Generalized Principal Component Analysis (GPCA)[*]

René Vidal[†] Yi Ma[‡] Shankar Sastry[†]
[†]Department of EECS, University of California, Berkeley, CA 94720
[‡]ECE Department, University of Illinois, Urbana, IL 61801

Abstract

We propose an algebraic geometric approach to the problem of estimating a mixture of linear subspaces from sample data points, the so-called Generalized Principal Component Analysis *(GPCA) problem. In the absence of noise, we show that GPCA is equivalent to factoring a homogeneous polynomial whose degree is the number of subspaces and whose factors (roots) represent normal vectors to each subspace. We derive a formula for the number of subspaces n and provide an analytic solution to the factorization problem using linear algebraic techniques. The solution is closed form if and only if $n \leq 4$. In the presence of noise, we cast GPCA as a constrained nonlinear least squares problem and derive an optimal function from which the subspaces can be directly recovered using standard nonlinear optimization techniques. We apply GPCA to the motion segmentation problem in computer vision, i.e. the problem of estimating a mixture of motion models from 2-D imagery.*

1. Introduction

Principal Component Analysis (PCA) [4] refers to the problem of estimating a linear subspace $S \subset \mathbb{R}^K$ of unknown dimension $k < K$ from N sample points $\{\boldsymbol{x}^j \in S\}_{j=1}^N$. This problem shows up in a variety of applications in many fields, e.g., pattern recognition, data compression, image analysis, regression, etc., and can be solved in a remarkably simple way from the singular value decomposition (SVD) of the data matrix $[\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N] \in \mathbb{R}^{K \times N}$.

Extensions of PCA include probabilistic PCA [9, 2], where the subspace is estimated in Maximum Likelihood sense using a probabilistic generative model, and nonlinear PCA (NLPCA) [6], where the subspace is estimated after applying a nonlinear embedding to the data. In this paper, we consider an alternative generalization called Generalized Principal Component Analysis (GPCA), in which the sample points $\{\boldsymbol{x}^j \in \mathbb{R}^K\}_{j=1}^N$ are drawn from n k-dimensional linear subspaces of $\mathbb{R}^K$, $\{S_i\}_{i=1}^n$, as illustrated in Figure 1 for $n = 3, k = 2$, and $K = 3$. In this case, the problem becomes one of identifying each subspace without knowing which sample points belong to which subspace.[1]

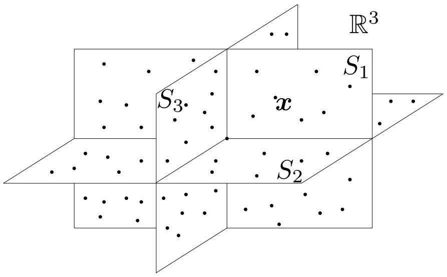


Figure 1: GPCA for $n = 3$, $k = 2$ and $K = 3$. Identifying three 2-dimensional subspaces S_1, S_2, S_3 in $\mathbb{R}^3$ from sample points $\{\boldsymbol{x}\}$ drawn from these subspaces.

Geometric approaches to mixtures of principal components have been proposed in the computer vision community in the context of 3-D motion segmentation. The main idea is to first segment the data associated with each subspace, and then apply standard PCA to each group. In [5] (see also [1, 3]) it was shown that if the pairwise intersection of the subspaces is trivial, which implies that $K \geq nk$, one can use the SVD of all the data to build a similarity matrix from which the segmentation can be easily extracted. When the subspaces *do* intersect, the segmentation of the data is usually obtained in an ad-hoc fashion using various clustering algorithms, e.g., K-means. An alternative algebraic solution for the case of two planes in $\mathbb{R}^3$ was proposed in [7] in the context of 2-D segmentation of transparent motions.[2] See also [14] for 3-D segmentation of two rigid motions.

Probabilistic approaches to mixtures of principal components [8] assume that sample points within each subspace are drawn from an unknown probability distribution. The membership of the data points to each one of the subspaces is modeled with a multinomial distribution whose parameters are referred to as the mixing proportions. The parameters of this mixture model are estimated in a Maximum Likelihood or Maximum a Posteriori framework as follows: one first estimates the mixing proportions given a current estimate for the subspaces, and then estimates the subspaces given a current estimate of the mixing proportions. This is usually done in an iterative manner using the Expectation Maximization (EM) algorithm. Unfortunately, EM is

[*]Research supported by grants ONR N00014-00-1-0621 and DARPA F33615-98-C-3614, and by UIUC ECE Department startup fund.

[1] If the association between sample points and subspaces was known, then the problem would reduce to standard PCA applied to each subspace.

[2] The authors thank Dr. David Fleet for pointing out this reference.

in general sensitive to initialization and may not converge to the global optimum. Another disadvantage is that it is hard to analyze some theoretical questions such as the existence and uniqueness of a solution to the problem. Also there are many cases in which it is hard to solve the grouping problem correctly, yet it is possible to obtain a precise estimate of the subspaces. In such cases a direct estimation of the subspaces (without grouping) seems more appropriate than an estimation based on incorrectly segmented data.

1.1. Contributions of this paper

We propose a novel algebraic geometric approach to mixtures of principal components, the so-called *Generalized Principal Component Analysis* (GPCA) problem.

In the absence of noise, we cast GPCA in an algebraic geometric framework in which the number of subspaces becomes the degree of a certain polynomial and the normals to each subspace become the factors (roots) of such a polynomial. We show that the number of subspaces n can be obtained from the rank of a certain matrix that depends on the data. Given n, the estimation of the subspaces $S_i \subset \mathbb{R}^K$ is essentially equivalent to a factorization problem in the space of homogeneous polynomials of degree n in K variables. We prove that the factorization problem has a unique solution which can be obtained from the roots of a polynomial of degree n in one variable and from the solution of $K-2$ linear systems in n variables. Hence, the solution is closed form when $n \leq 4$. Unlike previous work, GPCA allows for *arbitrary intersections* among an *arbitrary number* of different subspaces and *does not* require previous knowledge of the segmentation of the data or the number of subspaces. In fact, the subspaces are estimated directly using *segmentation independent constraints* that are satisfied by *all* the points, regardless of the subspace to which they belong.

In the presence of noise, we cast GPCA as a constrained nonlinear least squares problem that minimizes the error between the noisy points and their projections onto the subspaces. By converting this constrained problem into an unconstrained one, we obtain an optimal function from which the subspaces can be directly recovered using standard nonlinear optimization techniques. We show that the optimal objective function is just a *normalized* version of the algebraic error minimized by our analytic solution to GPCA. Although this means that the analytic solution to GPCA may be sub-optimal in the presence of noise, we can still use it as a global initializer for our nonlinear algorithm or any other iterative algorithm, such as K-means or EM.

Our solution to GPCA can be applied to various estimation problems in which the data comes simultaneously from multiple (approximately) linear models. In this paper, we apply GPCA to the motion segmentation problem in computer vision, i.e. the problem of estimating a mixture of motion models from 2-D imagery. Applications to segmentation of static and dynamic textures are forthcoming [10].

