conventional

[17] Earliest documented discussions among astronomers about outliers or "erroneous observations" date back to the mid-18th century. See [3, 28, 4] for a more thorough exposition of the studies of outliers in statistics.

[18] In situations when a data set contains samples drawn from some nonlinear algebraic sets, e.g., a quadratic surface, it is 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 [42] for a more detailed discussion along these lines.

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 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, can both 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 LME [44] and RANSAC [20]. These techniques have been widely used in the areas of computer vision, image processing, and pattern recognition.

4.1. Robust Estimation of Vanishing Polynomials.

4.1.1. Influence Functions. When we try to estimate the parameter θ of the distribution $p(\boldsymbol{z}, \theta)$ from a set of samples $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N\}$, every sample $\boldsymbol{z}_i$ might have an 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, 11, 23].

To measure the influence of a particular sample $\boldsymbol{z}_i$, 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:

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

and the influence of $\boldsymbol{z}_i$ on the estimation of θ can be measured by the difference

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

The function $I(\boldsymbol{z}_i; \theta)$ is also called the *sample influence function* in the literature of robust statistics.

If a set of sample points $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N\}$ 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 $Q(\boldsymbol{X}) = \{q_1(\boldsymbol{X}), q_2(\boldsymbol{X}), \dots, q_m(\boldsymbol{X})\}$ of degree n that vanish on the subspace arrangement. As we discussed in section 3, the coefficients $\boldsymbol{C} = \big(\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_m\big)$ 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 mainly affect the final results by influencing the eigenvectors $\hat{\boldsymbol{C}}$, and subsequently lead to erroneous estimates 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, $\hat{S} = \text{span}(\hat{\boldsymbol{C}})$. Therefore, the influence of the sample $\boldsymbol{z}_i$ 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 time that the sample influence function was proposed [8, 11], when computational resources were scarcer than they are today. In robust statistics, formulas 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 1. 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 sample influence above,

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

As derived in [11], 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

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

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 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 [30].

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 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 [19] 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* [28] and *multivariate trimming* (MVT) [22]. 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* [28, 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 of 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 [22]). 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 set of indexes of the 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.

4.2. Estimating the Outlier Percentage. All 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 introduce a solution to estimate the outlier percentage. The percentage will be determined so 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.

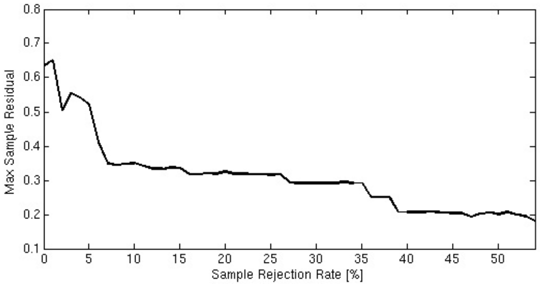


Fig. 4.2 *Maximal sample residuals with various rejection rates on subspace arrangement models estimated using the MVT covariance estimator. The data set is contaminated by* 16% *uniformly distributed outliers.*

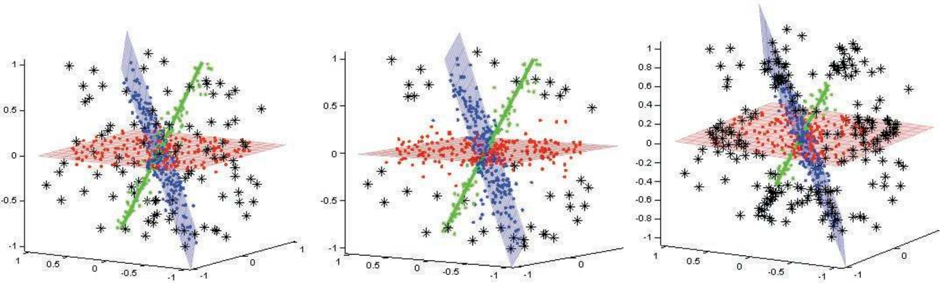


Fig. 4.3 *Subspace segmentation results. Left: a priori data (outliers are marked as black asterisks "∗"). Middle: the estimation result with* 7% *samples rejected. Right: the estimation result with* 38% *samples rejected.*

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 as an example 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 4.2 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 4.3 shows the segmentation results at rejection rates 7% and 38%, respectively.

