GPCA

Generalized Principal Component Analysis: Theory and Applications in Vision & Control

René Vidal
Center for Imaging Science
Johns Hopkins University

THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins

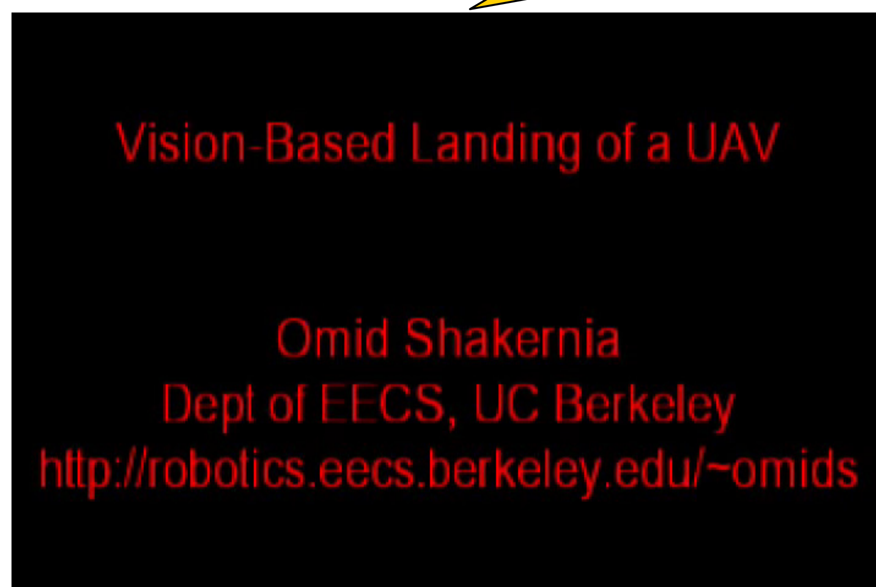
Outline

- Motivation: convergence of vision, learning and robotics
- Part I: Generalized Principal Component Analysis (1 h)
  - Principal Component Analysis (PCA) and extensions
  - Representing multiple subspaces as algebraic varieties
  - Segmenting subspaces by polynomial fitting & differentiation
  - Applications:
    - image segmentation/compression, face recognition
- Part II: Reconstruction of Dynamic Scenes (45 min)
  - 2-D and 3-D motion segmentation
  - Segmentation of dynamic textures
  - MRI-based heart motion analysis
  - DTI-based spinal cord injury detection

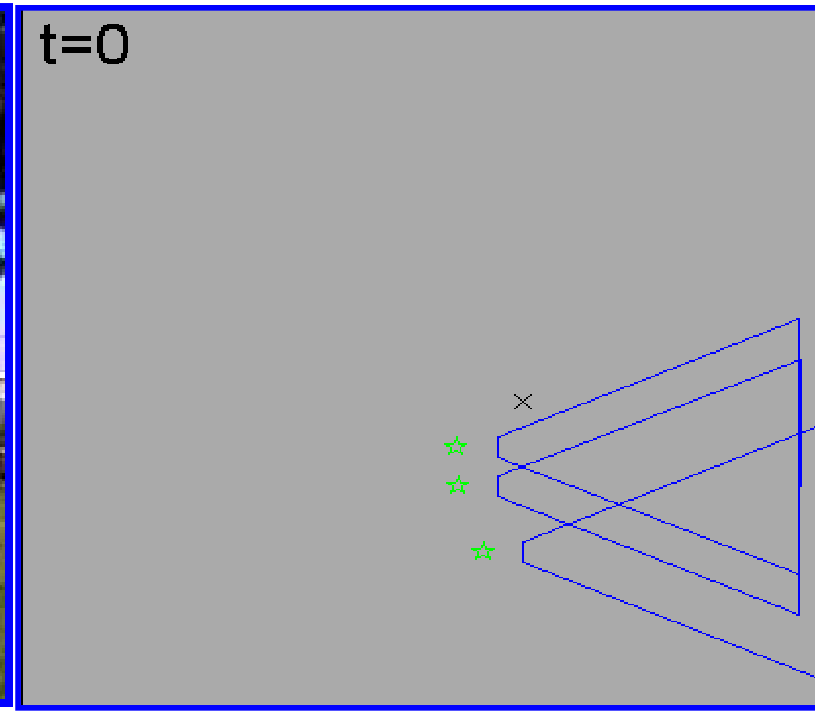
Vision based landing of a UAV

Landing on the ground

Tracking two meter waves



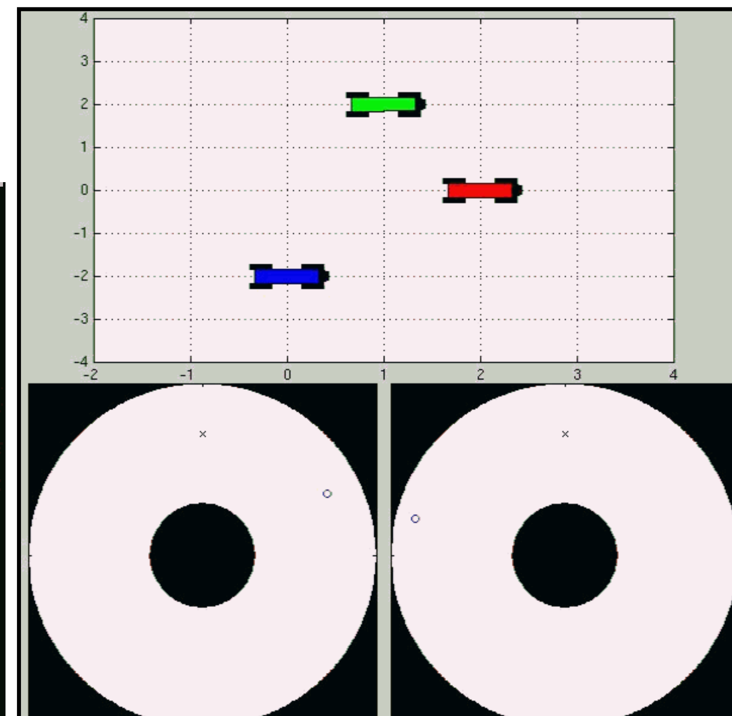
Probabilistic pursuit-evasion games

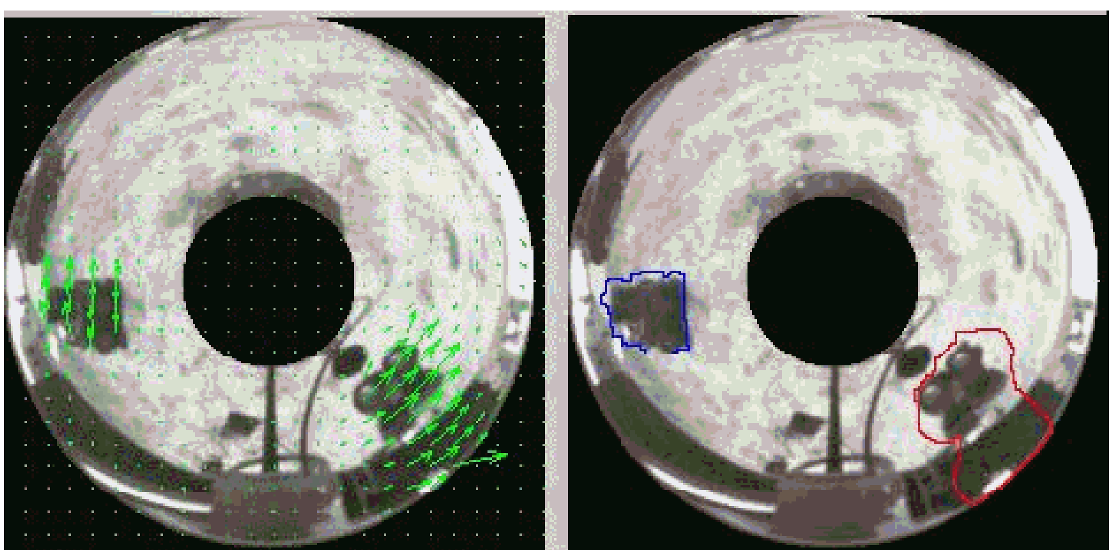


- Hierarchical control architecture
  - High-level: map building and pursuit policy
  - Mid-level: trajectory planning and obstacle avoidance
  - Low-level: regulation and control

