The algebra and statistics of generalized principal component analysis

Shankar Rao, Harm Derksen, Robert Fossum, Yi Ma, Andrew Wagner, et al.

Shankar Rao, Harm Derksen, Robert Fossum, Yi Ma, Andrew Wagner, Allen Yang, "The algebra and statistics of generalized principal component analysis," Proc. SPIE 6508, Visual Communications and Image Processing 2007, 65080G (29 January 2007); doi: 10.1117/12.707527

SPIE.

Event: Electronic Imaging 2007, 2007, San Jose, CA, United States

The Algebra and Statistics of Generalized Principal Component Analysis

Shankar Rao[a], Harm Derksen[b], Robert Fossum[c], Yi Ma[a], Andrew Wagner[a], and Allen Yang[d]

[a]Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA;
[b]Dept. of Mathematics, University of Michigan, USA;
[c]Dept. of Mathematics, University of Illinois at Urbana-Champaign, USA;
[d]Dept. of Electrical Engineering and Computer Science, University of California at Berkeley, USA

ABSTRACT

We consider the problem of simultaneously segmenting data samples drawn from multiple linear subspaces and estimating model parameters for those subspaces. This "subspace segmentation" problem naturally arises in many computer vision applications such as motion and video segmentation, and in the recognition of human faces, textures, and range data. Generalized Principal Component Analysis (GPCA) has provided an effective way to resolve the strong coupling between data segmentation and model estimation inherent in subspace segmentation. Essentially, GPCA works by first finding a global algebraic representation of the unsegmented data set, and then decomposing the model into irreducible components, each corresponding to exactly one subspace. We provide a summary of important algebraic properties and statistical facts that are crucial for making GPCA both efficient and robust, even when the given data are corrupted with noise or contaminated by outliers. We demonstrate the effectiveness of GPCA using a large testbed of synthetic and real experiments.

1. INTRODUCTION

Linear subspace models are popular choices for parametric modeling of data sets, primarily because they are simple to understand and easy to represent and compute. However data sets are often inhomogeneous and thus cannot be well described well by single linear models. This is especially the case of imagery data. For instance, a natural image typically contains multiple regions that 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. Therefore, it is reasonable to use multiple models to describe different regions of the image.

The above example about images reveals a challenging problem that permeates many research areas such as image processing, computer vision, pattern recognition, and system identification: *How to segment a given set of data into multiple subsets and find the best model for each subset?* In different contexts, such a data set, as well as the associated model, has been called "mixed," or "multi-modal," or "multi-model," or "piecewise," or "heterogeneous," or "hybrid." For simplicity, in this paper, we refer to the data as "mixed" and the model as "hybrid." We are particularly interested in the *hybrid linear model*: one linear model for every homogeneous subset of the data. The importance of hybrid linear models is multi-fold: 1. They are the natural generalizations to single linear models; 2. They are sufficiently expressive for representing or approximating arbitrary complex data structures; 3. The understanding of hybrid linear models has been significantly advanced in recent years and many efficient solutions have been developed.

A fundamental challenge in estimating such a hybrid model for mixed data is the "chicken-and-egg" problem. If the data were already segmented properly into homogeneous subsets, estimating a model for each subset would be easy. Or, if the hybrid model were already known, segmenting the data into multiple subsets would be straightforward. 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. Typically, this question is answered via iterative techniques that, given an initial estimate of the model, will cycle between segmentation and model estimation until a stationary point is found.

Further author information: (Send correspondence to S.R.R)
S.R.R.: E-mail: srrao@uiuc.edu, Telephone: 1 217 244 6626
H.D: E-mail: hderksen@umich.edu, Telephone: 1 734 763 2309
R.M.F.: E-mail: rmfossum@uiuc.edu, Telephone: 1 217 244 3572
Y.M: E-mail: yima@uiuc.edu, Telephone: 1 217 244 0871
A.W.W E-mail: awagner@uiuc.edu, Telephone: 1 217 244 6626
A.Y.Y E-mail:yang@eecs.berkeley.edu, Telephone: 1 510 643 5798

Visual Communications and Image Processing 2007, edited by Chang Wen Chen, Dan Schonfeld, Jiebo Luo,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6508, 65080G, © 2007 SPIE-IS&T · 0277-786X/07/$18

Iterative statistical methods have several drawbacks that limit their applicability in estimating hybrid models. First, if the algorithm is not properly initialized, the iterative process may converge to a local extremum that gives an invalid estimate of the model. In practice, to increase the chance of finding the global extremum, one often needs to run the algorithm multiple times with random initialization, which obviously reduces the efficiency of the algorithm. Second, the statistical formulation typically relies on explicit assumptions about the mixture distribution: the number of component distributions, the parametric models of the distributions, and the dimension or complexity of each model, etc. However, in many practical applications, such information is not readily available and needs to be inferred from the given data. Finally, statistical methods such as maximum likelihood are known to be less effective when dealing with situations in which the distributions are degenerate.[1] These situations arise very often for mixed data with a hybrid model.

In recent years, there has been increasing interest in an algebro-geometric approach to the subspace segmentation problem know as "Generalized Principal Component Analysis". The GPCA algorithm,[2] provides an effective way to resolve the coupling between data segmentation and model estimation. Basically, it seeks a global algebraic representation of the unsegmented data set. The segmentation and estimation of individual subspaces can then be obtained by decomposing the global algebraic model into irreducible components, each component corresponding to exactly one subspace.