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 an even smaller maximal residual as shown in Figure 4.2.

Therefore, one does not have to reject the a priori outlier percentage in order to obtain a good estimate of the arrangement model. 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 prespecified 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 4.2, 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.

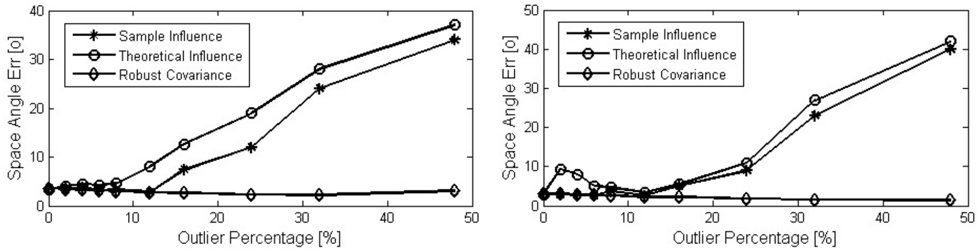
In this work, we propose to determine the outlier percentage as the smallest percentage such that the maximal sample residual is smaller than a given residual threshold for several consecutive rejection rates (i.e., the residual "stabilizes"). The residual threshold can be seen as the variance of the noise of the valid data. It plays a similar role as the error tolerance when we determine the MED 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 [20], later in this section. Algorithm 3 gives an outline of the resulting algorithm, which we refer to as Robust GPCA (GRGPCA).

ALGORITHM 3. *Robust GPCA.*

Given a set of samples $\boldsymbol{X} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_N\}$ in $\mathbb{R}^D$, a threshold τ for the subspace angle, and a residual threshold σ, fit n linear subspaces of codimensions $c_1, c_2, \dots, c_n$:

```
Set a maximal possible outlier percentage $M\%$.
Normalize the data such that the max vector magnitude is 1.
for all rejection rate $0 \le r \le M$ do
    $\boldsymbol{X}' \leftarrow$ removing $r\%$ samples from $\boldsymbol{X}$ using MVT or influence function.
    Estimate the subspace bases $\{\hat{B}_1, \hat{B}_2, \dots, \hat{B}_n\}$ by applying GPCA to $\boldsymbol{X}'$ with parameters $\tau$ and $c_1, c_2, \dots, c_n$.
    Maximal residual $\sigma_{\max} \leftarrow \max_{\boldsymbol{z} \in \boldsymbol{X}'} \min_k \|\boldsymbol{z} - \hat{B}_k \hat{B}_k^T \boldsymbol{z}\|$.
    if $\sigma_{\max}$ is consistently smaller than $\sigma$ then
        $B_k \leftarrow \hat{B}_k$ for $k = 1, 2, \dots, n$. Break.
    end if
end for
if $\sigma_{\max} > \sigma$ then
    ERROR: the given $\sigma$ is too small.
else
    Label $\boldsymbol{z} \in \boldsymbol{X}$ as an inlier if $\min_k \|\boldsymbol{z} - B_k B_k^T \boldsymbol{z}\| < \sigma$.
    Segment the inlying samples to their respective subspaces.
end if
```

To demonstrate the performance of the algorithm, we conduct two 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 $(400, 200, 200, 100)$. The maximal data magnitude is 1, and it is corrupted with 6% Gaussian noise and uniformly distributed outliers of a series of percentages from 0%



(a) $(2, 2, 1)$ in $\mathbb{R}^3$

(b) $(4, 2, 2, 1)$ in $\mathbb{R}^5$

Fig. 4.4 *Average space angle errors (in degree) of arrangements estimated by RGPCA.*

Table 4.1 *Average time for estimating the three subspace arrangements with* 24% *uniform distributed outliers via RGPCA.*

Arrangement	$(2, 2, 1)$ in $\mathbb{R}^3$	$(4, 2, 2, 1)$ in $\mathbb{R}^5$
Sample influence	3m	58m
Theoretical influence	1.4m	40m
MVT	46s	23m

to 48%. The experiment is repeated 100 times at each percentage. For RGPCA using either MVT or the influence functions, the residual threshold σ is fixed at 0.04 and the angle threshold τ is fixed at 0.3 rad. Figure 4.4 shows the results of the average angle error in the unit of degree. Table 4.1 shows the average time of the algorithm on a dual 2.7 GHz Macintosh workstation.