Formation control of nonholonomic robots

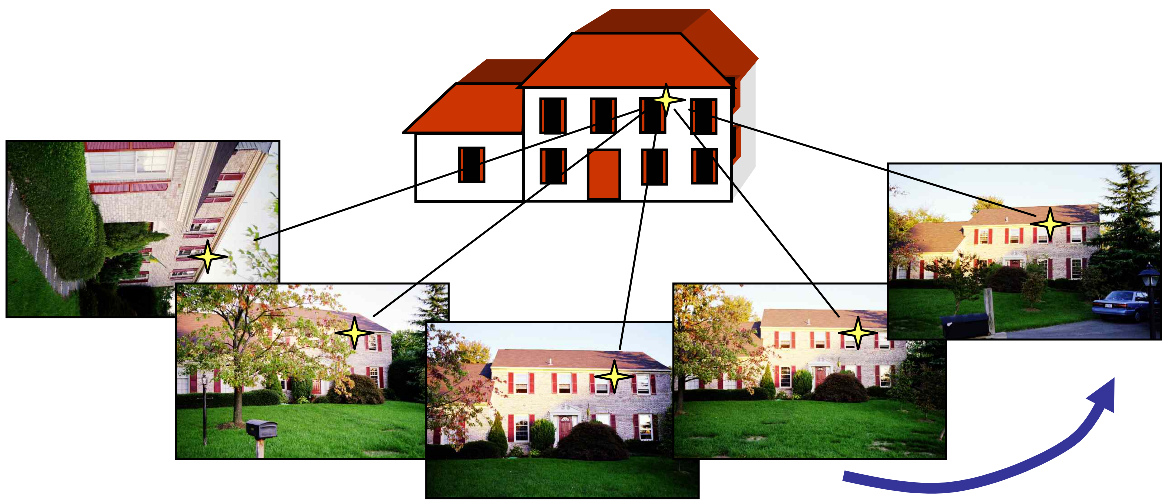
- Examples of formations
  - Flocks of birds
  - School of fish
  - Satellite clustering
  - Automatic highway systems





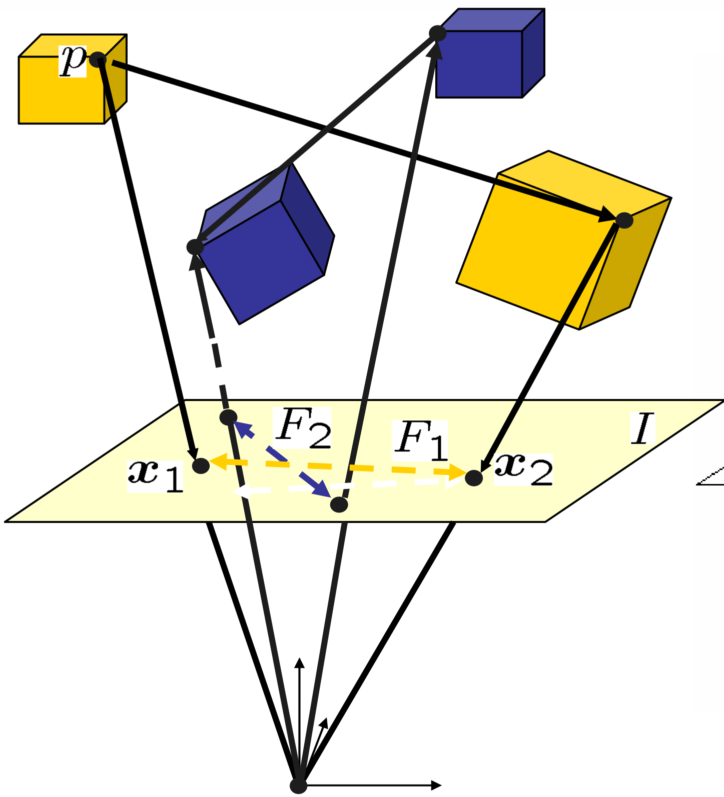
Reconstruction of static scenes

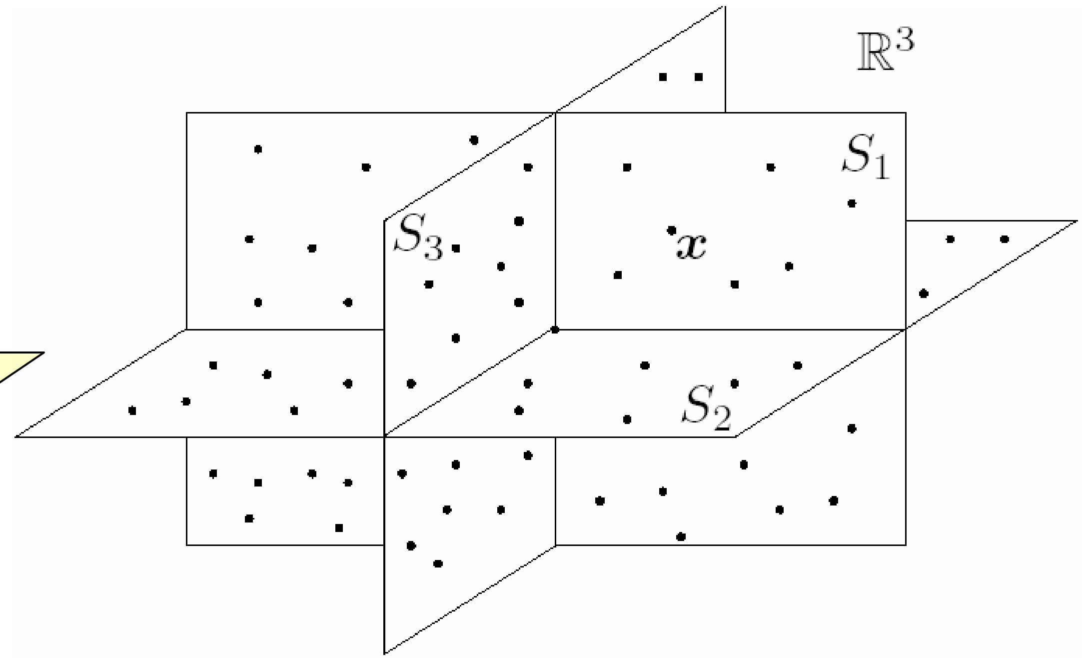
- Structure from motion and 3D reconstruction
  - Input: Corresponding points in multiple images
  - Output: camera motion, scene structure, calibration



- Theory
  - Multiview geometry: Multiple view matrix
  - Multiple view normalized epipolar constraint
  - Linear self-calibration
- Algorithms
  - Multiple view matrix factorization algorithm
  - Multiple view factorization for planar motions

Reconstruction of dynamic scenes





- Multibody Structure from Motion
  - 2D Motion Segmentation:
    - multibody brightness constancy constraint
  - 3D Motion Segmentation:
    - multibody fundamental matrix
    - multibody trifocal tensor

- Generalized PCA
  - Segmentation of mixtures of subspaces of unknown and varying dimensions
  - Segmentation of algebraic varieties: bilinear & trilinear

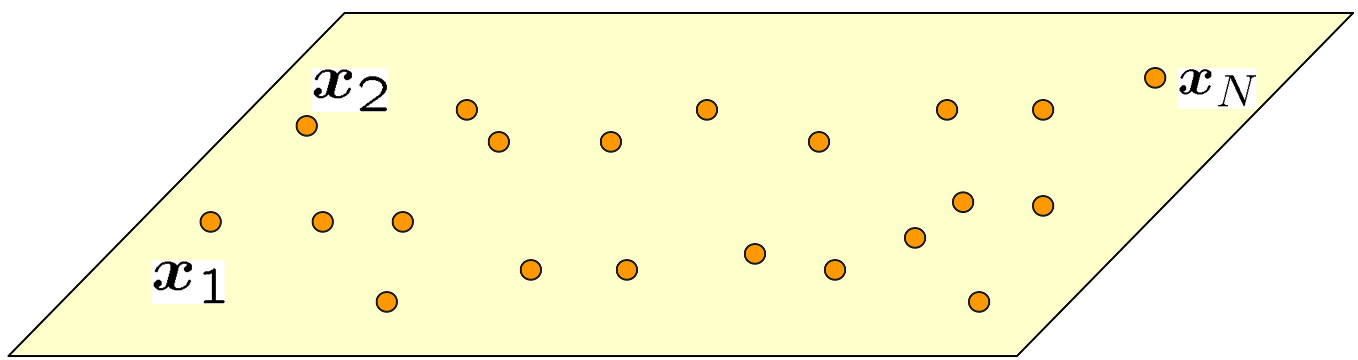
Part I: Generalized Principal Component Analysis

René Vidal
Center for Imaging Science
Johns Hopkins University

THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins

Principal Component Analysis (PCA)

- Applications: data compression, regression, image analysis (eigenfaces), pattern recognition
- Identify a linear subspace S from sample points



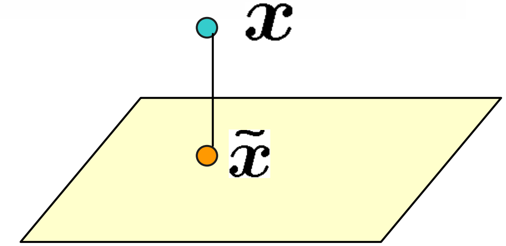
$$U \Sigma V^T = [x_1, x_2, \ldots x_N] \in \mathbb{R}^{K \times N}$$

Basis for S

$\dim(S) = \text{rank}(U)$

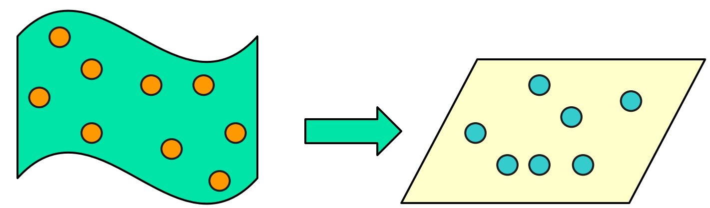
Extensions of PCA

- **Probabilistic PCA** (Tipping-Bishop '99)
  - Identify subspace from noisy data
  - Gaussian noise: standard PCA
  - Noise in exponential family (Collins et.al '01)