2 Problem formulation and analysis

In this paper, we consider the following generalization of principal component analysis (PCA).

Problem 1 (Generalized Principal Component Analysis) *Given a set of sample points $\boldsymbol{X} = \{\boldsymbol{x}^j \in \mathbb{R}^K\}_{j=1}^N$ drawn from $n > 1$ distinct linear subspaces $\{S_i \subseteq \mathbb{R}^K\}_{i=1}^n$ of dimension k, $0 < k < K$, identify each subspace S_i without knowing which sample points belong to which subspace. By identifying the subspaces we mean the following:*

1. *Identify the number of subspaces and their dimension;*
2. *Identify a basis for each subspace S_i (or for $S_i^\perp$);*
3. *Group or segment the given N data points into the subspace(s) they belong to;*

In our analysis of the GPCA problem, we will distinguish between the following two cases: The general case of subspaces of *unknown* dimension k, where $0 < k < K-1$, and the special case of hyperplanes of *known* dimension $k = K-1$. It turns out that the general case can always be reduced to the special case, as long as all the subspaces $\{S_i\}_{i=1}^n$ have the same dimension k (see [10] for the proof). This is because, from a geometric point of view, the segmentation of a sample set $\boldsymbol{X}$ drawn from n k-dimensional subspaces of a space of dimension $K > k$ is preserved after projecting the sample set $\boldsymbol{X}$ onto a *generic*[3] subspace P of dimension $k+1$ ($\leq K$). An example is shown in Figure 2, where two lines L_1 and L_2 in $\mathbb{R}^3$ are projected onto a plane P not orthogonal to the plane containing the lines.

In general, there are various technical details involved in reducing the general case $0 < k < K$ to the special case $k = K-1$, e.g., how to determine the dimension of the subspaces k and how to choose the $(k+1)$-dimensional subspace P. We refer the reader to [10] for all those details, and concentrate on the special but important case $k = K-1$ from now on.

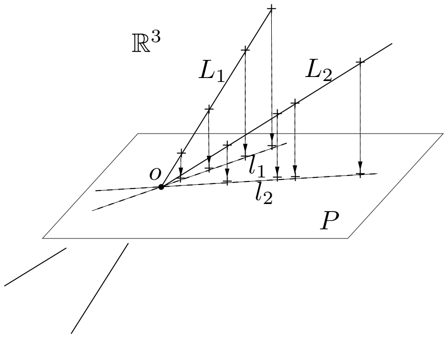


Figure 2: Samples on two 1-dimensional subspaces L_1, L_2 in $\mathbb{R}^3$ projected onto a 2-dimensional plane P. The membership of each sample is preserved through the projection.

[3] By generic we mean except for a zero-measure set $\{P\}$ of subspaces.

3. A solution to GPCA with $k = K - 1$

In this section, we give the following constructive solution to the GPCA problem in the case $k = K - 1$.

Theorem 1 (GPCA with $k = K - 1$) *The GPCA problem with $k = K - 1$ is algebraically equivalent to factoring a homogeneous polynomial of degree n in K variables into a product of n polynomials of degree 1. The factorization problem can be solved from the roots of a polynomial of degree n in one variable plus $K - 2$ linear systems in n variables. Thus GPCA with $k = K - 1$ has a unique solution, which can be obtained in closed form if and only if $n \leq 4$.*

We establish the equivalence between GPCA and polynomial factorization in Section 3.1. We show how to estimate the number of subspaces in Section 3.2 and give an analytic solution to the factorization problem in Section 3.3. We summarize the overall algorithm in Section 3.4 and present an optimal algorithm in the presence of noise in Section 3.5.

3.1. GPCA and polynomial factorization

We notice that every $(K - 1)$-dimensional space $S_i \subset R^K$ can be represented by a nonzero *normal* vector $\boldsymbol{b}_i \in \mathbb{R}^K$ as

$$S_i = \{\boldsymbol{x} \in \mathbb{R}^K : \boldsymbol{b}_i^T \boldsymbol{x} = b_{i1}x_1 + b_{i2}x_2 + \ldots + b_{iK}x_K = 0\}.$$

Since the subspaces S_i are all distinct from each other, the normal vectors $\{\boldsymbol{b}_i\}_{i=1}^n$ are pairwise linearly independent.

Imagine that we are given a point $\boldsymbol{x} \in \mathbb{R}^K$ lying on one of the subspaces S_i. Such a point must satisfy the formula:

$$(\boldsymbol{b}_1^T \boldsymbol{x} = 0) \ \vee \ (\boldsymbol{b}_2^T \boldsymbol{x} = 0) \ \vee \cdots \vee \ (\boldsymbol{b}_n^T \boldsymbol{x} = 0), \qquad (1)$$

which is equivalent to the following homogeneous polynomial of degree n in $\boldsymbol{x}$ with real coefficients:

$$\boxed{p_n(\boldsymbol{x}) = \prod_{i=1}^{n} (\boldsymbol{b}_i^T \boldsymbol{x}) = 0.} \qquad (2)$$

The problem of identifying each subspace S_i is then equivalent to that of solving for the vectors $\{\boldsymbol{b}_i\}_{1=1}^n$ from the *nonlinear* equation (2). A standard technique used in algebra to render a nonlinear problem into a linear one is to find an "embedding" that lifts the problem into a higher-dimensional space. Let $R_n(K) = R_n[x_1, \ldots, x_K]$ be the set of all homogeneous polynomials of degree n in K variables. We notice that each $R_n(K)$, can be made into a vector space under the usual addition and scalar multiplication. Furthermore, $R_n(K)$ is generated by the set of monomials $\boldsymbol{x}^{\boldsymbol{n}} = x_1^{n_1} x_2^{n_2} \cdots x_K^{n_K}$, with $0 \leq n_j \leq n$, $j = 1, \ldots, K$, and $n_1 + n_2 + \cdots + n_K = n$. Since there are a total of

$$M_n = \binom{n + K - 1}{K - 1} = \binom{n + K - 1}{n} \qquad (3)$$

different monomials, the dimension of $R_n(K)$ as a vector space is M_n. Therefore, we can define the following embedding (or lifting) from $\mathbb{R}^K$ into $\mathbb{R}^{M_n}$.

Definition 1 (Veronese map) *Given n and K, the* Veronese map *of degree n, $\nu_n : \mathbb{R}^K \to \mathbb{R}^{M_n}$, is defined as:*

$$\nu_n : \ [x_1, \ldots, x_K]^T \mapsto [\ldots, \boldsymbol{x}^{\boldsymbol{n}}, \ldots]^T, \qquad (4)$$

where $\boldsymbol{x}^{\boldsymbol{n}}$ is a monomial of the form $x_1^{n_1} x_2^{n_2} \cdots x_K^{n_K}$ with $\boldsymbol{n}$ chosen in the degree-lexicographic order.