1.1. Problem Statement

In this paper we address the following "subspace segmentation" 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 $\mathbb{R}^D$, estimate a basis for each subspace, and segment all sample points into their respective subspaces.*

Although this algebraic GPCA provides a theoretical solution to subspace segmentation, it does not answer some important questions regarding subspace segmentation in the presence of *noisy* data and statistical outliers. We show how GPCA can be modified to be both stable in the presence of noisy data, and robust in the presence of statistical outliers.

1.2. Organization of This Paper

In this paper we review the solutions to Problem 1.1, both in the ideal case, and with noise (and/or) outliers. If the sample points are noiseless, the problem is mostly an algebraic problem. Section 2 reviews the basic algebraic properties of subspace arrangements. As an algebraic set, the set of polynomials that vanish on a subspace arrangement form an ideal and the subspace arrangement is uniquely determined by this ideal. We give a complete characterization of the dimension of each graded component of the ideal, also known as the Hilbert function. We show how the dimensions of the subspaces can be uniquely determined from the values of the Hilbert function. We further show how the vanishing ideal can be determined from a sufficiently dense (nevertheless finite) set of sample points on the arrangement; and how the subspaces can be subsequently deduced from the vanishing polynomials. These concepts will naturally to the algorithm we call *Generalized Principal Component Analysis* (GPCA).

In Section 3.1, we review some statistical techniques that allow us to estimate the individual subspaces from sample points that are corrupted by noise. In Section 3.2, we study the problem under the assumption that the given sample points are contaminated with outliers. We introduce certain robust statistical techniques that can detect or diminish the effect of outliers, especially for subspace arrangements. Finally, we demonstrate how these methods can be applied to several real-world applications in Section 5.*

2. INFERENCE OF SUBSPACE ARRANGEMENTS VIA ALGEBRAIC TECHNIQUES

In this section we review key algebraic facts that serve as the theoretical foundation for generalized principal component analysis. We show that under very general conditions, a hybrid linear model with subspaces of known dimensions can be inferred from data samples via the "Hilbert functions".

*The source code for all the algorithms, as well as more applications, are available on-line at `http://perception.csl.uiuc.edu/gpca/`.

2.1. Basic Notation and Definitions

In what follows, if V is a d-dimensional subspace, then its **codimension** is $c \doteq D - d$.

DEFINITION 2.1 (SUBSPACE ARRANGEMENT). *A **subspace arrangement** in $\mathbb{R}^D$ is a union*

$$\mathcal{A} \doteq V_1 \cup V_2 \cup \cdots \cup V_n \tag{1}$$

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

For a nonempty subset S of the index set $\{1, 2, \ldots, n\}$, we define the intersection $V_S \doteq \cap_{s \in S} V_s$ with dimension $d_S \doteq \dim V_S$ and codimension $c_S \doteq D - d_S$.

DEFINITION 2.2 (TRANSVERSAL SUBSPACE ARRANGEMENT). *A subspace arrangement $\mathcal{A} = V_1 \cup V_2 \cup \ldots \cup V_n$ is called **transversal** if*

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

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

The ring of polynomial functions on our ambient space $\mathbb{R}^D$ is denoted by $\mathbb{R}^{[D]} \doteq \mathbb{R}[X_1, X_2, \ldots, X_D]$. It is the ring of polynomials in the functions $\{X_1, X_2, \ldots, X_D\}$, where X_j is the function that assigns the jth coordinate to a point in $\mathbb{R}^D$. Any polynomial $f \in \mathbb{R}^{[D]}$ 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 $\mathbb{R}_h^{[D]}$ denote the vector space of all polynomials homogeneous of degree h. Then there is a decomposition

$$\mathbb{R}^{[D]} = \mathbb{R} \oplus \mathbb{R}_1^{[D]} \oplus \mathbb{R}_2^{[D]} \oplus \cdots \tag{2}$$

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

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

$$M_h^{[D]} \doteq \binom{h + D - 1}{D - 1}. \tag{3}$$

One can verify this by observing that the monomials $\{X_1^h, X_1^{h-1} X_2, X_1^{h-1} X_3, \ldots, X_D^h\}$ form a basis of $\mathbb{R}_h^{[D]}$.

DEFINITION 2.3 (VERONESE MAP). *The **Veronese Map** of order h is the map $\nu_h : \mathbb{R}^D \to \mathbb{R}^{M_h^{[D]}}$ given by*

$$\nu_h([x_1, \boldsymbol{z}_2, \ldots, x_D]^T) = [x_1^h, x_1^{h-1} x_2, \ldots, x_D^h]^T.$$

2.2. Vanishing Ideals and Hilbert Functions of Subspace Arrangements

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

DEFINITION 2.4 (VANISHING IDEAL). *The **vanishing ideal** $I(W)$ of a subset $W \subseteq \mathbb{R}^D$ is defined by $I(W) \doteq \{f \in \mathbb{R}^{[D]} : f(\boldsymbol{z}) = 0, \forall \boldsymbol{z} \in W\}$.*

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

An ideal is homogeneous if and only if it is generated by homogeneous elements. The vanishing ideal $I(V)$ of a subspace $V \subseteq \mathbb{R}^D$ is generated by the linear functions in $V^\perp$, 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:

$$I(\mathcal{A}) = I(V_1 \cup V_2 \cup \cdots \cup V_n) = I(V_1) \cap I(V_2) \cap \cdots \cap I(V_n). \tag{4}$$

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 $I_h = I(\mathcal{A}) \cap \mathbb{R}_h^{[D]}$ 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

$$I(\mathcal{A}) = I_m \oplus I_{m+1} \oplus \cdots \oplus I_n \oplus I_{n+1} \oplus \cdots . \tag{5}$$