We compare the performance of the three different RGPCA algorithms. In terms of accuracy, MVT gives the best overall estimation on the two synthesized data sets. The subspace angle errors for the two cases are both within two degrees with up to 50% outliers. The two influence function approaches also give comparable results for the data sets that have outliers less than 30%. In terms of speed, MVT is also the fastest among the three algorithms, and the sample influence approach naturally falls to the slowest.

Finally, we need to caution the reader that the excellent performance given by MVT here is partially due to the fact that the outliers in the simulations are generated from an idealistic uniform distribution. However, in real applications where the outliers may come from any arbitrary distribution, the influence function approach may outperform the MVT approach. For more detailed discussion and comparisons in applications in computer vision, the reader is referred to [64, 63].

4.3. Other Robust Statistical Techniques. The above robust techniques have one thing in common: To begin with, they all rely on an estimate of the model from all the samples. This to some extent puts a limit on the number 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 [45, 49]. There is yet another category of robust statistical techniques that are based on *random sampling*, including but not limited to LME [44], RANSAC [20], 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 over 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 when 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 that the dimension of the vector $\nu_n(\boldsymbol{z})$ is 70 (which corresponds to the case of 4 subspaces in $\mathbb{R}^5$). To estimate a hyperplane of dimension 69 in $\mathbb{R}^{70}$ with 20% outliers, in order to have at least one subset of 69 inliers with probability 0.95, one needs to subsample 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 may consider applying the random sampling techniques to find one subspace at a time. The number of sampling subsets 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 subsample about 1,900 subsets. However, when an arrangement contains subspaces of different dimensions, samples from one subspace can also result in high consensus on other subspace models of higher or lower dimensions. Therefore, an algorithm has to be able to detect degenerate models in the estimation to achieve good performance.
3. *Estimating Mixture Models.* For a mixture model such as a subspace arrangement, another natural sampling scheme is to subsample 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 can subsample 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 sampling subsets 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 subsample over 1.2 billion subsets to have one subset of 15 inliers with probability of 95%.

In the computer vision literature, RANSAC has shown good performance for some special subspace arrangements. For instance, if all subspaces are of the same dimension, the second approach has been successfully used to iteratively recover one subspace model at a time [55]. If the subspaces have different dimensions, a *Monte Carlo* scheme can be applied to speed up the third approach for the estimation of a hybrid model [56, 48]. The reader can find more comparisons between RGPCA and RANSAC in [64].

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 subspace 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. In such cases, we apply GPCA to find a hybrid linear model that can *approximate* the data up to a desired degree of accuracy. The resulting model provides a compact (lossy) representation of the data as well as a partition of the data into approximately 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 scene observed in 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 feature points that belong to different moving objects. More precisely, denote by $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_N\} \subset \mathbb{R}^3$ 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 three-dimensional scene.

Of course, the problem depends on how the three-dimensional points $\boldsymbol{X}_1, \boldsymbol{X}_2, \dots \boldsymbol{X}_N$ 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 three- or two-dimensional 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 [59]. 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[19] from $\mathbb{R}^3$ to $\mathbb{R}^2$:

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

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

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

[19] 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.