With the so-defined Veronese map (also known as the *polynomial embedding*), equation (2) becomes the following linear expression in the vector of coefficients $\boldsymbol{c} \in \mathbb{R}^{M_n}$:

$$p_n(\boldsymbol{x}) = \nu_n(\boldsymbol{x})^T \boldsymbol{c} = \sum c_{n_1, \ldots, n_K} x_1^{n_1} \cdots x_K^{n_K} = 0, \quad (5)$$

where $c_{\boldsymbol{n}} \in \mathbb{R}$ represents the coefficient of the monomial $\boldsymbol{x}^{\boldsymbol{n}}$.

Example 1 *The case $n = 2$ and $K = 2$ corresponds to segmenting two lines in $\mathbb{R}^2$. These two lines are represented by the polynomial $p_2(\boldsymbol{x}) = (b_{11}x_1 + b_{12}x_2)(b_{21}x_1 + b_{22}x_2)$. In this case the Veronese map is $\nu_2(\boldsymbol{x}) = [x_1^2, x_1x_2, x_2^2]^T$ and the coefficients are $\boldsymbol{c} = [\underbrace{b_{11}b_{21}}_{c_{2,0}}, \underbrace{b_{11}b_{22} + b_{12}b_{21}}_{c_{1,1}}, \underbrace{b_{12}b_{22}}_{c_{0,2}}]^T$. Given $\boldsymbol{c}$, the slope of each line can be immediately computed from the roots of the polynomial $c_{2,0}w^2 + c_{1,1}w + c_{0,2} = 0$.*

3.2. Estimation of the number of subspaces n

Applying equation (5) to a given collection of $N \geq M_n - 1$ sample points $\{\boldsymbol{x}^j\}_{j=1}^N$ gives the following system of linear equations on the vector of coefficients $\boldsymbol{c}$

$$L_n \ \boldsymbol{c} \doteq \begin{bmatrix} \nu_n(\boldsymbol{x}^1)^T \\ \nu_n(\boldsymbol{x}^2)^T \\ \vdots \\ \nu_n(\boldsymbol{x}^N)^T \end{bmatrix} \boldsymbol{c} = 0 \in \mathbb{R}^N. \qquad (6)$$

Since the above linear system (6) depends explicitly on the number of subspaces n, we cannot estimate $\boldsymbol{c}$ directly without knowing n in advance. It turns out that the estimation of the number of subspaces n is very much related to the conditions under which the solution for $\boldsymbol{c}$ is unique (up to a scale factor), as stated by the following proposition.

Proposition 1 (Number of subspaces) *Assume that a collection of $N \geq M_n - 1$ sample points $\{\boldsymbol{x}^j\}_{j=1}^N$ on n different $(K - 1)$-dimensional subspaces of $\mathbb{R}^K$ is given. Let $L_i \in \mathbb{R}^{N \times M_i}$ be the matrix defined in (6), but computed with the Veronese map $\nu_i(\boldsymbol{x})$ of degree i. If the sample points are in general position and at least $K - 1$ points correspond to each subspace, then:*

$$\text{rank}(L_i) \begin{cases} > M_i - 1, & i < n, \\ = M_i - 1, & i = n, \\ < M_i - 1, & i > n. \end{cases} \qquad (7)$$

Therefore, the number n of subspaces is given by:

$$\boxed{n = \min\{i : \text{rank}(L_i) = M_i - 1\}.} \qquad (8)$$

The intuition behind Proposition 1 (see [10] for the proof) is that there is no polynomial of degree $i < n$ that is satisfied by all the data, hence $\text{rank}(L_i) = M_i$ for $i < n$. Conversely, there are multiple polynomials of degree $i > n$, namely any multiple of $p_n(\boldsymbol{x})$, which are satisfied by all the data, hence $\text{rank}(L_i) < M_i - 1$ for $i > n$. Thus the case $i = n$ is the only one in which system (6) has a unique solution, namely the coefficients $\boldsymbol{c}$ of the polynomial $p_n(\boldsymbol{x})$.

Remark 1 *In the presence of noise, one cannot directly estimate n from (8), because the matrix L_i is always full rank. In practice we declare the rank of L_i to be r if $\sigma_{r+1}/(\sigma_1 + \cdots + \sigma_r) < \epsilon$, where σ_k is the k-th singular value of L_i and $\epsilon > 0$ is a pre-specified threshold. We have found this simple criterion to work well in our experiments.*

Remark 2 *In the case of n subspaces of arbitrary dimension k, where $0 < k < K$, one can derive a rank condition similar to (8) from which one can jointly estimate n and k. Such a rank condition in GPCA is a natural generalization of the rank condition $k = rank(L_1)$ in standard PCA [10].*

3.3. Estimation of the subspaces $\{S_i\}_{i=1}^n$

Proposition 1 and the linear system in equation (6) allow us to determine the number of subspaces n and the vector of coefficients $\boldsymbol{c}$, respectively, from sample points $\{\boldsymbol{x}^j\}_{i=1}^N$. The rest of the problem becomes now how to recover the normal vectors $\{\boldsymbol{b}_i\}_{i=1}^n$ from $\boldsymbol{c}$. From (2) and (5) we have

$$p_n(\boldsymbol{x}) = \sum c_{\boldsymbol{n}} \boldsymbol{x}^{\boldsymbol{n}} = \prod_{i=1}^{n} \left(\sum_{j=1}^{K} b_{ij} x_j \right).$$

Therefore, recovering $\{\boldsymbol{b}_i\}_{i=1}^n$ from $\boldsymbol{c}$ is equivalent to factoring a given homogeneous polynomial $p_n(\boldsymbol{x}) \in R_n(K)$, into n *distinct* polynomials in $R_1(K)$.[4]

We now present a polynomial factorization algorithm that recovers the $\boldsymbol{b}_i$'s from $\boldsymbol{c}$. For simplicity, we will first present an example with the case of two planes in $\mathbb{R}^3$, i.e. $n = 2$ and $K = 3$, because it gives most of the intuition about our general algorithm for arbitrary n and K.

Example 2 *Consider the case $n = 2$ and $K = 3$. Then*

$$\begin{aligned} p_2(\boldsymbol{x}) &= (b_{11}x_1 + b_{12}x_2 + b_{13}x_3)(b_{21}x_1 + b_{22}x_2 + b_{23}x_3) \\ &= (\boldsymbol{b}_1^T \boldsymbol{x})(\boldsymbol{b}_2^T \boldsymbol{x}) = [x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2]\boldsymbol{c} = \\ & \underbrace{(b_{11}b_{21})}_{c_{2,0,0}} x_1^2 + \underbrace{(b_{11}b_{22}+b_{12}b_{21})}_{c_{1,1,0}} x_1x_2 + \underbrace{(b_{11}b_{23}+b_{13}b_{21})}_{c_{1,0,1}} x_1x_3 \\ & + \underbrace{(b_{12}b_{22})}_{c_{0,2,0}} x_2^2 + \underbrace{(b_{12}b_{23}+b_{13}b_{22})}_{c_{0,1,1}} x_2x_3 + \underbrace{(b_{13}b_{23})}_{c_{0,0,2}} x_3^2. \end{aligned}$$