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

$$J(\mathcal{A}) = J_n \oplus J_{n+1} \oplus \cdots . \tag{6}$$

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

$$J_n = V_1^{\perp} V_2^{\perp} \cdots V_n^{\perp} = I_1(V_1) I_1(V_2) \cdots I_1(V_n). \tag{7}$$

DEFINITION 2.6 (ZERO SET). *Given a set of polynomials $I \subseteq \mathbb{R}^{[D]}$, the **zero set** of I is defined to be $Z(I) \doteq \{\boldsymbol{z} \in \mathbb{R}^D : g(\boldsymbol{z}) = 0$ for all $g \in I\} \subseteq \mathbb{R}^D$.*

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

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

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

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

$$h_K(j) \doteq \dim(K_j), \tag{8}$$

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

The remainder of this section is devoted to providing a closed-form formula for the Hilbert function $h_I(i)$ of subspace arrangements that is valid for $i \geq n$. A more complete development is given in.[3]

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

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

$$\mathcal{H}(K, t) \doteq \sum_{i \in \mathbb{N}} h_K(i) t^i. \tag{9}$$

A recursive formula for the Hilbert series of $J(\mathcal{A})$ was given in.[3] Surprisingly, this formula depends only on the codimensions of the intersections (c_S, $S \subseteq \{1, 2, \ldots, n\}$) and D, the dimension of the ambient vector space. This means that the Hilbert series $\mathcal{H}(J(\mathcal{A}), t)$ is a *combinatorial invariant* of the arrangement $\mathcal{A}$. Combinatorial invariants play an important role in the study of subspace arrangements and hyperplane arrangements. In general, the Hilbert series of $I(\mathcal{A})$ is *not* a combinatorial invariant. This means the series $\mathcal{H}(I(\mathcal{A}), t)$ depends more delicately on the geometry of the arrangement. 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

$$F_{\mathcal{A}}(t) \doteq \frac{\prod_{i=1}^{n} (1 - (1-t)^{c_i})}{(1-t)^D}. \tag{10}$$

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

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 2.10 ([3]). *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 (10) 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 2.11. *If $\mathcal{A} = V_1 \cup V_2 \cup \cdots \cup V_n$ is transversal, then for all $i \geq n$,*

$$h_I(i) = h_J(i) = g(i) = \sum_S (-1)^{|S|} \binom{i + D - 1 - c_S}{D - 1 - c_S}, \tag{11}$$

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

The formula in Corollary 2.11 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 $g(i)$, $i = n, n+1, n+2, \ldots$.

COROLLARY 2.12. *Consider a transversal arrangement of n subspaces. The codimensions $c_1, \ldots, c_n$ (and hence the dimensions) of the subspaces are uniquely determined by the values of the Hilbert function $h_I(i)$ for $i = n, n+1, \ldots, n+D-1$.*

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

1. The equality $h_I(i) = h_J(i)$ for $i \geq n$ implies that $I_i = J_i$ for $i \geq n$ and in particular $I_n = J_n$. That is, the homogeneous component I_n of the vanishing ideal of a transversal subspace arrangement is always generated by products of linear forms. (This is called *pl-generated* in.[4]). This fact was used (but not established at the time) in the early development of the GPCA algorithm because the algorithm would be much easier to explain (to engineers) by using products of linear forms.

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

The reader needs to be aware that formula (11) for $h_I(i)$ is valid only for $i \geq n$. For $i < n$ there is no known closed-form formula for $h_I(i)$. One must resort to symbolic or numerical computation to find those values. Fortunately, for most practical applications that we have seen so far, it is typically good enough to know the values of $h_I(i)$ for fewer than 10 subspaces in an ambient space of dimension less than 15.[‡]

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

2.3. Generalized Principal Component Analysis

In this subsection, we introduce an efficient algebraic algorithm to retrieve a subspace arrangement and its individual subspaces from a given set of samples. This process is known as *generalized principal component analysis* (GPCA). In this section, we assume the samples to be noise-free. We will discuss samples corrupted by noise in Section 3.1 and samples contaminated by outliers in Section 3.2.

The first version of the algebraic GPCA algorithm was proposed in.[2] Several different variations have been proposed since then. All variants consist of three steps. First, a set of polynomials that vanish on the given data samples is retrieved. Second, the vectors normal to the subspaces are estimated from the derivatives of these polynomials. Third, the samples are segmented into their respective subspaces based on the normals. We give a brief description of each step.

2.3.1. Retrieving the Vanishing Polynomials

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

$$L_n \doteq \big(\nu_n(\boldsymbol{z}_1), \nu_n(\boldsymbol{z}_2), \ldots, \nu_n(\boldsymbol{z}_N)\big) \quad \in \mathbb{R}^{M_n^{[D]} \times N}. \tag{12}$$

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

$$Q(\boldsymbol{X}) \doteq \big(q_1(\boldsymbol{X}), q_2(\boldsymbol{X}), \ldots, q_m(\boldsymbol{X})\big)^T, \tag{13}$$

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

The matrix $\boldsymbol{C}$ can be computed from the eigenvectors of the matrix $W \doteq \frac{1}{N} L_n L_n^T \in \mathbb{R}^{M_n^{[D]} \times M_n^{[D]}}$ that correspond to its m eigenvalues with eigenvalue 0. In the case of small noise or numerical round-off errors, we can take the m eigenvectors associated with the m smallest eigenvalues. Numerically, this can be done via singular value decomposition (SVD) of L_n. Statistically, this the same as principal component analysis (PCA). In the next section, we will see how the estimate of $\boldsymbol{C}$ can be further improved when the samples are noisy.