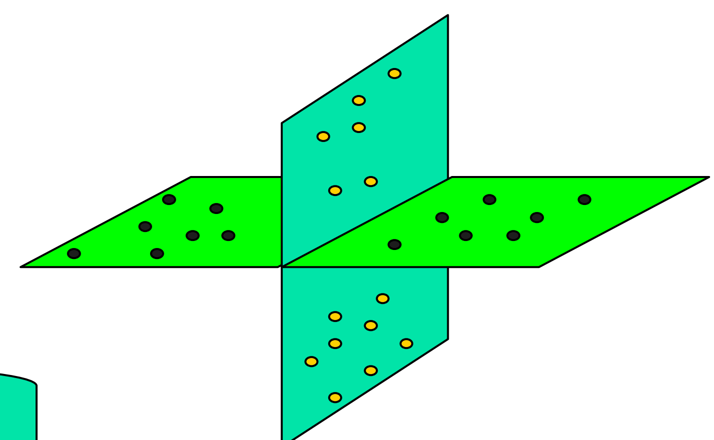


$x = \tilde{x} + \text{noise}$

- **Nonlinear PCA** (Scholkopf-Smola-Muller '98)
  - Identify a nonlinear manifold from sample points
  - Embed data in a higher dimensional space and apply standard PCA
  - What embedding should be used?



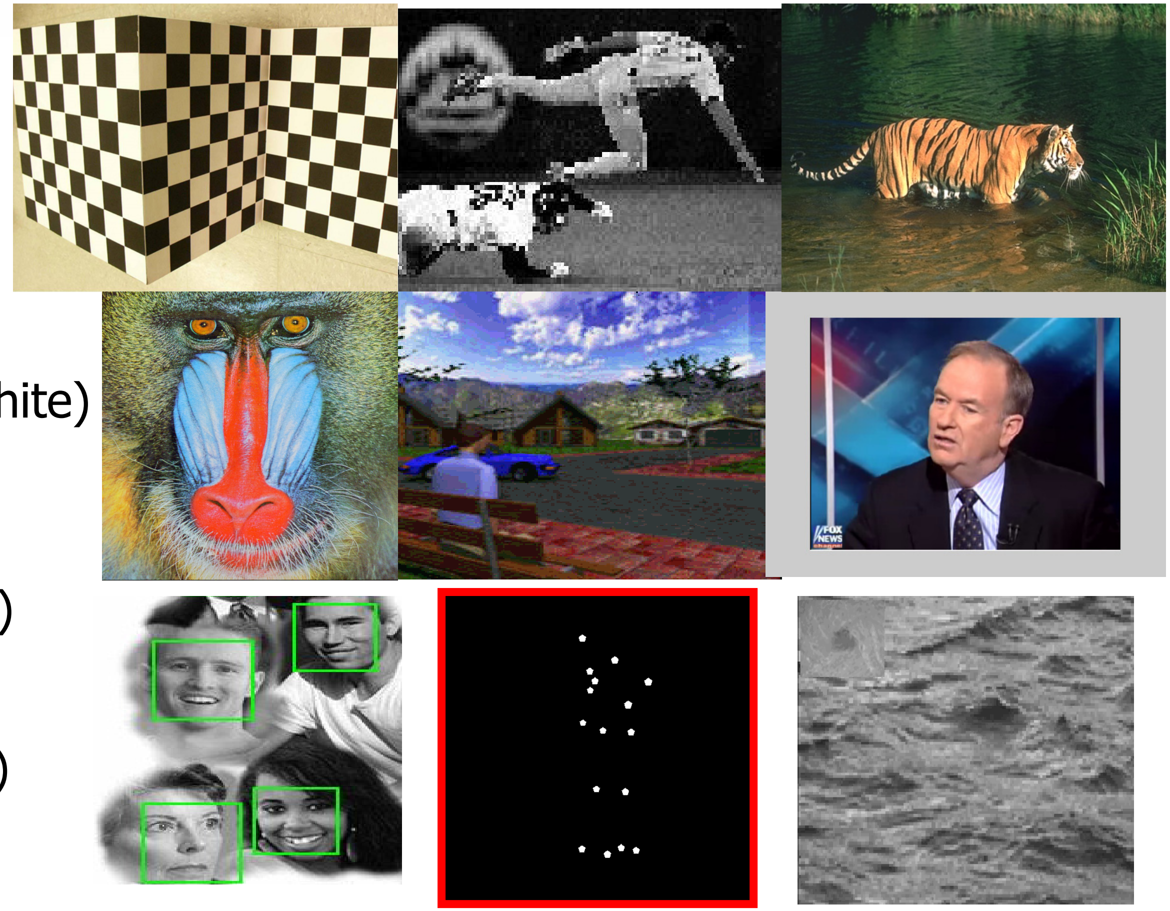
- **Mixtures of PCA** (Tipping-Bishop '99)
  - Identify a collection of subspaces from sample points

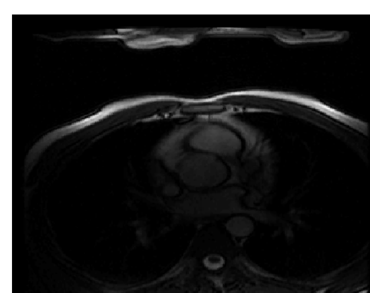


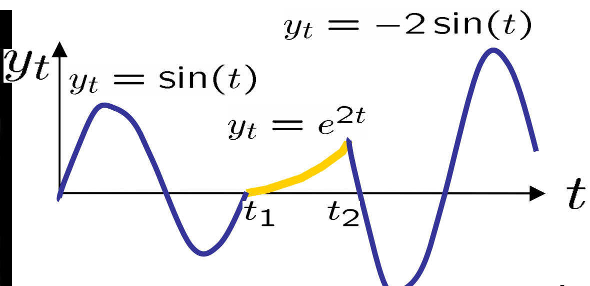
Generalized PCA (GPCA)

Applications of GPCA in vision and control

- Geometry
  - Vanishing points
- Image compression
- Segmentation
  - Intensity (black-white)
  - Texture
  - Motion (2-D, 3-D)
  - Scene (host-guest)
- Recognition
  - Faces (Eigenfaces)
    - Man - Woman
  - Human Gaits
  - Dynamic Textures
    - Water-steam
- Biomedical imaging
- Hybrid systems identification

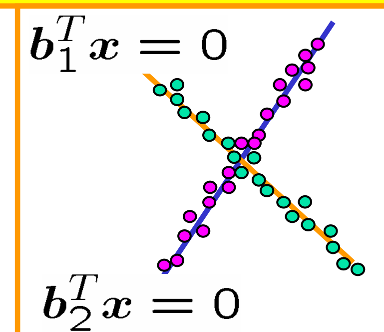






Generalized Principal Component Analysis

- Given points on multiple subspaces, identify
  - The number of subspaces and their dimensions
  - A basis for each subspace
  - The segmentation of the data points
- "Chicken-and-egg" problem
  - Given segmentation, estimate subspaces
  - Given subspaces, segment the data



- Prior work
  - Iterative algorithms: K-subspace (Ho et al. '03), RANSAC, subspace selection and growing (Leonardis et al. '02)
  - Probabilistic approaches learn the parameters of a mixture model using e.g. EM (Tipping-Bishop '99):
- Initialization?
  - Geometric approaches: 2 planes in R^3 (Shizawa-Maze '91)
  - Factorization approaches: independent, orthogonal subspaces of equal dimension (Boult-Brown '91, Costeira-Kanade '98, Kanatani '01)

Our approach to segmentation: GPCA

- Towards an analytic solution to segmentation
  - Can we estimate ALL models simultaneously using ALL data?
  - When can we do so analytically? In closed form?
  - Is there a formula for the number of models?
- We consider the most general case
  - Subspaces of unknown and possibly different dimensions
  - Subspaces may intersect arbitrarily (not only at the origin)
  - Subspaces do not need to be orthogonal
- We propose an algebraic geometric approach to data segmentation
  - Number of subspaces = degree of a polynomial
  - Subspace basis = derivatives of a polynomial
  - Subspace clustering is algebraically equivalent to
    - Polynomial fitting
    - Polynomial differentiation

Introductory example: algebraic clustering in 1D

$$x = b_1 \qquad x = b_2$$

$$x = b_1 \text{ or } x = b_2$$

$$(x - b_1)(x - b_2) = 0$$

$$x^2 - (b_1 + b_2)x + b_1 b_2 = 0$$

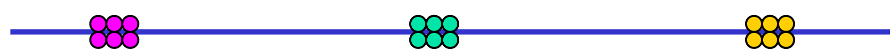
$$\underbrace{\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N & 1 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} 1 \\ -(b_1 + b_2) \\ b_1 b_2 \end{bmatrix}}_{\boldsymbol{c}} = 0$$

- Number of groups?

$$\text{rank}(P) = 1: \text{ one group only}$$

$$\text{rank}(P) = 2: \text{ two groups}$$

Introductory example: algebraic clustering in 1D



$$x = b_1 \text{ or } x = b_2 \cdots x = b_n$$

$$p_n(x) = (x-b_1)\cdots(x-b_n) = 0$$

$$p_n(x) = x^n + c_1 x^{n-1} + \cdots c_n = 0$$

$$p_n(x) = \begin{bmatrix} x^n & \cdots & x & 1 \end{bmatrix} \boldsymbol{c} = 0$$

$$P_n \boldsymbol{c} = \underbrace{\begin{bmatrix} x_1^n & \cdots & x_1 & 1 \\ x_2^n & \cdots & x_2 & 1 \\ \vdots & & \vdots & \vdots \\ x_N^n & \cdots & x_N & 1 \end{bmatrix}}_{P_n \in \mathbb{R}^{N \times (n+1}} \boldsymbol{c} = 0$$

How to compute n, c, b's?