We notice that the last three terms $c_{0,2,0}x_2^2 + c_{0,1,1}x_2x_3 + c_{0,0,2}x_3^2$ correspond to a polynomial in x_2 and x_3 only, which is equal to the product of the last two terms of the original factors, i.e. $(b_{12}x_2 + b_{13}x_3)(b_{22}x_2 + b_{23}x_3)$. After dividing by x_3^2 and letting $w = x_2/x_3$ we obtain $q_2(w) \doteq$

$$c_{0,2,0}w^2 + c_{0,1,1}w + c_{0,0,2} = (b_{12}w + b_{13})(b_{22}w + b_{23}).$$

Since $\boldsymbol{c} \in \mathbb{R}^6$ is known, so is the second order polynomial $q_2(w)$. Thus we can obtain $\frac{b_{13}}{b_{12}}$ and $\frac{b_{23}}{b_{22}}$ from the roots w_1 and w_2 of $q_2(w)$. Since $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are only computable up to a scale factor, we can actually divide $\boldsymbol{c}$ by $c_{0,2,0}$ (if nonzero) and set the last two entries of $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ to be

$$b_{12} = 1, \quad b_{13} = -w_1, \quad b_{22} = 1, \quad \textit{and} \quad b_{23} = -w_2.$$

We are left with the computation of the first entry of $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. We notice that the coefficients $c_{1,1,0}$ and $c_{1,0,1}$ are linear functions of the unknowns b_{11} and b_{21}. Therefore, if $b_{22}b_{13} - b_{23}b_{12} \neq 0$, i.e. if $w_1 \neq w_2$, then we can obtain b_{11} and b_{21} from the linear system

$$\begin{bmatrix} b_{22} & b_{12} \\ b_{23} & b_{13} \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} = \begin{bmatrix} c_{1,1,0} \\ c_{1,0,1} \end{bmatrix}. \tag{9}$$

We conclude from the Example 2 that, except for the degenerate cases $c_{0,2,0} = b_{12}b_{22} = 0$ or $b_{22}b_{13} - b_{23}b_{12} = 0$, the factorization of a homogeneous polynomial of degree $n = 2$ in $K = 3$ variables can be done as follows.

1. Solve for the last two entries of $\{\boldsymbol{b}_i\}_{i=1}^n$ from the roots of a polynomial $q_n(w)$ associated with the last $n + 1$ coefficients of $p_n(\boldsymbol{x})$.

2. Solve for the first $K - 2$ entries of $\{\boldsymbol{b}_i\}_{i=1}^n$ by solving $K - 2$ linear systems in n variables.

We now show in Sections 3.3.1 and 3.3.2 how the above example can be generalized to arbitrary n and K, except for some degenerate cases. We analyze such degenerate cases in Section 3.3.3 and briefly outline how to handle them. We summarize the overall algorithm in Section 3.4.

3.3.1 Solving for the last 2 entries of each $\boldsymbol{b}_i$

Consider the last $n + 1$ coefficients of $p_n(\boldsymbol{x})$:

$$[c_{0,\ldots,0,n,0} \ , \ c_{0,\ldots,0,n-1,1} \ , \ \cdots \ , \ c_{0,\ldots,0,0,n}]^T \in \mathbb{R}^{n+1},$$

which define the following homogeneous polynomial of degree n in the two variables x_{K-1} and x_K:

$$\sum c_{0,\ldots,0,n_{K-1},n_K} x_{K-1}^{n_{K-1}} x_K^{n_K} = \prod_{i=1}^{n} (b_{iK-1}x_{K-1} + b_{iK}x_K).$$

Letting $w = x_{K-1}/x_K$, we obtain

$$\prod_{i=1}^{n} (b_{iK-1}x_{K-1} + b_{iK}x_K) = 0 \Leftrightarrow \prod_{i=1}^{n} (b_{iK-1}w + b_{iK}) = 0.$$

[4] One can interpret $\boldsymbol{c}$ as a vector representation of the symmetric part of the tensor $\boldsymbol{b}_1 \otimes \cdots \otimes \boldsymbol{b}_n$. In this case estimating $\{\boldsymbol{b}_i\}_{i=1}^n$ from $\boldsymbol{c}$ is equivalent to factoring such a symmetric tensor. The factorization algorithm we are about to present can be thought of as a SVD for symmetric tensors.

Hence the n roots of the polynomial

$$q_n(w) = c_{0,\ldots,0,n,0}w^n + c_{0,\ldots,0,n-1,1}w^{n-1} + \cdots + c_{0,\ldots,0,0,n}$$

are exactly $w_i = -b_{iK}/b_{iK-1}$, for all $i = 1,\ldots,n$. Therefore, after dividing $\boldsymbol{c}$ by $c_{0,\ldots,0,n,0}$, we obtain the last two entries of each $\boldsymbol{b}_i$ as:

$$(b_{iK-1}\ ,\ b_{iK}) = (1\ ,\ -w_i). \tag{10}$$

If $b_{iK-1} = 0$ for some i, then some of the leading coefficients of $q_n(w)$ are zero and we cannot proceed as before, because $q_n(w)$ has less than n roots. More specifically, assume that the first $\ell \leq n$ coefficients of $q_n(w)$ are zero and divide $\boldsymbol{c}$ by the $(\ell+1)$-st coefficient. In this case, we can choose $(b_{iK-1}, b_{iK}) = (0,1)$, for $i = 1,\ldots,\ell$, and obtain $\{(b_{iK-1}, b_{iK})\}_{i=n-\ell+1}^{n}$ from the $n-\ell$ roots of $q_n(w)$ using equation (10). Finally, if all the coefficients of $q_n(w)$ are zero, we set $(b_{iK-1}, b_{iK}) = (0,0)$, for all $i = 1,\ldots,n$.

3.3.2 Solving for the first $K-2$ entries of each $\boldsymbol{b}_i$

We have demonstrated how to obtain the last two entries of each $\boldsymbol{b}_i$ from the roots of a polynomial of degree n in one variable. We are now left with the computation of the first $K-2$ entries of each $\boldsymbol{b}_i$. We assume that we have computed $\{b_{ij}\}_{i=1}^{n}$, $j = J+1,\ldots K$ for some J, starting with $J = K-2$, and show how to linearly solve for $\{b_{iJ}\}_{i=1}^{n}$. As in Example 2, the key is to consider the coefficients of $p_n(\boldsymbol{x})$ which are *linear* in x_J. These coefficients are of the form $c_{0,\ldots,0,1,n_{J+1},\ldots,n_K}$ and are linear in b_{iJ}. To see this, notice that the polynomial $\sum c_{0,\ldots,0,1,n_{J+1},\ldots,n_K} x_{J+1}^{n_{J+1}} \cdots x_K^{n_K}$ is equal to the partial derivative of $p_n(\boldsymbol{x})$ with respect to x_J evaluated at $x_1 = x_2 = \cdots = x_J = 0$. Since