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

DEFINITION 2.13 (JACOBIAN MATRIX). *The **Jacobian Matrix** of the polynomials* $Q(\boldsymbol{X}) = \big(q_1(\boldsymbol{X}), q_2(\boldsymbol{X}), \ldots, q_m(\boldsymbol{X})\big)^T$ *is the* $m \times D$ *matrix*

$$\mathcal{J}(Q)(\boldsymbol{X}) \doteq \begin{pmatrix} \frac{\partial q_1}{\partial X_1} & \cdots & \frac{\partial q_1}{\partial X_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial q_m}{\partial X_1} & \cdots & \frac{\partial q_m}{\partial X_D} \end{pmatrix} \quad \in \mathbb{R}^{m \times D}. \tag{14}$$

2.3.2. Retrieving the Normal Vectors and Bases of the Subspaces

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

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

[§]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.

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

2.3.3. The Algebraic GPCA Algorithm and Its Variations

These concepts about fitting polynomials and the Hilbert function lead to a natural algorithm for recovering hybrid linear models. We list this algorithm below.

Algorithm 1 (Generalized Principal Component Analysis).

Given a set of samples $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N\}$ from n linear subspaces of dimensions $d_1, d_2, \ldots, d_n$ in $\mathbb{R}^D$:

1: Construct the matrix $L_n = \big(\nu_n(\boldsymbol{z}_1), \nu_n(\boldsymbol{z}_2), \ldots, \nu_n(\boldsymbol{z}_N)\big)$.
2: Compute the singular value decomposition (SVD) of L_n and let $\boldsymbol{C}$ be the singular vectors associated with the $m = h_I(n)$ smallest singular values.
3: Construct the polynomials $Q(\boldsymbol{X}) = \boldsymbol{C}^T \nu_n(\boldsymbol{X})$.
4: **for all** $1 \leq i \leq n$ **do**
5: Pick one point $\boldsymbol{z}_i$ per subspace, and compute the Jacobian $\mathcal{J}(Q)(\boldsymbol{z}_i)$.
6: Compute a basis $B_i = \big(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{d_i}\big)$ 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**

3. STATISTICAL IMPROVEMENTS TO GPCA

In this section, we outline how to improve the algebraic GPCA algorithm to be stable in the presence of noisy data and robust to statistical outliers.

3.1. Dealing with Noise

In the algebraic GPCA algorithm, the basis of each subspace is computed as the orthogonal complement to the derivatives of the fitting polynomials at a representative sample point. However, if the chosen point is noisy, it may cause a large error in the estimated basis and subsequently cause a large error in the segmentation. From a statistical point of view, more accurate estimates of the basis can be obtained only if we are able to compute an average of the derivatives at many points in the same subspace. However, we do not know which points belong to the same subspace.

Suppose we know the number of subspaces n and their dimensions $d_1, d_2, \ldots, d_n$. For a given sample $\boldsymbol{z}$, we can look at the rank d_1 approximation of $\mathcal{J}(Q(\boldsymbol{z}))$ as an estimate of the bases for a d_1-dimensional subspace. Similarly we can look at the rank d_2 through d_n approximations of $\mathcal{J}(Q(\boldsymbol{z}))$ as estimates for subspaces. Among all samples, the true subspaces will be those that have the most consistency. Thus we can more reliably estimate the subspace parameters by choosing the n subspace candidates that most agree with estimates proposed by the samples.

From these notions, we have developed an algorithm to estimate subspace parameters from the fitting polynomials. The algorithm relies on a voting method on the feature space of subspace basis parameters, which was inspired by the classical Hough transform.[5,6] An important difference here is that we do not quantize the feature space of basis vector parameters, since it is impossible to store the whole quantized space in a computer when the dimensions of the subspaces are high.

We summarize the overall process as Algorithm 2.

There are important features about the above voting scheme that are quite different from the well-known statistical learning methods K-Subspaces[7] and EM[8] 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.

Algorithm 2 (Generalized Principal Component Analysis with Voting).

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

1: Suppose there are l distinct codimensions, ordered as $c'_1 < c'_2 < \cdots < c'_l$. Allocate $u_1, u_2, \ldots, u_l$ to be l stacks of counters and $U_1, U_2, \ldots, U_l$ be l stacks of candidate bases.
2: Estimate the set of fitting polynomials $Q(\boldsymbol{X})$, and compute their derivatives $\mathcal{J}(Q)(\boldsymbol{X})$ for all $\boldsymbol{z}_k$.
3: **for all** sample $\boldsymbol{z}_k$ **do**
4: **for all** $1 \leq i \leq l$ **do**
5: Assume $\boldsymbol{z}_k$ is drawn from a subspace of codimension c'_i. Find the first c'_i principal vectors of $\mathcal{J}(P)(\boldsymbol{z}_k)$ and stack them into the matrix $B_i(\boldsymbol{z}_k) \in \mathbb{R}^{D \times c'_i}$.
6: If $\langle B_i(\boldsymbol{z}_k), U_i(j) \rangle < \tau$ for some j, increase $u_i(j)$ by one and rescale $U_i(j)$. Otherwise, create a new candidate basis in U_i and a new counter in u_i with initial value one.
7: **end for**
8: **end for**
9: **for all** $1 \leq i \leq l$ **do**
10: Choose the highest vote(s) in u_i with their corresponding basis/bases in U_i.
11: Assign the samples to their closest subspaces, and remove their votes in other counters and bases of higher codimensions.
12: **end for**
13: Segment the remaining samples that are not in the stacks of the highest votes based on the estimated bases.