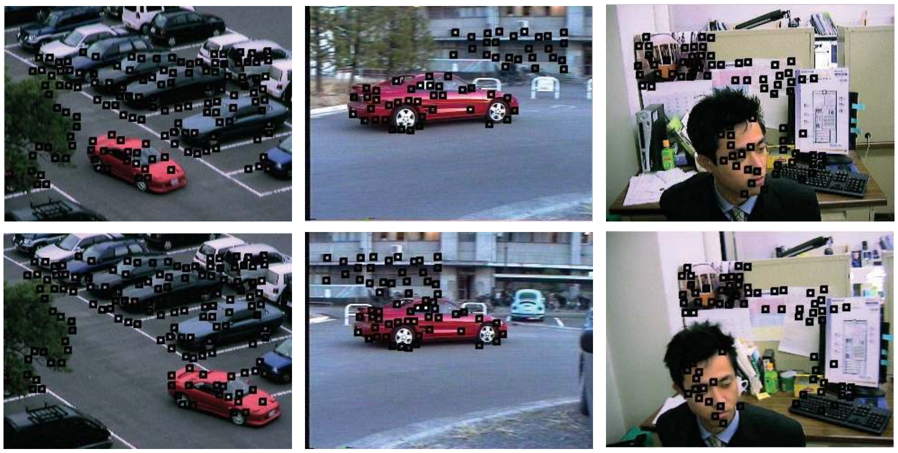


Fig. 5.1 *The first and last frames of sequences A (left), B (middle), and C (right) with point correspondences superimposed.*

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 two-dimensonal 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 into their respective subspaces; see [10, 32, 50] and the 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 [51], 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,

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

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 [51] from other existing *multiframe* algorithms for motion segmentation.

Next, we demonstrate the performance of RGPCA (Algorithm 3) with MVT on sequences A and C with original tracking outliers added in, as shown in Figure 5.2. The data are borrowed from [50] and are also available at the website http://www.suri.it.okayama-u.ac.jp/e-program-separate.html. The reported outlier percentages in [50]

Table 5.1 *Classification rates given by various subspace segmentation algorithms for sequences A, B, and 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: multistage optimization	100.0%	100.0%	100.0%
GPCA-voting	100.0%	100.0%	100.0%

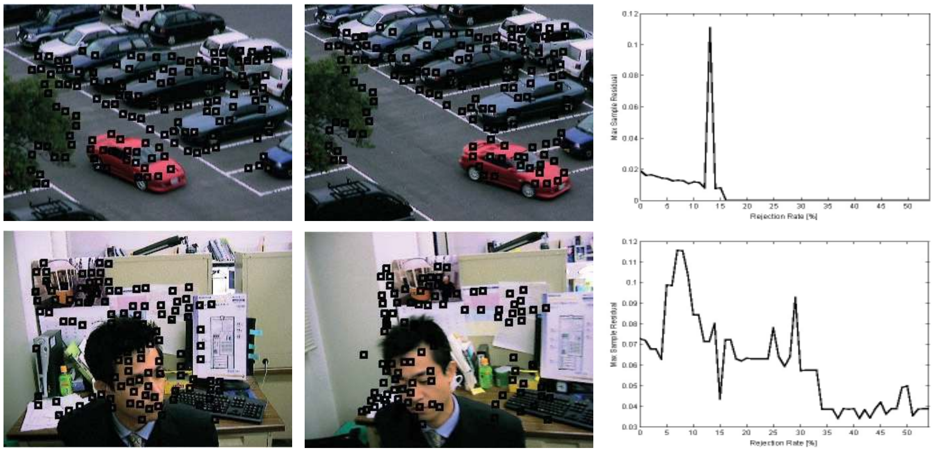
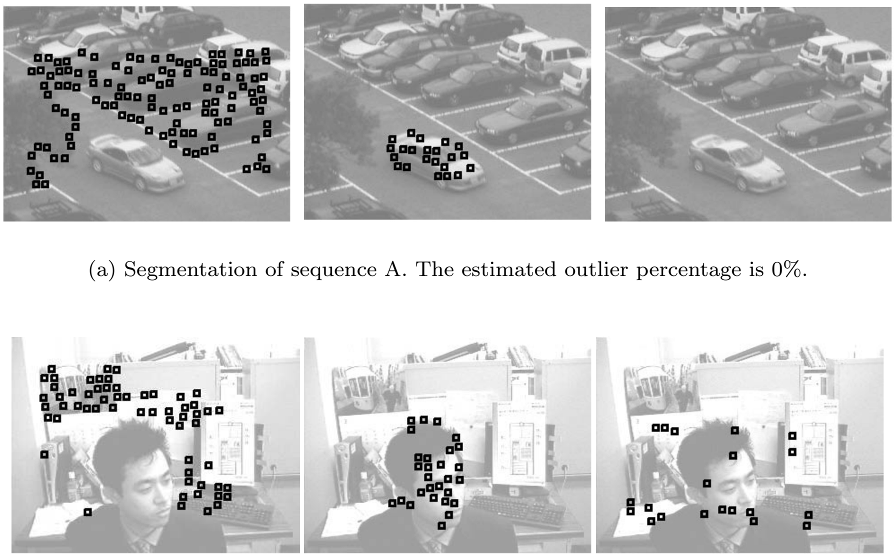