$$\frac{\partial}{\partial x_J}\left(\prod_{i=1}^{n}(\boldsymbol{b}_i^T\boldsymbol{x})\right) = \sum_{i=1}^{n} b_{iJ}\left(\prod_{\ell=1}^{i-1}(\boldsymbol{b}_i^T\boldsymbol{x})\prod_{\ell=i+1}^{n}(\boldsymbol{b}_i^T\boldsymbol{x})\right),$$

after evaluating at $x_1 = x_2 = \cdots = x_J = 0$ we obtain

$$\sum c_{0,\ldots,0,1,n_{J+1},\ldots,n_K} x_{J+1}^{n_{J+1}} \cdots x_K^{n_K} = \sum_{i=1}^{n} b_{iJ} g_i^J(\boldsymbol{x}), \tag{11}$$

where

$$g_i^J(\boldsymbol{x}) = \prod_{\ell=1}^{i-1}\left(\sum_{j=J+1}^{K} b_{\ell j}x_j\right)\prod_{\ell=i+1}^{n}\left(\sum_{j=J+1}^{K} b_{\ell j}x_j\right) \tag{12}$$

is a homogeneous polynomial of degree $n-1$ in the last $K-J$ variables in $\boldsymbol{x}$. Let $\mathcal{V}_i^J$ be the vector of coefficients of the polynomial $g_i^J(\boldsymbol{x})$. Notice that the vectors $\{\mathcal{V}_i^J\}_{i=1}^{n}$ are known, because they are functions of the known b_{ij}'s, for $j \geq J+1$. Therefore we can use equation (11) to solve for the unknowns $\{b_{iJ}\}_{i=1}^{n}$ from the linear system

$$\begin{bmatrix}\mathcal{V}_1^J & \mathcal{V}_2^J \cdots \mathcal{V}_n^J\end{bmatrix}\begin{bmatrix} b_{1J} \\ b_{2J} \\ \vdots \\ b_{nJ}\end{bmatrix} = \begin{bmatrix} c_{0,\ldots,0,1,n-1,0,\ldots,0} \\ c_{0,\ldots,0,1,n-2,1,\ldots,0} \\ \vdots \\ c_{0,\ldots,0,1,0,0,\ldots,n-1}\end{bmatrix}. \tag{13}$$

3.3.3 Degenerate cases

In order for the linear system in (13) to have a unique solution, the column vectors $\{\mathcal{V}_i^J\}_{i=1}^{n}$ (in the matrix on the left hand side) must be linearly independent. We showed in [10] that this is indeed the case if and only if for all $r \neq s$, $1 \leq r,s \leq n$, the vectors $(b_{rJ+1}, b_{rJ+2}, \ldots, b_{rK})$ and $(b_{sJ+1}, b_{sJ+2}, \ldots, b_{sK})$ are pairwise linearly independent. This latter condition is always satisfied, except for some degenerate cases described in Remark 3 below. In those degenerate cases, as long as the original polynomial $p_n(\boldsymbol{x})$ has n distinct factors, one can always perform an invertible linear transformation on the data points

$$\boldsymbol{x} \mapsto \boldsymbol{x}' = T\boldsymbol{x}, \quad T \in \mathbb{R}^{K\times K} \tag{14}$$

that induces a linear transformation on the vector of coefficients $\boldsymbol{c} \mapsto \boldsymbol{c}' = \tilde{T}\boldsymbol{c}$, $\tilde{T} \in \mathbb{R}^{M_n \times M_n}$, such that the new vectors $(b'_{rJ+1}, b'_{rJ+2}, \ldots, b'_{rK})$ are pairwise linearly independent. We refer the reader to [10] for further details on the solution of these degenerate cases.

Remark 3 (Degenerate cases) *There are essentially three cases in which the vectors $(b_{rJ+1}, b_{rJ+2}, \ldots, b_{rK})$ are not pairwise linearly independent:*

1. *The original polynomial $p_n(\boldsymbol{x})$ is such that the polynomial $q_n(w)$ has repeated roots, e.g.,* $p_3(\boldsymbol{x}) = (x_1+x_2+x_3)(x_1+2x_2+2x_3)(x_1+2x_2+x_3)$.
2. *The polynomial $q_n(w)$ associated with some factorable $p_n(\boldsymbol{x})$, e.g., $p_n(\boldsymbol{x}) = (x_1+x_3)x_3$, has more than one zero leading coefficients. In this case we have $(b_{i2}, b_{i3}) = (0,1)$ for more than one i.*
3. *The original polynomial $p_n(\boldsymbol{x})$ is not factorable. This happens, for example, when the vector of coefficients* **c** *is corrupted with noise. In this case the polynomial $q_n(w)$ may have complex roots, e.g., $p_n(\boldsymbol{x}) = x_1^2 + x_2^2 + x_2x_3 + x_3^2$, and one could "project" these complex roots onto their real parts. This typically introduces repeated real roots in the resulting polynomial, e.g., after "projection" the above polynomial $p_n(\boldsymbol{x})$ becomes $x_1^2 + x_2^2 + x_2x_3 + \frac{1}{4}x_3^2$.*

3.4. GPCA algorithm for $k = K-1$

Algorithm 1 (GPCA algorithm for the case $k = K-1$) *Given sample points $\{\boldsymbol{x}^j\}_{j=1}^{N}$, find the number of subspaces n and their normals $\{\boldsymbol{b}_i \in \mathbb{R}^K\}_{i=1}^{n}$ as follows:*

1. *Apply the Veronese map of degree i, for $i = 1,2,\ldots$, to the vectors $\{\boldsymbol{x}^j\}_{j=1}^{N}$ and form the matrix $L_i \in \mathbb{R}^{N\times M_i}$ as in (6). Stop when rank$(L_i) = M_i - 1$ and set the number of subspaces n to be the current i. Then solve for the vector of coefficients $\boldsymbol{c} \in \mathbb{R}^{M_n}$ from the linear system $L_n\boldsymbol{c} = 0$ and normalize so that $\|\boldsymbol{c}\| = 1$.*

2. *(a) Divide* $\boldsymbol{c}$ *by the first nonzero coefficient of* $q_n(w)$.
   (b) If the first ℓ, $0 \leq \ell \leq n$, *coefficients of* $q_n(w)$ *are equal to zero, set* $(b_{iK-1}, b_{iK}) = (0, 1)$ *for* $i = 1, \ldots, \ell$. *Compute* $\{(b_{iK-1}, b_{iK})\}_{i=n-\ell+1}^{n}$ *from the* $n - \ell$ *roots of* $q_n(w)$ *using (10).*
   (c) If all the coefficients of $q_n(w)$ *are zero, set* $(b_{iK-1}, b_{iK}) = (0, 0)$, *for* $i = 1, \ldots, n$.
   (d) If (b_{rK-1}, b_{rK}) *is parallel to* (b_{sK-1}, b_{sK}) *for some* $r \neq s$, *apply the transformation* $\boldsymbol{x} \mapsto \boldsymbol{x}'$ *in (14) and repeat 2(a), 2(b) and 2(c) for the new polynomial* $p_n'(\boldsymbol{x}')$ *to obtain* $\{(b'_{iK-1}, b'_{iK})\}_{i=1}^{n}$.
3. *Given* (b_{iK-1}, b_{iK}), $i = 1, \ldots, n$, *solve for* $\{b_{iJ}\}_{i=1}^{n}$ *from (13) for* $J = K - 2, \ldots, 1$. *If a transformation* $T \in \mathbb{R}^{K \times K}$ *was used in 2(d), then set* $\boldsymbol{b}_i = T^{-T}\boldsymbol{b}_i'$.