- Number of clusters

$$n \doteq \min\{i : rank(P_i) = i\}$$

- Cluster centers

  Roots of $p_n(x)$

- Solution is unique if

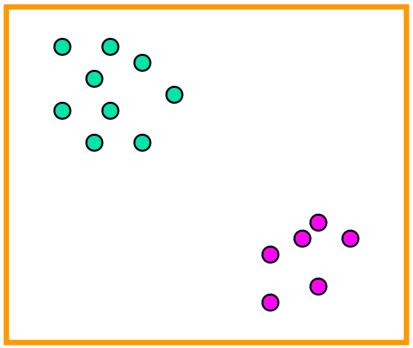
  $N_{points} \geq n_{groups}$

- Solution is closed form if

  $n_{groups} \leq 4$

Introductory example: algebraic clustering in 2D

- What about dimension 2?



$$z = x + iy \in \mathbb{C} \qquad \underbrace{\begin{bmatrix} z_1^n & \cdots & z_1 & 1 \\ z_2^n & \cdots & z_2 & 1 \\ \vdots & & \vdots & \vdots \\ z_N^n & \cdots & z_N & 1 \end{bmatrix}}_{P_n \in \mathbb{C}^{N \times (n+1)}} \boldsymbol{c} = 0$$

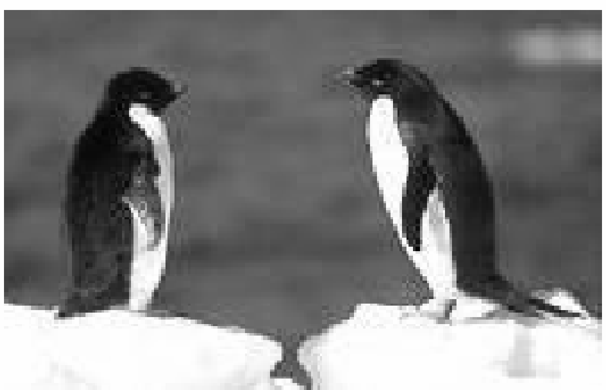
- What about higher dimensions?
  - Complex numbers in higher dimensions?
  - How to find roots of a polynomial of quaternions?

- Instead
  - Project data onto one or two dimensional space
  - Apply same algorithm to projected data

Intensity-based image segmentation

- Apply GPCA to vector of image intensities N=#pixels

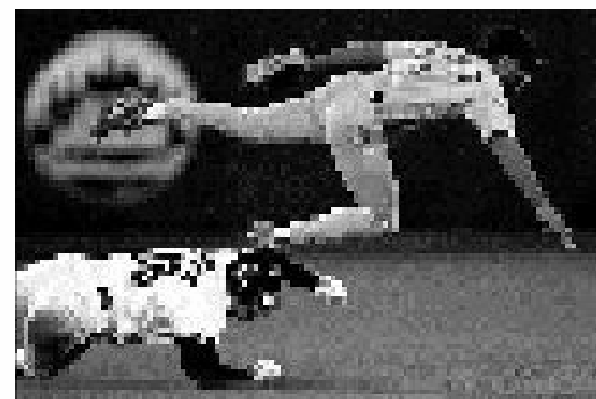
- n=3 groups
  - Black
  - Gray
  - White