3.2. Estimation of Subspace Arrangements with Outliers

In many practical situations the sample points can be contaminated by some atypical samples known as "outliers" in addition to the noise that we have discussed above. In the robust statistics literature, there have been extensive studies about outlier detection and rejection.[9–15] Most of them are conducted with the assumption that the valid samples points, i.e., the *inliers*, are drawn from a conventional statistical (or geometric) model.

The application of any of the GPCA algorithms to valid samples contaminated with statistical outliers can lead to disastrous results. Both the estimated subspaces and the segmentation can be far from the ground truth. Thus, in this section we introduce some relevant *robust statistical techniques* that can detect or diminish the effect of outliers in estimating subspace arrangements.

Outlier samples corrupt the estimation of the fitting polynomials. We obtain the polynomials, essentially, by using PCA on in the Veronese space. Thus, one way of making the polynomial estimation robust is to replace PCA with a robust PCA process. Several method for robust PCA have been proposed in the literature.[9,16–18] We highlight one of these methods below and apply it to GPCA.

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 $d_i = \boldsymbol{u}_i^T \Sigma^{-1} \boldsymbol{u}_i$ 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*[20] and *multivariate trimming* (MVT)[21] . 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*.[9,20] Roughly speaking, for the M-estimator it is inversely proportional to the dimension of the samples, and it usually becomes prohibitive when the data dimension is higher than 20. For MVT, it is equal to the percentage of samples trimmed from the data set, which can be very high. The convergence rate of MVT is also the fastest among all methods of this kind. In the case of subspace arrangements, the dimension of $\boldsymbol{u} = \nu_n(\boldsymbol{z})$, i.e. $M_n^{[D]}$, is normally very high. Thus, the M-estimator becomes impractical and MVT becomes the method of choice.

The MVT method proceeds as follows. As the random vector $\nu_n(\boldsymbol{z})$ is not necessarily zero mean, we first obtain a robust estimate of the mean $\bar{\boldsymbol{u}}$ of the samples $\{\boldsymbol{u}_i = \nu_n(\boldsymbol{z}_i)\}$ (using techniques such as in[21]). 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 $\Sigma_0 = \frac{1}{|U|-1}\sum_{h\in U}(\boldsymbol{u}_h - \bar{\boldsymbol{u}})(\boldsymbol{u}_h - \bar{\boldsymbol{u}})^T$, 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. Algorithm 3 gives an outline of the resulting algorithm, which we refer to as Robust Generalized Principal Component Analysis (RGPCA).

Algorithm 3 (Robust Generalized Principal Component Analysis).

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

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

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

4. SIMULATIONS

4.1. Simulations with Noisy Data

We provide a comparison of various algorithms for the estimation and segmentation of subspace arrangements in the presence of noise. They include: The EM algorithm, the K-Subspaces algorithm, the algebraic GPCA algorithm, the GPCA algorithm with voting, as well as some combination of them.

We randomly generate some subspace arrangements of some pre-chosen dimensions. For instance, $(2, 2, 1)$ indicates an arrangement of three subspaces of dimensions $2, 2, 1$, respectively. We then randomly draw a set of samples from them. The samples are corrupted with Gaussian noises. Here we choose the level of noise to be 4%. The error is measured in terms of the percentage of sample points that are wrongfully grouped.[¶] All cases are averaged over 100 trials. The performance of all the algorithms are compared in Table 4.1. The reader can download the MATLAB codes from our website.

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

4.2. Simulations with Statistical Outliers

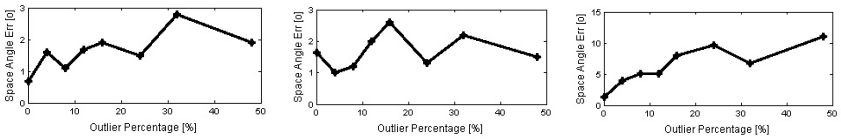
To demonstrate the performance of the algorithm, we conduct three simulated experiments: 1. Three subspaces with dimensions $(2, 2, 1)$ in $\mathbb{R}^3$ and sample sizes $(200, 200, 100)$. 2. Four subspaces with dimensions $(4, 2, 2, 1)$ in $\mathbb{R}^5$ and sample sizes $(400, 200, 200, 100)$. 3. Three subspaces with dimensions $(5, 5, 5)$ in $\mathbb{R}^6$ and sample sizes $(600, 600, 600)$.

[¶]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.

Table 1. The percentage of sample points mis-grouped by different algorithms. The number of subspaces and their dimensions are given to all algorithms. The EM and K-Subspaces algorithms are randomly initialized. "GPCA-Voting+K-Subspaces" means the K-Subspaces method initialized with the GPCA-Voting algorithm. The sample number for each subspace is 200 times its dimension.

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

The maximal data magnitude is 1, and it is corrupted with 6% Gaussian noise and uniformly distributed outliers of a series of percentages between 0% to 48%. The experiment is repeated 100 times at each percentage. In the experiment, the residual threshold is fixed at 0.05. Figure 1 shows the results of the average angle error. Table 2 shows the average time of the algorithm on a dual 2.7G Hz Macintosh workstation.



(a) $(2,2,1)$ in $\mathbb{R}^3$ (b) $(4,2,2,1)$ in $\mathbb{R}^5$ (c) $(5,5,5,)$ in $\mathbb{R}^6$

Figure 1. Average space angle errors (in degree) of arrangements estimated by RGPCA.

Table 2. Space angle errors and average time for estimating the three subspace arrangements with 20% uniform distributed outliers via RGPCA.