3.5. Optimal GPCA in the presence of noise

In the previous section, we proposed a "linear" algorithm for estimating a collection of subspaces from sample data points $\{\boldsymbol{x}^j\}_{j=1}^N$ lying on those subspaces. In essence, Algorithm 1 solves for the normal vectors $\{\boldsymbol{b}_i\}_{i=1}^n$ from the set of nonlinear equations $\prod_{i=1}^n(\boldsymbol{b}_i^T\boldsymbol{x}^j) = 0$, $j = 1, \ldots, N$. From an optimization point of view, Algorithm 1 gives a "linear" solution to the nonlinear least squares problem

$$\min_{\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n \in \mathbb{S}^{K-1}} \quad \sum_{j=1}^{N}\left(\prod_{i=1}^{n}(\boldsymbol{b}_i^T\boldsymbol{x}^j)\right)^2 \tag{15}$$

where $\mathbb{S}^{K-1}$ is the unit sphere in $\mathbb{R}^K$.

In this section, we derive an optimal algorithm for reconstructing the subspaces when the sample data points are corrupted with i.i.d. zero-mean Gaussian noise. We show that the optimal solution can be obtained by minimizing a function similar to the algebraic error in (15), but properly normalized. Since our derivation is based on segmentation independent constraints, we do not need to model the membership of each data point with a probability distribution. Therefore, we do not need to iterate between model estimation and data segmentation, as most iterative techniques do, e.g., K-means and EM. Instead, our approach eliminates the data segmentation step *algebraically* and solves the GPCA problem by directly optimizing over the normals to each subspace.

Let $\{\boldsymbol{x}^j\}_{j=1}^N$ be the given collection of *noisy* data points. We would like to find a collection of subspaces $\{S_i\}_{i=1}^n$ such that the corresponding noise free data points $\{\tilde{\boldsymbol{x}}^j\}_{j=1}^N$ lie on those subspaces. That is, we would like to solve the constrained nonlinear least squares optimization problem

$$\begin{array}{cl} \min & \sum_{j=1}^N \|\tilde{\boldsymbol{x}}^j - \boldsymbol{x}^j\|^2 \\ \text{subject to} & \prod_{i=1}^n(\boldsymbol{b}_i^T\tilde{\boldsymbol{x}}^j) = 0 \quad j = 1, \ldots, N. \end{array} \tag{16}$$

By using Lagrange multipliers λ^j for each constraint, the above optimization problem is equivalent to minimizing the Lagrangian function

$$\sum_{j=1}^{N}\left(\|\tilde{\boldsymbol{x}}^j - \boldsymbol{x}^j\|^2 + \lambda^j\prod_{i=1}^{n}(\boldsymbol{b}_i^T\tilde{\boldsymbol{x}}^j)\right). \tag{17}$$

After taking partial derivatives w.r.t. $\tilde{\boldsymbol{x}}^j$ we obtain

$$2(\tilde{\boldsymbol{x}}^j - \boldsymbol{x}^j) + \lambda^j\sum_{i=1}^{n}\boldsymbol{b}_i\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\tilde{\boldsymbol{x}}^j) = 0, \tag{18}$$

from which we can solve for $\lambda^j/2$ as

$$\frac{\sum_{i=1}^n\boldsymbol{b}_i^T(\boldsymbol{x}^j - \tilde{\boldsymbol{x}}^j)\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\tilde{\boldsymbol{x}}^j)}{\|\sum_{i=1}^n\boldsymbol{b}_i\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\tilde{\boldsymbol{x}}^j)\|^2} = \frac{\sum_{i=1}^n(\boldsymbol{b}_i^T\boldsymbol{x}^j)\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\tilde{\boldsymbol{x}}^j)}{\|\sum_{i=1}^n\boldsymbol{b}_i\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\tilde{\boldsymbol{x}}^j)\|^2}. \tag{19}$$

Similarly, after premultiplying (18) by $(\tilde{\boldsymbol{x}}^j - \boldsymbol{x}^j)^T$ we get

$$\|\tilde{\boldsymbol{x}}^j - \boldsymbol{x}^j\|^2 = \frac{\lambda^j}{2}\sum_{i=1}^{n}(\boldsymbol{b}_i^T\boldsymbol{x}^j)\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\tilde{\boldsymbol{x}}^j). \tag{20}$$

Replacing (19) and (20) on the objective function (16) gives

$$\tilde{E}_n(\{\tilde{\boldsymbol{x}}^j\}, \{\boldsymbol{b}_i\}) = \sum_{j=1}^{N}\frac{\left(\sum_{i=1}^n(\boldsymbol{b}_i^T\boldsymbol{x}^j)\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\tilde{\boldsymbol{x}}^j)\right)^2}{\left\|\sum_{i=1}^n\boldsymbol{b}_i\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\tilde{\boldsymbol{x}}^j)\right\|^2}. \tag{21}$$

We can obtain an objective function on the normal vectors only by considering first order statistics of $\boldsymbol{c}^T\nu_n(\boldsymbol{x}^j)$. Since this is equivalent to setting $\tilde{\boldsymbol{x}}^j = \boldsymbol{x}^j$ in (21), we obtain the simplified objective function

$$E_n(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = \sum_{j=1}^{N}\frac{\left(n\prod_{i=1}^n(\boldsymbol{b}_i^T\boldsymbol{x}^j)\right)^2}{\left\|\sum_{i=1}^n\boldsymbol{b}_i\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\boldsymbol{x}^j)\right\|^2}, \tag{22}$$

which is essentially the same as the algebraic error (15), but properly normalized according to the chosen noise model. By construction, the error function in (22) does not depend on the segmentation of the data, hence it can be used to directly recover the subspace normals $\{\boldsymbol{b}_i\}_{i=1}^n$ from a set of $N \geq n(K-1)$ data points $\{\boldsymbol{x}^j\}_{j=1}^N$. One can use Algorithm 1 to obtain an initial estimate for n and $\{\boldsymbol{b}_i\}_{i=1}^n$ and then use standard nonlinear optimization techniques to minimize (22). However, Algorithm 1 requires a much larger number of points $N \geq M_n - 1$, because it uses an overparameterized representation $\boldsymbol{c} \in \mathbb{R}^{M_n}$ of the normal vectors.