(a) Penguin

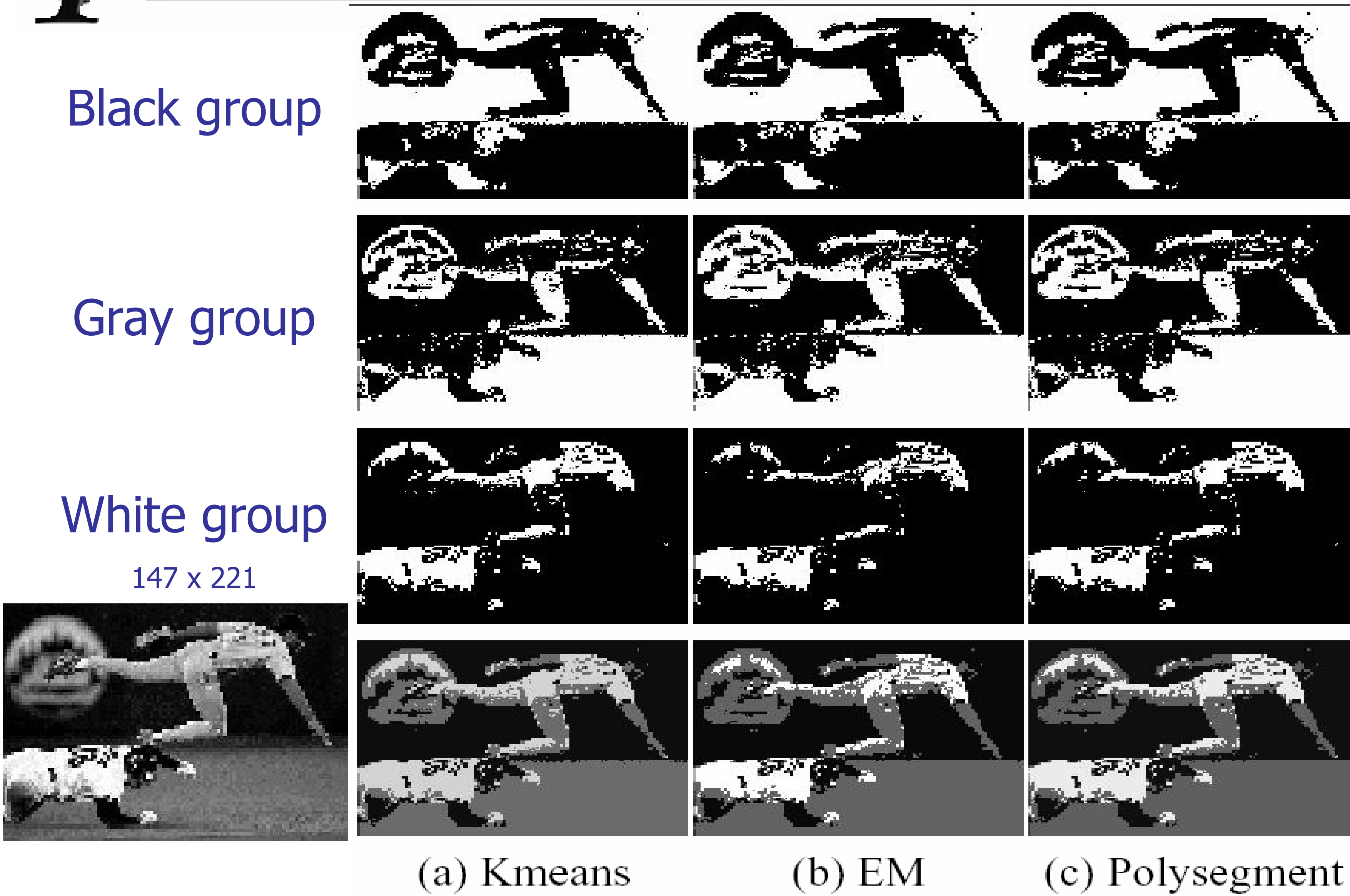
(b) Dancer

- n=3 groups
  - Black
  - Gray
  - White



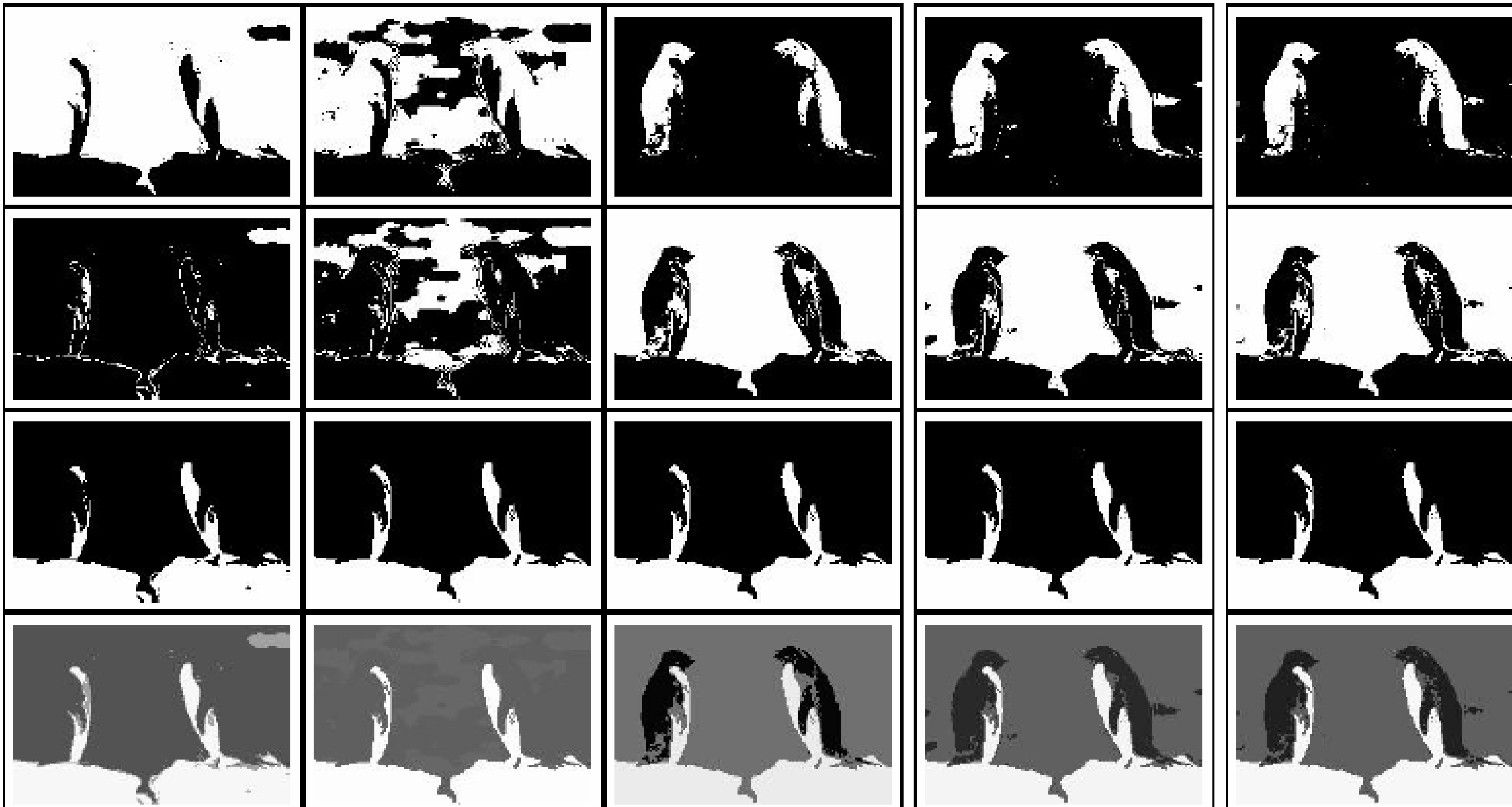
(c) Baseball

Intensity-based image segmentation



Time: 2 (sec) 10 (sec) 0.4 (sec)

Intensity-based image segmentation

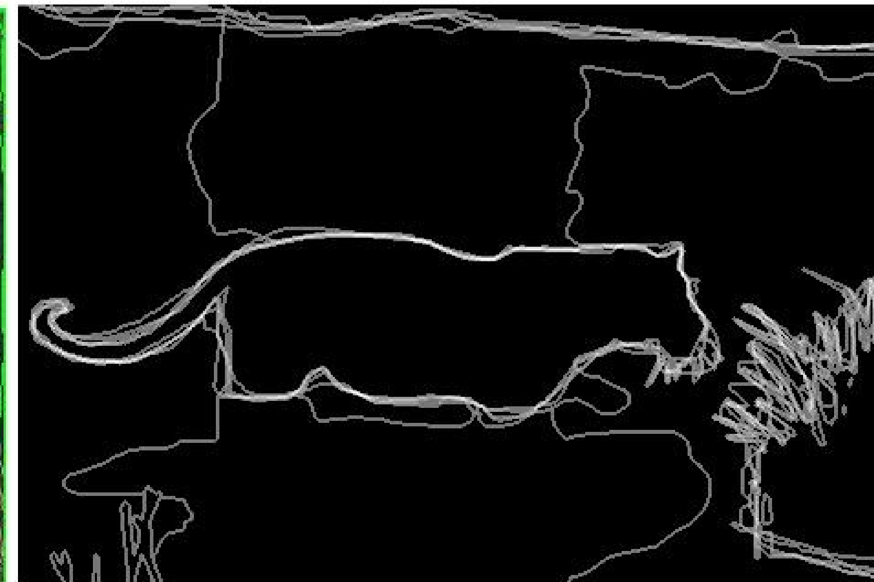


(a) Kmeans (b) EM (c) Polysegment (d) Polysegment+Kmeans (e) Polysegment+EM

1.6 (sec) 11.4 (sec) 0.3 (sec) 1.3 (sec) 7.2 (sec)

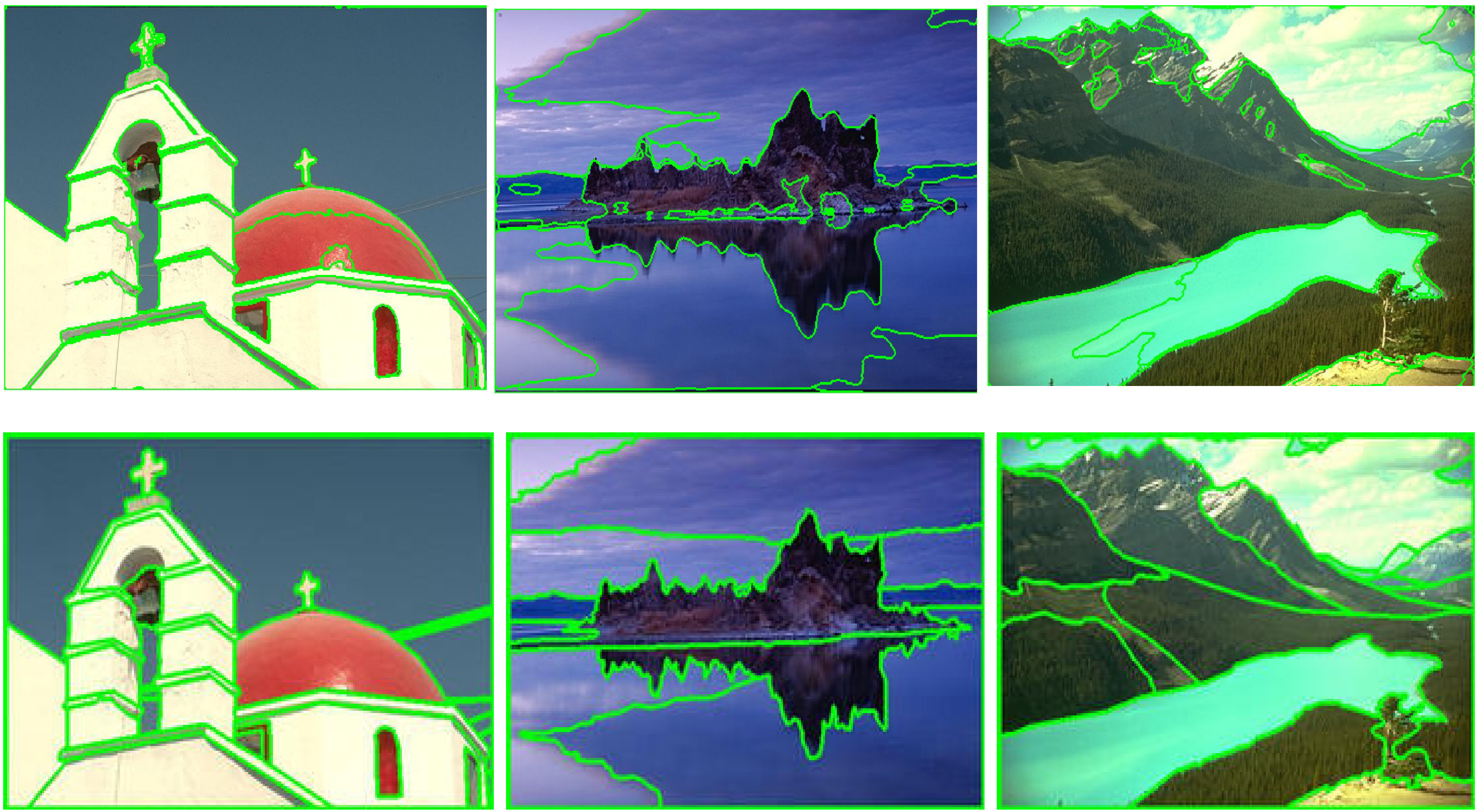
Texture-based image segmentation

GPCA: 24 (sec)