Fig. 5.2 *The first (left) and last (middle) frames of sequences A and C with the original tracking outliers. The right column shows the maximal residual values of the two sequences with various rejection rates using the MVT algorithm. Sequence A contains* 140 *feature points, and sequence B contains* 107 *feature points.*

were 1.4% and 30%, respectively. We use Algorithm 3 with MVT 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 of 0% and 18% for sequences A and C, respectively, which outperforms the results reported in [50].

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 the actual camera projection well. Furthermore, the motion of the man's upper body is nonrigid, 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 indicates that a good segmentation can be achieved at the 30% rejection rate, which conforms to the percentage given in [50]. To make the comparison complete, we show



(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.*

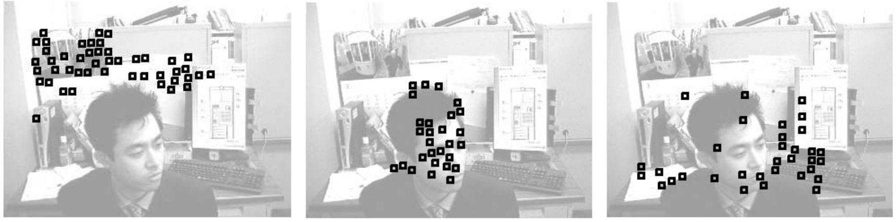


Fig. 5.4 *Segmentation results of C with* 30% *rejection rate. Left: group* 1. *Middle: group* 2. *Right: outliers.*

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, 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.

A linear transformation can be either *prefixed* for all images (such as the discrete cosine transform (DCT) used for the JPEG standard and the wavelet transform for the JPEG2000 standard) or *adaptive* for each image (such as the Karhunen–Loève

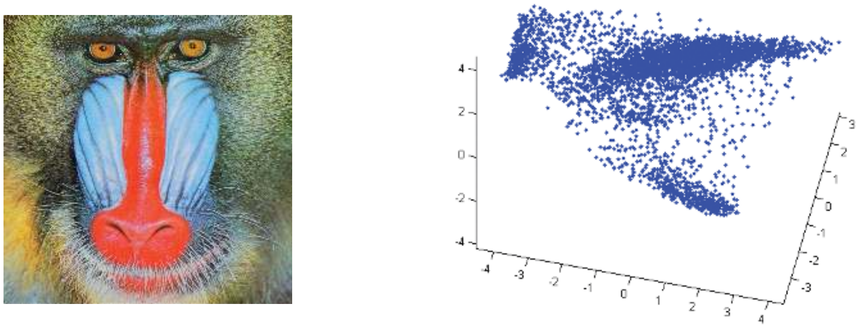


Fig. 5.5 *Left: the baboon image. Right: the coordinates of each dot are the first three principal components of a 2×2 color block of the image (stacked into a vector in $\mathbb{R}^{12}$). There is a visible multimodal structure in the data.*

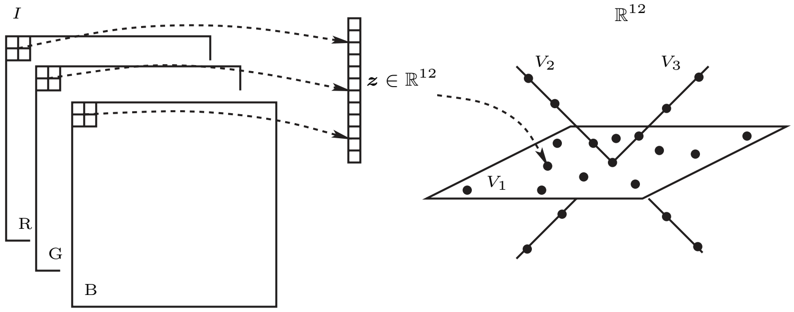


Fig. 5.6 *In a hybrid linear model, vectors $\{\boldsymbol{z}_i\}$ (obtained by stacking the image blocks) are assumed to reside in multiple subspaces which may have different dimensions.*

transform (KLT) 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 homogeneous subset. Figure 5.6 illustrates the basic idea.