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

5. APPLICATIONS

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.

5.1. Motion Segmentation in Computer Vision

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

Of course, the problem depends on how the 3-D points $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots \boldsymbol{X}_N$ are projected onto the image plane (i.e., the camera model) and what class of motions we consider for $\boldsymbol{X}_j$ or for $\boldsymbol{z}_{ij}$ (i.e., the 3-D or 2-D motion models). Nevertheless, it has been shown that the motion segmentation problem can be converted to a subspace-segmentation problem for most motion models that have been considered in computer vision.[22] 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[‖] from $\mathbb{R}^3$ to $\mathbb{R}^2$:

$$\boldsymbol{z}_{ij} = A_i \boldsymbol{X}_j + b_i \in \mathbb{R}^2 \quad \text{for all } i = 1, 2, \ldots, F, \tag{15}$$

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

$$W \doteq \begin{bmatrix} \boldsymbol{z}_{11} \cdots \boldsymbol{z}_{1N} \\ \vdots \quad\quad \vdots \\ \boldsymbol{z}_{F1} \cdots \boldsymbol{z}_{FN} \end{bmatrix}_{2F\times N} = \begin{bmatrix} A_1 & b_1 \\ \vdots & \vdots \\ A_F & b_F \end{bmatrix}_{2F\times 4} \begin{bmatrix} \boldsymbol{X}_1 \cdots \boldsymbol{X}_N \\ 1 \quad \cdots \quad 1 \end{bmatrix}_{4\times N}. \tag{16}$$

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

For multiple moving objects, it can be shown under mild conditions that the trajectories of their image points span different subspaces in $\mathbb{R}^{2F}$. Thus, if we view the columns of W as the sample points, then these sample points belong to multiple subspaces (of dimension less than 5) in $\mathbb{R}^{2F}$. In the computer vision literature, many algorithms have been developed to solve the problem of segmenting the points to their respective subspaces, see[23–25] and references therein.

We first give the experimental results of GPCA-Voting (Algorithm 2) on two outdoor sequences taken by a moving camera tracking a car moving in front of a parking lot and a building (sequences A and B), and one indoor sequence taken by a moving camera tracking a person moving his head (sequence C), as shown in Figure 2. The data for these sequences are borrowed from,[26] 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`.

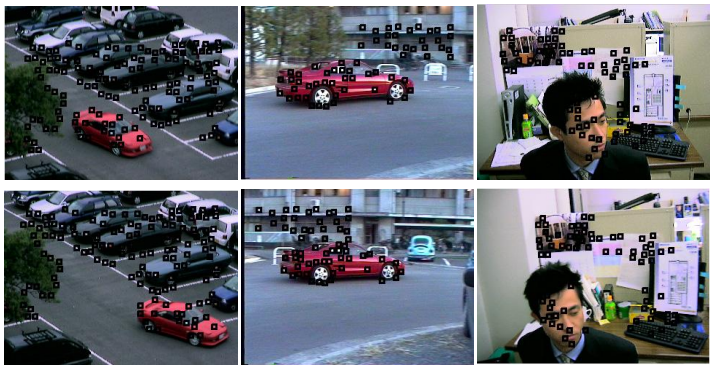


Figure 2. The first and last frames of sequences A (left), B (middle) and C (right) with point correspondences superimposed.

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

$$\boldsymbol{z}_j = [\boldsymbol{z}_{1j}^T, \boldsymbol{z}_{2j}^T, \ldots, \boldsymbol{z}_{Fj}^T]^T \in \mathbb{R}^{2F}, \quad j = 1, 2, \ldots, N, \tag{17}$$

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 3. The table also shows results reported in[26] from other existing *multiframe* algorithms for motion segmentation.

Next, we demonstrate the performance of RGPCA (Algorithm 3) on sequence A and C with original tracking outliers added in, as shown in Figure 3. The data are borrowed from,[25] which are also available at the website: `http://www.suri.it.okayama-u.ac.jp/e-program-separate.html`. The reported outlier percentages in[25] were 1.4% and 30%, respectively. We use Algorithm 3 to segment two hyperplanes of dimension 4 in both sequences. For both cases, the angle tolerance is fixed at 0.3 rad, and the boundary tolerance is fixed at 0.065. The segmentation results are shown in Figure 4. The RGPCA algorithm achieves perfect segmentation with the rejection rate at 0% and 18% for sequence A and C, respectively, which outperforms the results reported in.[25]

Table 3. Classification rates given by various subspace segmentation algorithms for sequences A, B, C.

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

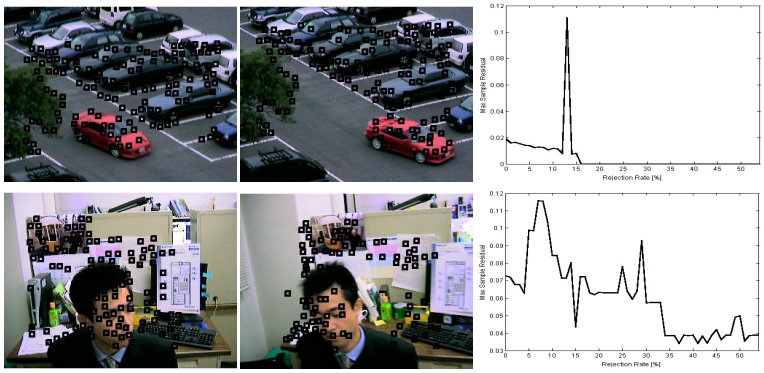
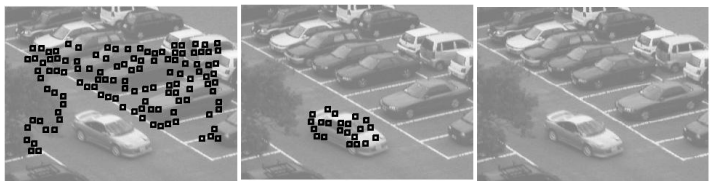


Figure 3. The first (left) and last (middle) frames of sequences A and C with the original tracking outliers. The right column shows the maximal residual values of the two sequences with various rejection rates using the MVT algorithm. Sequence A contains 140 feature points, and sequence B contains 107 feature points.