Human

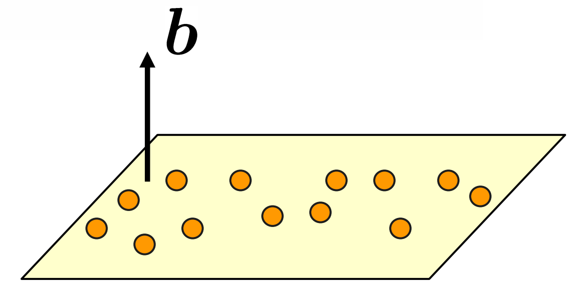
Texture-based image segmentation



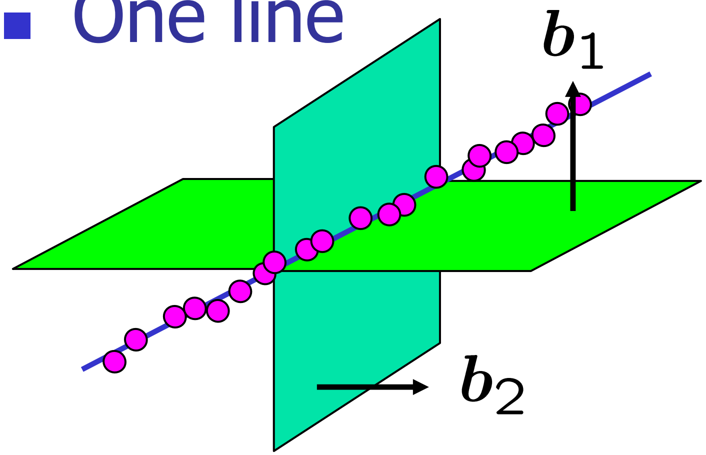
Representing one subspace

- One plane

$$\boldsymbol{b}^T\boldsymbol{x} = b_1x_1 + b_2x_2 + b_3x_3 = 0$$



- One line



$$\boldsymbol{b}_1^T\boldsymbol{x} = b_1x_1 + b_2x_2 + b_3x_3 = 0$$
$$\boldsymbol{b}_2^T\boldsymbol{x} = b_4x_1 + b_5x_2 + b_6x_3 = 0$$

- One subspace can be represented with
  - Set of linear equations $S = \{\boldsymbol{x} : B^T\boldsymbol{x} = 0\}$
  - Set of polynomials of degree 1

Representing n subspaces

- Two planes

  $(b_1^T x = 0)$ or $(b_2^T x = 0)$

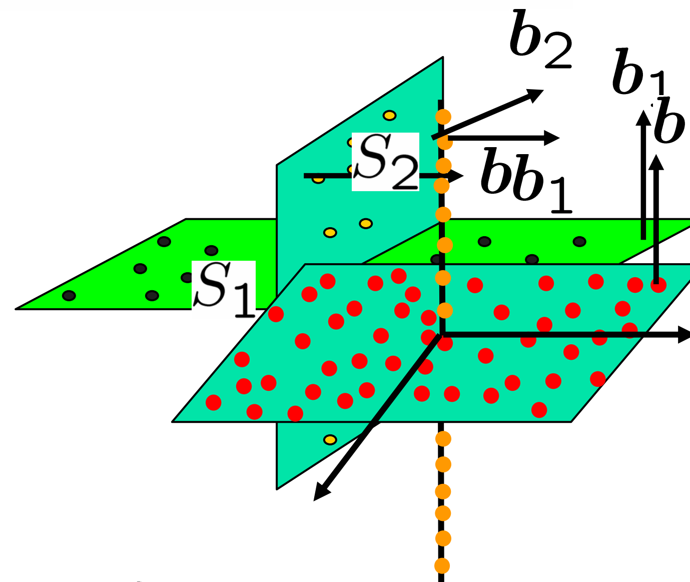
  $p_2(x) = (b_1^T x)(b_2^T x) = 0$

- One plane and one line
  - Plane: $S_1 = \{x : b^T x = 0\}$
  - Line: $S_2 = \{x : b_1^T x = b_2^T x = 0\}$

  $S_1 \cup S_2 = \{x : (b^T x = 0)$ or $(b_1^T x = b_2^T x = 0\}$

  De Morgan's rule

  $S_1 \cup S_2 = \{x : (b^T x)(b_1^T x) = 0$ and $(b^T x)(b_2^T x) = 0\}$

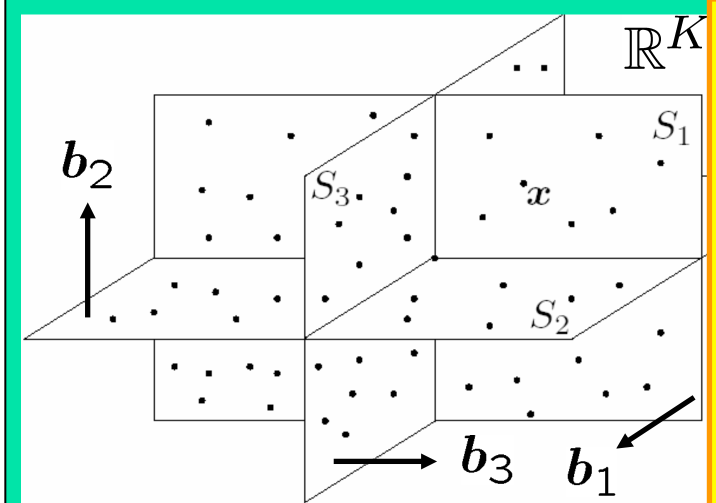


- A union of n subspaces can be represented with a set of homogeneous polynomials of degree n

Fitting polynomials to data points

- Polynomials can be written linearly in terms of the vector of coefficients by using polynomial embedding

$$(\boldsymbol{b}_1^T\boldsymbol{x})(\boldsymbol{b}_2^T\boldsymbol{x}) = c_1x_1^2 + c_2x_1x_2 + c_3x_2^2 = \boldsymbol{c}^T\nu_n(\boldsymbol{x}) = 0$$

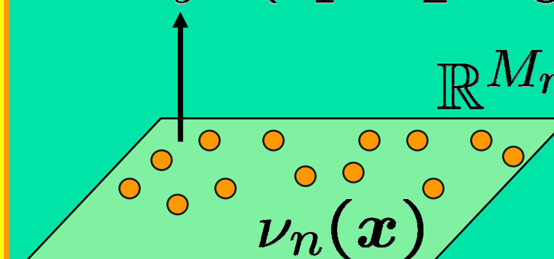


Veronese map

$$\nu_n : \mathbb{R}^K \mapsto \mathbb{R}^{M_n}$$

$$\nu_2\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_1x_2 \\ x_2^2 \end{pmatrix}$$

$$\boldsymbol{c} = Sym(\boldsymbol{b}_1 \otimes \boldsymbol{b}_2 \otimes \boldsymbol{b}_3)$$



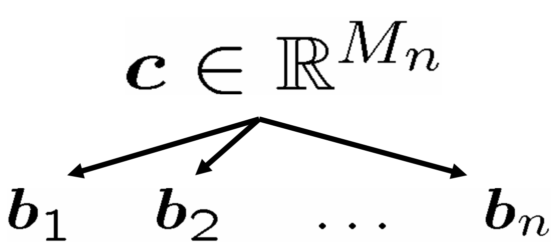
$$M_n = \binom{n+K-1}{n}$$

- Coefficients of the polynomials can be computed from nullspace of embedded data
  - Solve using least squares
  - N = #data points

$$L_n\boldsymbol{c} = \begin{bmatrix} \nu_n(\boldsymbol{x}_1)^T \\ \vdots \\ \nu_n(\boldsymbol{x}_N)^T \end{bmatrix} \boldsymbol{c} = 0$$

Finding a basis for each subspace

- Case of hyperplanes:
  - Only one polynomial $c^T \nu_n(x) = (b_1^T x) \cdots (b_n^T x)$
  - Number of subspaces $n = \min\{i : \text{rank}(L_i) = M_i - 1\}$
  - Basis are normal vectors $b_1, b_2, \cdots b_n$



> Polynomial Factorization Alg. (PFA) [CVPR 2003]
> - Find roots of polynomial of degree n in one variable
> - Solve $K-2$ linear systems in n variables
> - Solution obtained in closed form for $n \leq 4$

- Problems
  - Computing roots may be sensitive to noise
  - The estimated polynomial may not perfectly factor with noisy
  - Cannot be applied to subspaces of different dimensions
    - Polynomials are estimated up to change of basis, hence they may not factor, even with perfect data

Finding a basis for each subspace

$$\nu_n(\boldsymbol{x})^T \boldsymbol{c} = (\boldsymbol{b}_1^T \boldsymbol{x}) \cdots (\boldsymbol{b}_n^T \boldsymbol{x})$$

$\boldsymbol{c} \in \mathbb{R}^{M_n}$ → $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, $\ldots$, $\boldsymbol{b}_n$

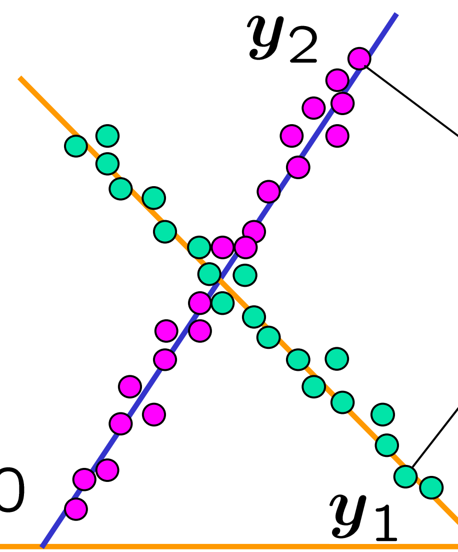
Theorem: GPCA by differentiation

$$\boldsymbol{b}_i = Dp_n(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{y}_i} \qquad \boldsymbol{y}_i \in S_i$$