Remark 4 *The optimal error in (21) has a very intuitive interpretation. If point* j *belongs to group* i, *then* $\boldsymbol{b}_i^T\tilde{\boldsymbol{x}}^j = 0$. *Thus the contribution of point* j *to* $\tilde{E}_n$ *reduces to*

$$\frac{\left(\boldsymbol{b}_i^T\boldsymbol{x}^j\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\boldsymbol{x}^j)\right)^2}{\left(\prod_{\ell\neq i}(\boldsymbol{b}_\ell^T\boldsymbol{x}^j)\right)^2} = (\boldsymbol{b}_i^T\boldsymbol{x}^j)^2, \tag{23}$$

which is the optimal function to minimize for subspace i. *Therefore, the optimal error* $\tilde{E}_n$ *is just a clever algebraic way of writing a mixture of optimal functions for each subspace into a* single *objective function for* all *the subspaces.*

4. Applications of GPCA

In this section, we test GPCA on synthetic data and present various applications on 2-D and 3-D motion segmentation from 2-D imagery.

4.1. Experiments on synthetic data

We first test GPCA (Algorithm 1) and optimal GPCA (see Section 3.5) on synthetically generated data. We randomly pick $n = 2, 3, 4$ collections of $N = 600$ points on $k = 2$ dimensional subspaces of $\mathbb{R}^3$. Zero-mean Gaussian noise with standard deviation from 0% to 5% is added to the sample points. We run 1000 trials for each noise level. For each trial the error between the true (unit) normals $\{\boldsymbol{b}_i\}_{i=1}^n$ and the estimates $\{\hat{\boldsymbol{b}}_i\}_{i=1}^n$ is computed as

$$\text{error} = \frac{1}{n}\sum_{i=1}^{n} \text{acos}\left(\boldsymbol{b}_i^T \hat{\boldsymbol{b}}_i\right) \text{ (degrees).} \tag{24}$$

Figure 3 plots the mean error as a function of the noise level. In all the trials, the number of subspaces was correctly estimated from equation (8) as $n = 2, 3, 4$.[5] Notice that the estimates of the algebraic algorithm (left) are within $3.8°$, $8.5°$ and $13.3°$ of the ground truth for $n = 2$, $n = 3$ and $n = 4$, respectively, while the estimates of the optimal algorithm (right) are within $3.1°$, $6.4°$ and $9.7°$ of the ground truth for $n = 2, 3$ and 4, respectively. This is expected, because the algebraic algorithm uses an overparameterized representation $\boldsymbol{c} \in \mathbb{R}^{M_n}$ of the normal vectors $[\boldsymbol{b}_1, ..., \boldsymbol{b}_n] \in \mathbb{R}^{K\times n}$. Notice also that, as expected, the performance of both algorithms deteriorates as n increases.

4.2. Segmentation of 2D translational motions

Consider an image sequence whose 2-D motion field can be modeled as a mixture of purely translational motion models. That is, we assume that the optical flow $\boldsymbol{u} = [\mathrm{u}, \mathrm{v}, 1]^T \in \mathbb{P}^2$ in a window around every pixel can take one out of n possible values $\{\boldsymbol{u}_i\}_{i=1}^n$, where the number of models n is unknown. Under the Lambertian model, the optical flow $\boldsymbol{u}$ at pixel $\boldsymbol{x} = [x_1, x_2, 1]^T \in \mathbb{P}^2$ is related to the partials of the image intensity $\boldsymbol{y} = [I_{x_1}, I_{x_2}, I_t]^T \in \mathbb{R}^3$ at $\boldsymbol{x}$ by the well-known *brightness constancy constraint* (BCC) $\boldsymbol{y}^T\boldsymbol{u} = I_{x_1}\mathrm{u} + I_{x_2}\mathrm{v} + I_t = 0$. Thus the estimation of multiple translational motion models can be casted as a GPCA problem with $k = 2$ and $K = 3$, i.e. the segmentation of planes in $\mathbb{R}^3$. The optical flows $\{\boldsymbol{u}_i\}_{i=1}^n$ correspond to the normals to the planes, and the image partial derivatives $\{\boldsymbol{y}^j\}_{j=1}^N$ are the data points. Furthermore, we interpret the polynomial $p_n(\boldsymbol{y}) = \prod_{i=1}^n(\boldsymbol{u}_i^T\boldsymbol{y}) = \tilde{\boldsymbol{u}}^T\nu_n(\boldsymbol{y}) = 0$ as the *multibody brightness constancy constraint* (MBCC) and the vector of coefficients $\tilde{\boldsymbol{u}} \in \mathbb{R}^{M_n}$, where $M_n = (n+1)(n+2)/2$, as the *multibody optical flow*.

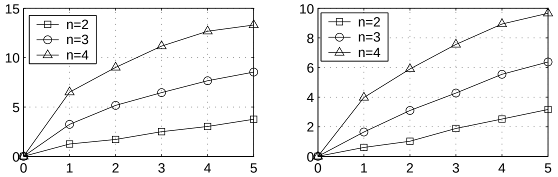


Figure 3: Error in the estimation of the subspaces as a function of noise for GPCA (left) and optimal GPCA (right).

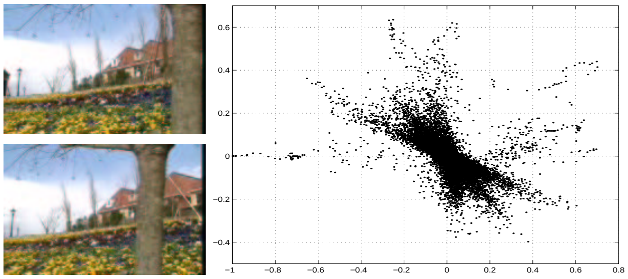
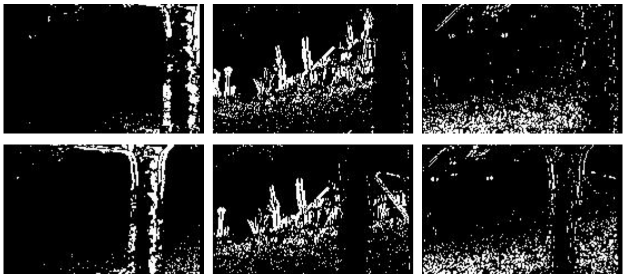


Figure 4: Frames from the flower garden sequence (left) and the image partials projected onto the I_{x_1}-I_t plane (right).



(a) Tree (b) Houses (c) Grass

Figure 5: Segmentation of two frames from the flower garden sequence using a mixture of three translational models.

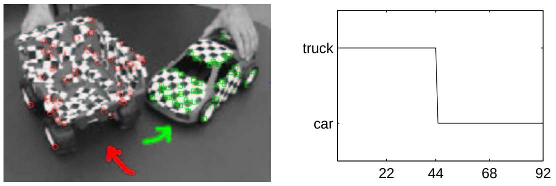
Since the MBCC incorporates multiple motion models, one can use a larger window in the computation of optical flow without having the problem of integrating image data across motion boundaries. Figures 4 and 5 show the extreme situation in which the *whole* image is used to estimate three translational models for the flower garden sequence. Figure 4 shows two frames of the sequence and the image partials for one frame projected onto the I_{x_1}-I_t plane to facilitate visualization. We observe that the image partials lie approximately on three planes through the origin, although the data is noisy and contains many outliers. We estimated three motion models by applying Algorithm 1 to the image data, followed by the nonlinear algorithm described in Section 3.5. Figure 5 shows the segmentation of the image pixels for two frames of the flower garden sequence according to the estimated motion models. Although we used a simple mixture of three translational motions to model the 2-D motion field of the sequence, a good segmentation of the tree, the houses and the grass is obtained. We did not cluster pixels with low texture ($\boldsymbol{y} \approx 0$), e.g., pixels in the sky, since they can be assigned to either of the three models.