Obviously, the same assumptions can be made for any transformed image, say, a subsampled 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) subsampled version and its residuals at two higher levels. Figure 5.7 (right) shows the segmentation of the subsampled 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 [26].

Typically, such a multiscale scheme can achieve a more compact representation because it extracts low-frequency parts of the image first.[20] Figure 5.8 gives a com-

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

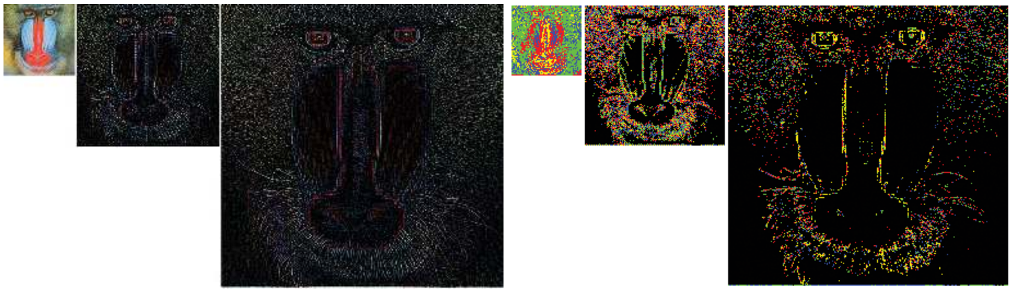


Fig. 5.7 *A multiscale representation of the baboon image. Left: twice subsampled image and its residuals at two higher levels. Right: the segmentation of vectors (2×2 blocks) by a hybrid linear model at each level—different subspaces are denoted by different colors. The black regions correspond to data vectors whose energy is below a given error threshold.*

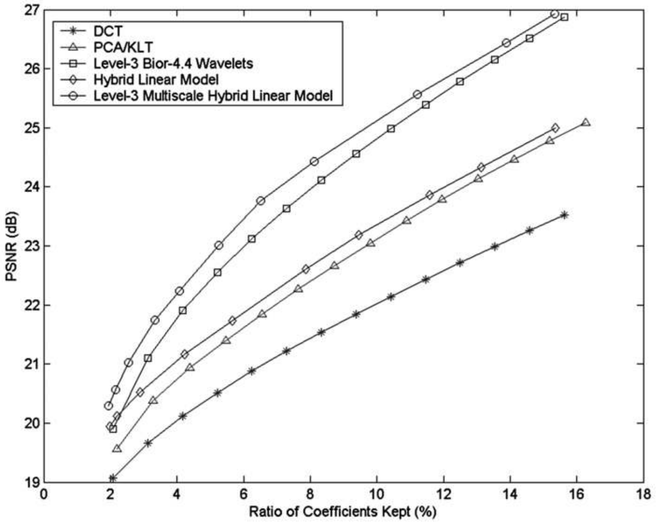


Fig. 5.8 *Comparison of several lossy image representation schemes for the baboon image. Vertical axis: here the signal is the original image and the noise is the difference between the original image and its approximation given by different representation schemes. Horizontal axis: percentage of the ratio of coefficients kept.*

parison of the efficiency of different lossy image representation schemes for the baboon image: the DCT, the KLT, the hybrid linear model (without subsampling), the level-3 biorthogonal 4.4 wavelets (used in JPEG2000), and the level-3 multiscale hybrid linear model.[21]

Potentially, there might be many other ways of applying the (multiscale) 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 multiscale hybrid linear model in the wavelet domain (see [26]).

[21] The experimental results given here are attributed to Dr. Wei Hong.

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:

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 on 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 linear systems such as the hybrid autoregressive exogeneous (ARX) model [36] and the hybrid autoregressive moving average (ARMA) model [27].
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 [25]. Thus, the problem of clustering face images that belong to different people can also be characterized as a problem of subspace arrangement estimation. The GPCA algorithm has been quite successful in solving this problem [61]. 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, in both the spatial and temporal domains [27, 61].