$p_n(\boldsymbol{x}) = 0$

$\boldsymbol{b}_1^T \boldsymbol{x} = 0$

$\boldsymbol{b}_2^T \boldsymbol{x} = 0$



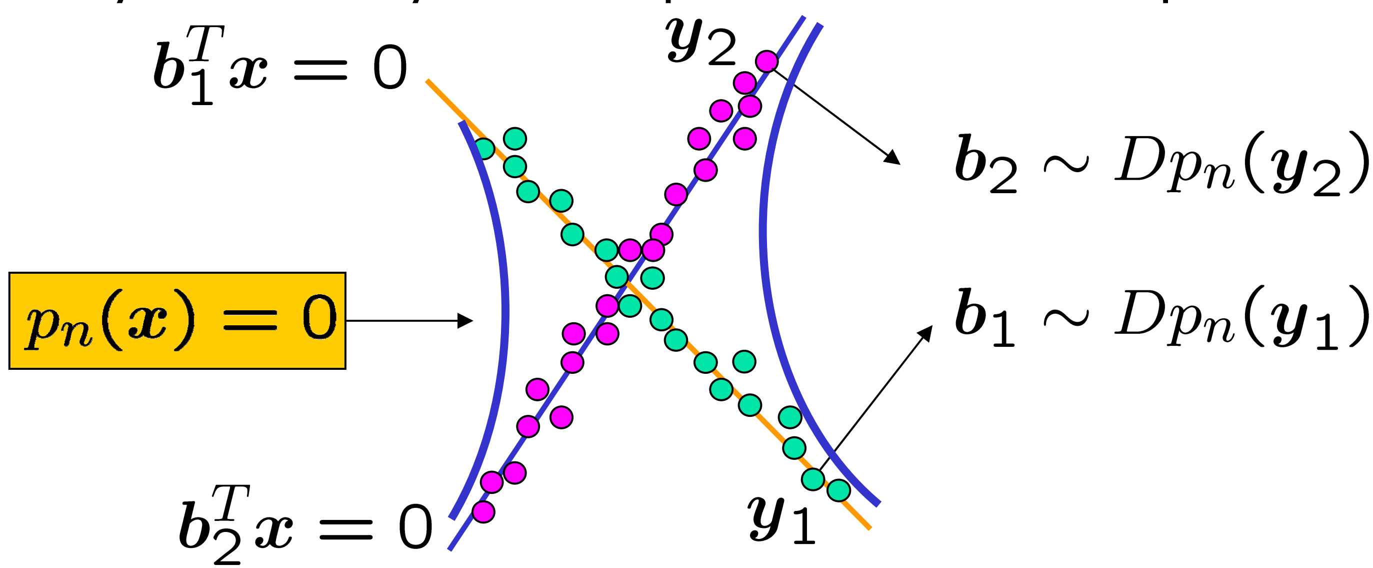
$\boldsymbol{b}_2 \sim Dp_n(\boldsymbol{y}_2)$

$\boldsymbol{b}_1 \sim Dp_n(\boldsymbol{y}_1)$

- To learn a mixture of subspaces we just need one positive example per class

Choosing one point per subspace

- With noise and outliers
  - Polynomials may not be a perfect union of subspaces



  - Normals can estimated correctly by choosing points optimally
- Distance to closest subspace without knowing segmentation?

$$\|x - \tilde{x}\| = \sqrt{\frac{|p_n(x)|}{\|Dp_n(x)\|}} + O\left(\|x - \tilde{x}\|^2\right)$$

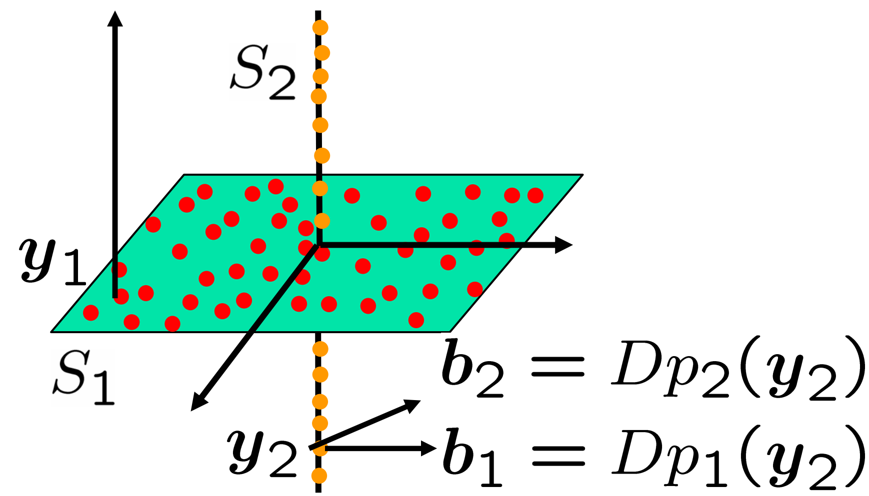
Subspaces of different dimensions

- There are multiple polynomials fitting the data

$$p_1(\boldsymbol{x}) = (\boldsymbol{b}^T\boldsymbol{x})(\boldsymbol{b}_1^T\boldsymbol{x}) = 0$$
$$p_2(\boldsymbol{x}) = (\boldsymbol{b}^T\boldsymbol{x})(\boldsymbol{b}_2^T\boldsymbol{x}) = 0$$

$$\boldsymbol{b} = Dp_1(\boldsymbol{y}_1) = Dp_2(\boldsymbol{y}_1)$$

- The derivative of each polynomial gives a different normal vector



- Can obtain a basis for the subspace by applying PCA to normal vectors

$$\{B_i = PCA(DP_n(\boldsymbol{y}_i))\}_{i=1}^n$$

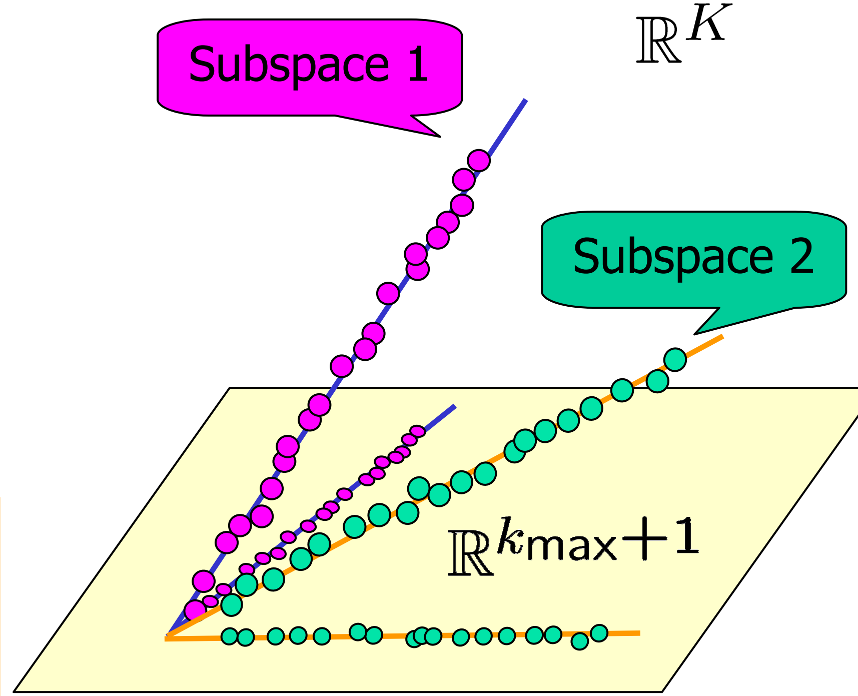
Dealing with high-dimensional data

- Minimum number of points
  - K = dimension of ambient space
  - n = number of subspaces

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

- In practice the dimension of each subspace k_i is much smaller than K

$$k_i << K$$



- Number and dimension of the subspaces is preserved by a linear projection onto a subspace of dimension

$$\max\{k_i\} + 1 << K$$

- Can remove outliers by robustly fitting the subspace

- Open problem: how to choose projection?
  - PCA?