In sequence A, the camera is far away from the scene, so the projection relation is well described by the affine camera model (15), which also results in very small sample residuals in Figure 3. 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

[‖] 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.



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

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

Figure 4. Segmentation results of sequences A and C. Left: group 1. Middle: group 2. Right: outliers.

well the actual camera projection. Furthermore, the motion of the man's upper body is non-rigid, which leads to outliers on the face and shoulders. These observations are consistent with the result of the maximal sample residuals shown in Figure 3. 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.[25] To make the comparison complete, we show the segmentation result of RGPCA with 30% rejection rate using MVT in Figure 5. Although more samples are trimmed as outliers, the algorithm still gives good segmentation on the inlying samples.

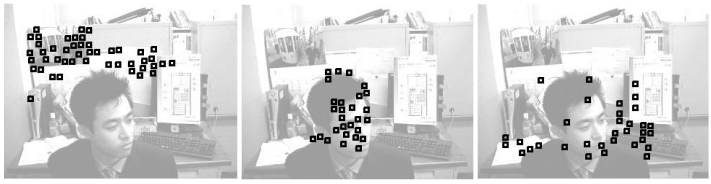


Figure 5. Segmentation results of C with 30% rejection rate. Left: group 1. Middle: group 2. Right: outliers.

5.2. Hybrid Linear Representation of Images

An important problem in image processing is to find efficient and sparse representations of images (rather than the original bitmaps). Such representations are often the first step for many subsequent processes of the images: compression, classification, retrieval, and synthesis, etc. A popular and still dominant approach to represent images is to transform the images via certain linear transformations so that the energy of the image will be concentrated in the coefficients of a sparse set of bases.

A linear transformation can be either *pre-fixed* for all images (such as the discrete cosine transform used for the JPEG standard and the wavelet transform for the JPEG2000 standard) or *adaptive* for each image (such as the Karhunen-Loève transform that is equivalent to PCA). However, natural images typically exhibit multi-modal statistics as they usually contain many heterogeneous regions with significantly different geometric or statistical characteristics, loosely known as "textures." Such heterogeneous data can be better-represented using a mixture of linear models, one for each a homogeneous subset.

Obviously, the same assumptions can be made for any transformed image, say a sub-sampled version of the image and its residuals. Figure 6 left shows a three-level representation of the baboon image in terms of a (twice) sub-sampled version and its residuals at two higher level. Figure 6 right shows the segmentation of the sub-sampled image and its residuals according to the subspaces of their associated hybrid linear models. Using a slight variation of the GPCA algorithm, the number and dimensions of the subspaces of each hybrid linear model are found automatically in such a way that they minimize the effective dimension of the imagery data subject to a given error threshold. For more details about the algorithmic implementation, the reader may refer to.[27]

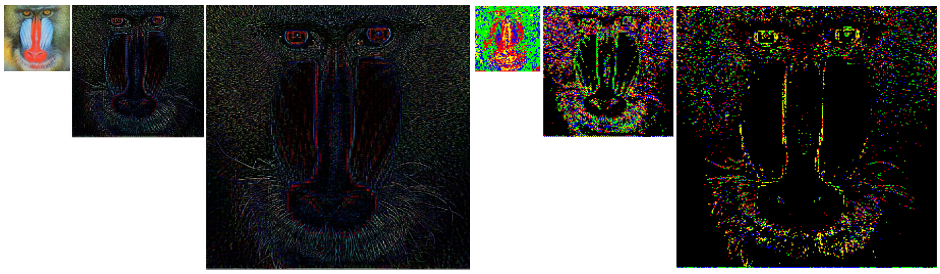


Figure 6. A multi-scale representation of the Baboon image. Left: Twice sub-sampled image and its residuals at two higher levels. Right: The segmentation of vectors (2×2 blocks) by a hybrid linear model at each level – different subspaces are denoted by different colors. The black regions correspond to data vectors whose energy is below a given error threshold.

Typically, such a multi-scale scheme can achieve a more compact representation because it extracts low-frequency parts of the image first.** Figure 7 gives a comparison of the efficiency of different lossy image representation schemes for the baboon image: the discrete cosine transform (DCT), the Karhunen-Loéve transform (KLT), hybrid linear model (without

**The energy of typical natural images concentrates more in low frequencies.

sub-sampling), the level-3 biorthogonal 4.4 wavelets (used in JPEG2000), and the level-3 multi-scale hybrid linear model.[††]

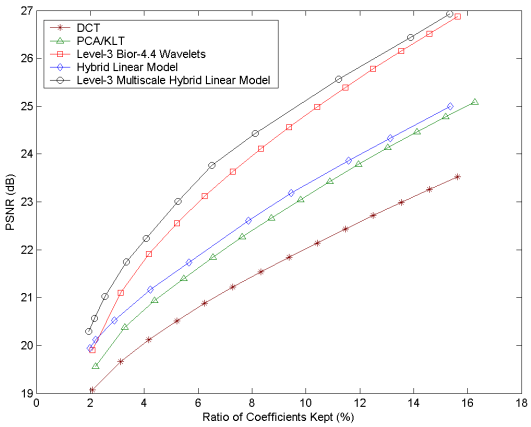


Figure 7. Comparison of several lossy image representation schemes for the baboon image. Vertical axis: here the signal is the original image and the noise is the difference between the original image and its approximation given by different representation schemes; Horizontal axis: percentage of the ratio of coefficients kept.