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 has 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 EM and K-means. As a consequence, the resulting algorithms, GPCA and its variations, require no initialization and can be used in combination with EM and K-means.

These new algorithms have been shown to be particularly effective in the modeling and segmenting of imagery data, including but not limited to conventional images, videos, and biological images, as well as hyperspectral images. The initial success of these tools in the identification of hybrid systems [36, 27] 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 [53, 46]. GPCA provides yet another class of tools that allow us to obtain a piecewise linear approximation of the manifold (subject to an error threshold). Important geometric or topological properties, e.g., dimension(s) or components, of the manifold can be extracted from such an approximation. Our recent work has also shown that it is possible to extend the ideas of GPCA to other polynomial rings such as the ideals of quadratic algebraic surfaces [42]. This suggests that there is much more to come for modeling and segmenting mixed data with other classes of hybrid algebraic manifolds.

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 inspiration for many of the new algorithms. In our most recent work, parallel to the algebraic framework that we have covered in this paper, we have revealed a somewhat unusual connection between subspace arrangements and information theory: data from a subspace arrangement can be very effectively segmented via lossy data compression [35]. Other ongoing research also suggests that tools from sparse representation and ℓ^1-minimization [16, 15] may also lead to effective algorithms for segmenting data from arrangements of low-dimensional subspaces. Given this intensifying confluence, we would not be surprised if even more powerful algorithms are found in the near future, capable of handling massive, multivariate, and mixed data.

Acknowledgments. The authors thank Professor René Vidal of the Johns Hopkins University, Professor Kun Huang of the Ohio State University, Dr. Wei Hong of Texas Instruments, and Shankar Rao, Andrew Wagner, and John Wright of the University of Illinois at Urbana-Champaign. Their research has contributed to part of the material presented in this paper. The authors also thank the editor and the referees whose comments and suggestions have greatly enhanced the quality of the paper.

REFERENCES

[1] H. Akaike, *A new look at the statistical model selection*, IEEE Trans. Automat. 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*, 2nd ed., John Wiley & Sons, New York, 1983.

[4] P. 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*, Math. Comp., 27 (1973), pp. 579–594.

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

[7] A. Björner, I. Peeva, and J. Sidman, *Subspace arrangements defined by products of linear forms*, J. London Math. Soc. (2), 71 (2005), pp. 273–288.

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

[9] N. Campbell, *Robust procedures in multivariate analysis* I: *Robust covariance estimation*, Appl. Statist., 29 (1980), pp. 231–237.

[10] J. Costeira and T. Kanade, *A multibody factorization method of independently moving objects*, Internat. J. Comput. Vision, 29 (1998), pp. 159–179.

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

[12] A. Dempster, N. Laird, and D. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Statist. Soc. Ser. B, 39 (1977), pp. 1–38.

[13] H. Derksen, *Hilbert Series of Subspace Arrangements*, preprint, arXiv.org, 2005; available online from http://arxiv.org/abs/math/0510584.

[14] D. Donoho, *Sparse component analysis and optimal atomic decomposition*, Constr. Approx., 17 (1998), pp. 353–382.

[15] D. Donoho, *For Most Large Underdetermined Systems of Linear Equations, the Minimal l^1-norm Near-Solution Approximates the Sparsest Near-Solution*, Technical report, Stanford University, Stanford, CA, 2004.

[16] D. Donoho, *For Most Large Underdetermined Systems of Linear Equations the Minimal l^1-norm Solution Is Also the Sparsest Solution*, Technical report, Stanford University, Stanford, CA, 2004.

[17] D. Donoho, *Neighborly Polytopes and Sparsest Solution of Underdetermined Linear Equations*, Technical report, Stanford University, Stanford, CA, 2004.

[18] D. Eisenbud, *Commutative Algebra with a View Toward Algebraic Geometry*, Springer-Verlag, New York, 1995.

[19] T. Ferguson, *On the rejection of outliers*, in Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability I, University of California Press, 1961, pp. 253–287.