[5] We used a threshold of $\epsilon = 3 \times 10^{-3}$ to compute the rank of L_n.

Remark 5 (Affine motion segmentation) *GPCA can also be applied to the estimation of a mixture of affine motion models $\{A_i \in \mathbb{R}^{3\times3}\}_{i=1}^n$ from image data $\{(\boldsymbol{x}^j, \boldsymbol{y}^j)\}_{j=1}^N$. In this case the optical flow is modeled with the affine model $\boldsymbol{u} = A_i\boldsymbol{x}$, thus the BCC becomes $\boldsymbol{y}^T A_i \boldsymbol{x} = 0$. Affine motion segmentation is then equivalent to estimating $\{A_i\}_{i=1}^n$ from the multibody affine constraint $\prod_{i=1}^n(\boldsymbol{y}^T A_i\boldsymbol{x}) = 0$. This can be done by factoring this product of* bilinear *forms. This problem can be reduced to a collection of GPCA problems with $k = K - 1 = 2$ as demonstrated in [11].*

4.3. Segmentation of linearly moving objects

Consider the problem of segmenting the 3-D motion of multiple objects undergoing a linear motion. That is, we assume that the scene can be modeled as a mixture of purely translational motion models, $\{\boldsymbol{e}_i \in \mathbb{R}^3\}_{i=1}^n$, where $\boldsymbol{e}_i$ represents the *epipole* (translation) of object i relative to the camera between two consecutive frames. Therefore, given the images $\boldsymbol{x}_1 \in \mathbb{P}^2$ and $\boldsymbol{x}_2 \in \mathbb{P}^2$ of a point in object i in the first and second frame, respectively, the rays $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{e}_i$ must satisfy the well-known *epipolar constraint* for linear motions $\boldsymbol{x}_2^T(\boldsymbol{e}_i \times \boldsymbol{x}_1) = 0$. Since the epipolar constraint can be conveniently rewritten as $\boldsymbol{e}_i^T(\boldsymbol{x}_2 \times \boldsymbol{x}_1) = 0$, the segmentation of linear motions is a GPCA problem with $K = 3$ and $k = 2$ where the data points are the *epipolar lines* $\boldsymbol{\ell} = \boldsymbol{x}_2 \times \boldsymbol{x}_1 \in \mathbb{R}^3$ and the normal vectors are the *epipoles* $\{\boldsymbol{e}_i\}_{i=1}^n$. Furthermore, we interpret the polynomial $p_n(\boldsymbol{\ell}) = \prod_{i=1}^n(\boldsymbol{e}_i^T\boldsymbol{\ell}) = \tilde{\boldsymbol{e}}^T\nu_n(\boldsymbol{\ell}) = 0$ as the *multibody epipolar constraint* and the vector of coefficients $\tilde{\boldsymbol{e}} \in \mathbb{R}^{M_n}$ as the *multibody epipole*. We tested GPCA on a sequence with $n = 2$ linearly moving objects. Figure 6(a) shows the first frame with $N = 92$ tracked features: 44 for the truck and 48 for the car. Figure 6(b) plots the segmentation of the features. There are no mismatches. The estimation error for the epipoles was 3.3° for the truck and 1.2° for the car.



(a) First frame (b) Feature segmentation

Figure 6: Segmentation of $n = 2$ linearly moving objects.

Remark 6 (Multibody structure from motion (MSFM)) *GPCA can also be applied to the problem of estimating a mixture of fundamental matrices $\{F_i \in \mathbb{R}^{3\times3}\}_{i=1}^n$ from image pairs $\{(\boldsymbol{x}_1^j, \boldsymbol{x}_2^j)\}_{j=1}^N$. In this case the epipolar constraint reads $\boldsymbol{x}_2^T F_i\boldsymbol{x}_1 = 0$ and the* multibody epipolar constraint *reads $\prod_{i=1}^n(\boldsymbol{x}_2^T F_i\boldsymbol{x}_1) = 0$. The MSFM problem is then equivalent to factoring this product of* bilinear *forms. Such a problem can be reduced to a collection of GPCA problems with $k = K - 1 = 2$ as shown in [13, 12].*

5. Discussion and open issues

We have proposed a novel geometric approach to the identification of mixtures of subspaces (GPCA). We derived a formula for estimating the number of subspaces and showed that GPCA is equivalent to estimating and factoring homogeneous polynomials. In the absence of noise, we presented an analytic solution to the factorization problem based on linear algebraic techniques. In the presence of noise, we presented nonlinear algorithm that minimizes the optimal error. We tested GPCA on synthetic data and presented various applications on 2-D and 3-D motion segmentation.

Open issues include an analysis of the robustness of the polynomial factorization algorithm in the presence of noise. At present the algorithm works well when the number and dimension of the subspaces is small, but the performance deteriorates as the number of subspaces increases. This is because the algorithm uses an overparameterized representation of the normal vectors that needs at least $N \geq M_n - 1$ points, as opposed to the $n(K - 1)$ points needed by the nonlinear algorithm. Whether it is possible to avoid working in a space of dimension M_n by using something similar to the *kernel trick* in NLPCA [6], remains an open question.

References

[1] T. Boult and L. Brown. Factorization-based segmentation of motions. In *IEEE Workshop on Motion Understanding*, pages 179–186, 1991.

[2] M. Collins, S. Dasgupta, and R. Schapire. A generalization of principal component analysis to the exponential family. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, 2001.

[3] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3):159–179, 1998.

[4] I. Jollife. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[5] K. Kanatani. Motion segmentation by subspace separation and model selection. In *ICCV*, volume 2, pages 586–591, 2001.

[6] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[7] M. Shizawa and K. Mase. A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *CVPR*, pages 289–295, 1991.

[8] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443-482, 1999.

[9] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 61(3):611–622, 1999.

[10] R. Vidal. *Generalized Principal Component Analysis (GPCA)*. PhD thesis, University of California at Berkeley, 2003.

[11] R. Vidal and S. Sastry. Segmentation of dynamic scenes from image intensities. In *IEEE Workshop on Motion and Video Computing*, pages 44–49, 2002.

[12] R. Vidal and S. Sastry. Optimal segmentation of dynamic scenes from the multibody epipolar constraint. In *CVPR*, 2003.

[13] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. Segmentation of dynamic scenes from the multibody fundamental matrix. In *ECCV Workshop on Visual Modeling of Dynamic Scenes*, 2002.

[14] L. Wolf and A. Shashua. Two-body segmentation from two perspective views. In *CVPR*, pages 263–270, 2001.