Potentially, there might be many other ways of applying the (multi-scale) hybrid linear model to images that could achieve even better performance. In fact, a higher PSNR curve can be achieved for the baboon image if we apply the multi-scale hybrid linear model in the wavelet domain (see[27]).

6. CONCLUSION

In this paper we investigated how to apply the theory of subspace arrangements to the subspace segmentation problem. Based on the algebraic and statistical properties of subspace arrangements, a set of new computational tools have been developed for the modeling and segmenting mixed data. One important feature of these tools is that they take a "top-down" approach to the estimation of multiple subspaces. That is, the overall algebraic structure of the data set is found first and then the geometric information of the individual subspaces and segmentation of the data are subsequently retrieved. This runs somewhat contrary to the conventional approach taken by existing data clustering methods in statistical learning, such as the EM and the K-means. As a consequence, the resulting algorithms, GPCA and its variations, require no initialization and can be used in combination with EM and K-means.

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

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

In many scientific studies, the structure of the data can be modeled as a low-dimensional (nonlinear) manifold embedded in a high-dimensional space. Many algorithms have been proposed to identify such a manifold.[28,29] GPCA provides yet another class of tools that allow us to obtain a piece-wise linear approximation of the manifold (subject to an error threshold). Important geometric or topical properties, e.g. dimension(s), components, of the manifold can be extracted from such an approximation.

[††]The experimental results given here are attributed to John Wright and Wei Hong of the University of Illinois at Urbana-Champaign.

REFERENCES

1. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, N.Y., 1995.
2. R. Vidal, Y. Ma, and J. Piazzi, "A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing Polynomials," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 510–517, 2004.
3. H. Derksen, "Hilbert series of subspace arrangements," 2005 (preprint, math arXiv, math.AC/0510584).
4. A. Björner, I. Peeva, and J. Sidman, "Subspace arrangements defined by products of linear forms," *Journal of the London Math. Society* , pp. 273–288, 2005.
5. D. Ballard, "Generalizing the Hough transform to detect arbitrary patterns," *Pattern Recognition* **13**(2), pp. 111–122, 1981.
6. W. Tong, C. Tang, and G. Medioni, "Simultaneous two-view epipolar geometry estimation and motion segmentation by 4D tensor voting," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(9), pp. 1167–1184, 2004.
7. J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2003.
8. G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, John Wiley & Sons, 1997.
9. F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel, *Robust statistics: the approach based on influence functions*, John Wiley & Sons, 1986.
10. V. Barnett and T. Lewis, *Outliers in Statistical Data*, John Wiley & Sons, second ed., 1983.
11. P. J. Bickel, "Another look at robustness: A review of reviews and some new developments," *Scand. J. Statist.* **3**(28), pp. 145–168, 1976.
12. M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM* **24**, pp. 381–85, 1981.
13. I. Jollife, *Principal Component Analysis*, Springer-Verlag, 2nd ed., 2002.
14. P. Torr and D. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *International Journal on Computer Vision* **24**(3), pp. 271–300, 1997.
15. C. Stewart, "Robust parameter estimation in computer vision," *SIAM Review* **41**(3), pp. 513–537, 1999.
16. N. Campbell, "The influence function as an aid in outlier detection in discriminant analysis," *Applied Statistics* **27**(3), pp. 251–258, 1978.
17. F. Critchley, "Influence in principal components analysis," *Biometrika* **72**(3), pp. 627–636, 1985.
18. P. Rousseeuw, "Least median of squares regression," *Journal of American Statistics Association* **79**, pp. 871–880, 1984.
19. T. Ferguson, "On the rejection of outliers," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, 1961.
20. P. Huber, *Robust statistics*, John Wiley & Sons, 1981.
21. R. Gnanadesikan and J. Kettenring, "Robust estimates, residuals, and outlier detection with multiresponse data," *Biometrics* **28**(1), pp. 81–124, 1972.
22. 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.
23. J. Costeira and T. Kanade, "A multibody factorization method of independently moving objects," *International Journal on Computer Vision* **29**(3), pp. 159–179, 1998.
24. K. Kanatani, "Motion segmentation by subspace separation: Model selection and reliability evaluation," *International Journal of Image and Graphics* **2**(2), pp. 179–197, 2003.
25. Y. Sugaya and K. Kanatani, "Outlier removal for motion tracking by subspace separation," *IEICE Transactions on Information and Systems* **E86-D**(6), pp. 1095–1102, 2003.
26. K. Kanatani and Y. Sugaya, "Multi-stage optimization for multi-body motion segmentation," in *Proceedings of the Australia-Japan Advanced Workshop on Computer Vision*, 2003.
27. W. Hong, J. Wright, K. Huang, and Y. Ma, "A multi-scale hybrid linear model for lossy image representation," in *Proceedings of IEEE International Conference on Computer Vision*, 2005.
28. J. Tenenbaum, V. de Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Demensionality Reduction," *SCIENCE* **290**, pp. 2319–2323, 2000.
29. S. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *SCIENCE* **290**, pp. 2323–2326, 2000.