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

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

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

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

[24] M. Hansen and B. Yu, *Model selection and the principle of minimum description length*, J. Amer. Statist. Assoc., 96 (2001), pp. 746–774.

[25] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman, *Clustering appearances of objects under varying illumination conditions*, in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition I, 2003, pp. 11–18.

[26] W. Hong, J. Wright, K. Huang, and Y. Ma, *A multi-scale hybrid linear model for lossy image representation*, in Proceedings of the IEEE International Conference on Computer Vision I, 2005, pp. 764–771.

[27] K. Huang, A. Wagner, and Y. Ma, *Identification of hybrid linear time-invariant systems via subspace embedding and segmentation*, in Proceedings of the IEEE Conference on Decision and Control 3, 2004, pp. 3227–3234.

[28] P. Huber, *Robust Statistics*, John Wiley & Sons, New York, 1981.

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

[30] I. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer-Verlag, New York, 2002.

[31] K. Kanatani, *Model selection for geometric inference*, in Proceedings of Asian Conference on Computer Vision, 2002, pp. xxi–xxxii.

[32] K. Kanatani, *Motion segmentation by subspace separation: Model selection and reliability evaluation*, Internat. J. Image Graphics, 2 (2002), pp. 179–197.

[33] S. Lang, *Algebra*, Springer-Verlag, New York, 2002.

[34] S. Lloyd, *Least squares quantization in PCM*, IEEE Trans. Inform. Theory, 28 (1982), pp. 129–137.

[35] Y. Ma, H. Derksen, W. Hong, and J. Wright, *Segmentation of multivariate mixed data via lossy coding and compression*, IEEE Trans. Pattern Anal. Machine Intelligence, 29 (2007), pp. 1546–1562.

[36] Y. Ma and R. Vidal, *A closed form solution to the identification of hybrid ARX models via the identification of algebraic varieties*, in Proceedings of the International Conference on Hybrid Systems Computation and Control, 2005, pp. 449–465.

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

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

[39] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, John Wiley & Sons, New York, 1997.

[40] B. Olshausen and D. Field, *Emergence of simple-cell receptive field properties by learning a sparse code for natural images*, Nature, 381 (1996), pp. 607–609.

[41] P. Orlik, *Introduction to Arrangements*, CBMS Regional Conf. Ser. Math. 72, AMS, Providence, RI, 1989.

[42] S. Rao, A. 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, pp. 2–9.

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

[44] P. Rousseeuw, *Least median of squares regression*, J. Amer. Statist. Assoc., 79 (1984), pp. 871–880.

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

[46] S. Roweis and L. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science, 290 (2000), pp. 2323–2326.

[47] P. 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.

[48] K. Schindler and D. Suter, *Two-view multibody structure-and-motion with outliers*, in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2005, pp. 643–648.

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

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

[51] Y. Sugaya and K. Kanatani, *Multi-stage optimization for multi-body motion segmentation*, IEICE Trans. Inform. Systems, E87-D (2004), pp. 1935–1942.

[52] G. Taubin, *Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation*, IEEE Trans. Pattern Anal. Machine Intelligence, 13 (1991), pp. 1115–1138.

[53] J. Tenenbaum, V. de Silva, and J. Langford, *A global geometric framework for nonlinear dimensionality reduction*, Science, 290 (2000), pp. 2319–2323.

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

[55] P. Torr, *Geometric motion segmentation and model selection*, R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci., 356 (1998), pp. 1321–1340.

[56] P. Torr and C. Davidson, *IMPSAC: Synthesis of importance sampling and random sample consensus*, IEEE Trans. Pattern Anal. Machine Intelligence, 25 (2003), pp. 354–364.

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

[58] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[59] 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, pp. 1–15.

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

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

[62] C. Wallace and D. Boulton, *An information measure for classification*, Computer J., 11 (1968), pp. 185–194.

[63] A. Yang, *Estimation of Subspace Arrangements: Its Algebra and Statistics*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 2006.

[64] A. Yang, S. Rao, and Y. Ma, *Robust statistical estimation and segmentation of multiple subspaces*, in Workshop on 25 years of RANSAC, IEEE International Conference on Computer Vision and Pattern Recognition, 2006, p. 99.