GPCA: Algorithm summary

- Apply polynomial embedding to projected data

$$L_n = [\nu_n(\boldsymbol{x}^1), \ldots, \nu_n(\boldsymbol{x}^N)]^T \in \mathbb{R}^{N \times M_n}$$

- Obtain multiple subspace model by polynomial fitting

$$P_n(\boldsymbol{x}) \doteq [p_{n1}(\boldsymbol{x}), \ldots, p_{n,m_n}(\boldsymbol{x})] \in \mathbb{R}^{1 \times m_n}$$

  - Solve $L_n \boldsymbol{c} = 0$ to obtain $\{\boldsymbol{c}_{n\ell}\}_{\ell=1}^{m_i} \in \text{null}(L_n)$,
  - Need to know number of subspaces

- Obtain bases & dimensions by polynomial differentiation

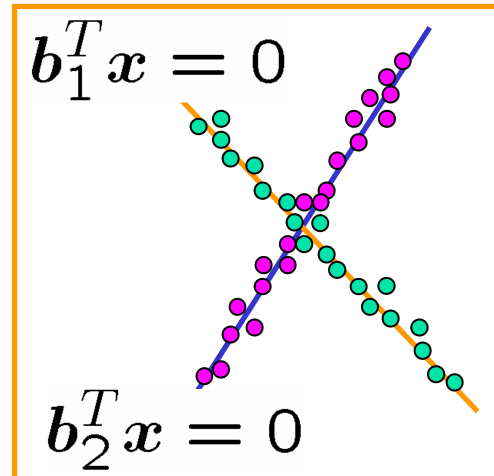
$$B_i = PCA(DP_n(\boldsymbol{y}_i)) \qquad i = 1, \ldots, n$$
$$k_i = K - \text{rank}\big(DP_n(\boldsymbol{y}_i)\big) \qquad i = 1, \ldots, n$$

- Optimally choose one point per subspace using distance

$$\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| = \sqrt{P_n(\boldsymbol{x})\big(DP_n(\boldsymbol{x})^T DP_n(\boldsymbol{x})\big)^{\dagger} P_n(\boldsymbol{x})^T} + O\big(\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2\big)$$

GPCA with spectral clustering

- Spectral clustering
  - Build a similarity matrix between pairs of points
  - Use eigenvectors to cluster data
- How to define a similarity for subspaces?
  - Want points in the same subspace to be close
  - Want points in different subspaces to be far



- Use GPCA to get basis

$$B_i = PCA(DP_n(\boldsymbol{y}_i))$$
$$B_j = PCA(DP_n(\boldsymbol{y}_j))$$

- Distance: subspace angles $\mathcal{D}_{ij} \doteq \langle B_i, B_j \rangle$

Summary

- GPCA: algorithm for clustering subspaces
  - Deals with unknown and possibly different dimensions
  - Deals with arbitrary intersections among the subspaces
- Our approach is based on
  - Projecting data onto a low-dimensional subspace
  - Fitting polynomials to projected subspaces
  - Differentiating polynomials to obtain a basis
- Applications in image processing and computer vision
  - Image segmentation: intensity and texture
  - Image compression
  - Face recognition under varying illumination

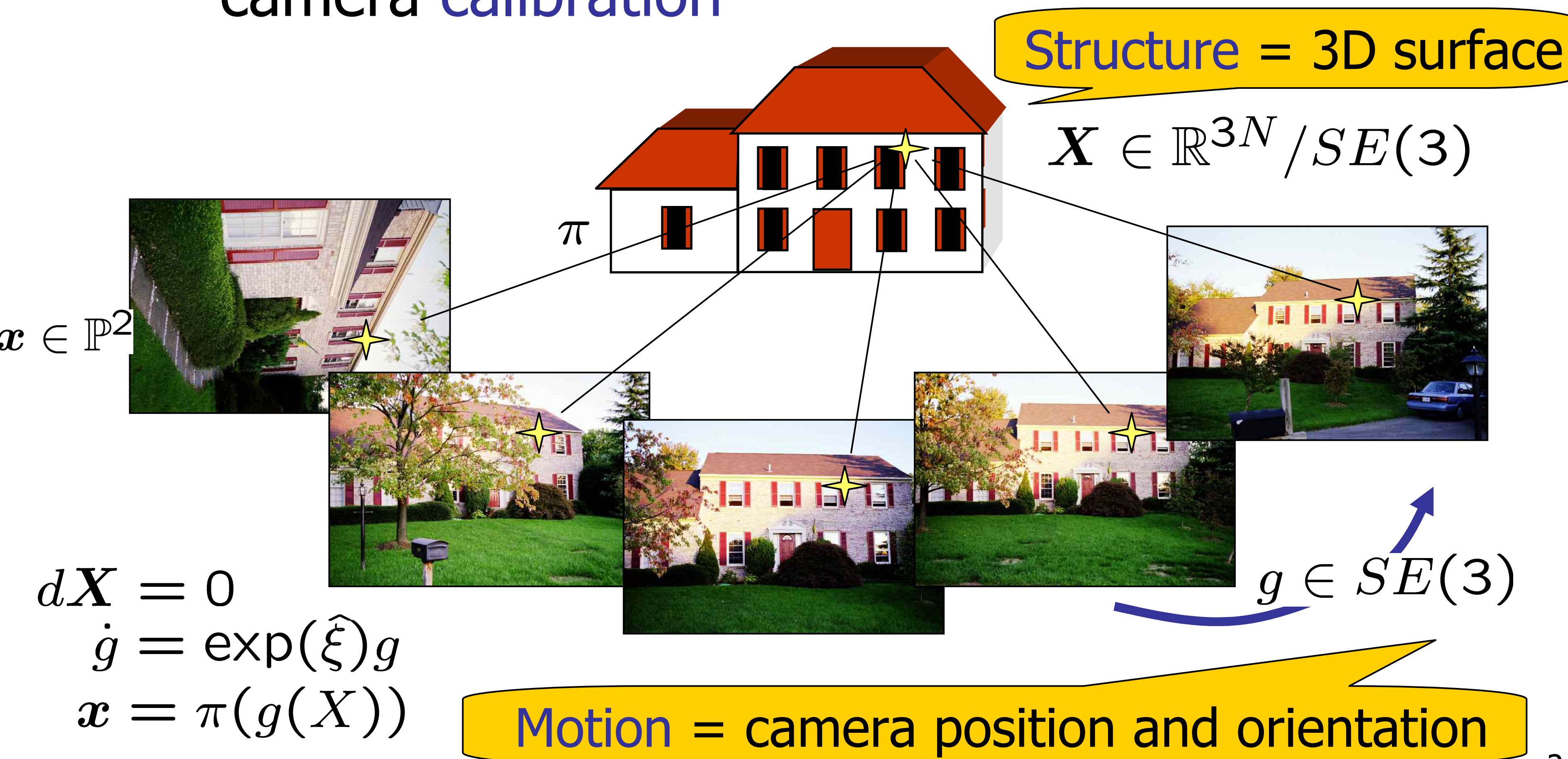
Part II
Reconstruction of Dynamic Scenes

René Vidal
Center for Imaging Science
Johns Hopkins University

THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins

Structure and motion recovery

- Input: Corresponding points in multiple images
- Output: camera motion, scene structure, camera calibration



$$dX = 0$$
$$\dot{g} = \exp(\hat{\xi})g$$
$$\boldsymbol{x} = \pi(g(X))$$

2-D and 3-D motion segmentation

- A static scene: multiple 2-D motion models

- A dynamic scene: multiple 3-D motion models

- Given an image sequence, determine
  - Number of motion models (affine, Euclidean, etc.)
  - Motion model: affine (2-D) or Euclidean (3-D)
  - Segmentation: model to which each pixel belongs

Prior work on 2-D motion segmentation

- Probabilistic approaches (Jepson-Black'93, Ayer-Sawhney '95, Darrel-Pentland'95, Weiss-Adelson'96, Weiss'97, Torr-Szeliski-Anandan '99)
  - Generative model: data membership + motion model
  - Obtain motion models using Expectation Maximization
    - E-step: Given motion models, segment image data
    - M-step: Given data segmentation, estimate motion models
- How to initialize iterative algorithms?
  - Local methods (Wang-Adelson '94)
    - One model per window + Kmeans
    - Aperture, motion across boundaries
  - Global methods (Irani-Peleg '92)
    - Dominant motion: fit one motion model to all pixels
    - Look for misaligned pixels & fit a new model to them

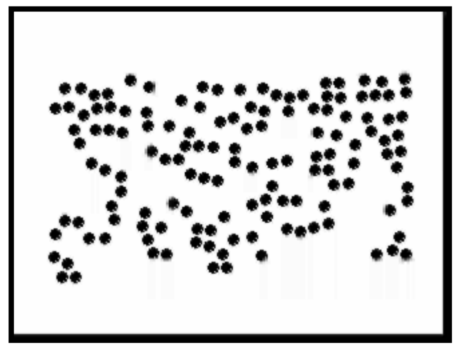
  - Spectral clustering: normalized cuts (Shi-Malik '98)
    - Similarity matrix based on motion profile

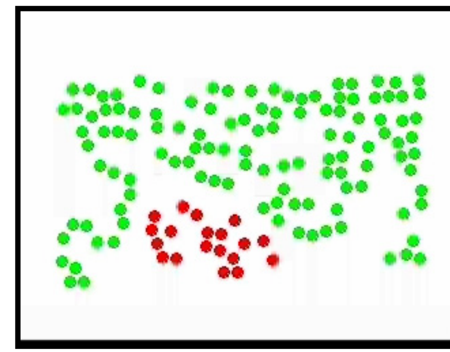
Prior work on 3-D motion segmentation

- Affine cameras, multiple views
  - Factorization methods: (Costeira-Kanade ’98, Gear ’98, Han-Kanade ’01, Kanatani ’01-02, Han-Kanade ‘00)
- Perspective cameras (statistical approaches)
  - Motion segmentation & model selection (Torr’98)
  - Multiple rigid motions using NCuts+EM (Feng-Perona ’98)
- Perspective cameras (geometric approaches)
  - Points in a line (Shashua-Levin ‘01)
  - Points in a plane moving linearly at constant speed
  - Points in moving in planes (Sturm ’02)
  - Segmentation of two rigid motions (Wolf-Shashua ‘01)

Data: point correspondences

- Given point correspondences in multiple views, estimate
  - Number of motion models
  - Motion models: affine, projective, fundamental matrices, etc.
  - Segmentation: motion model associated which each

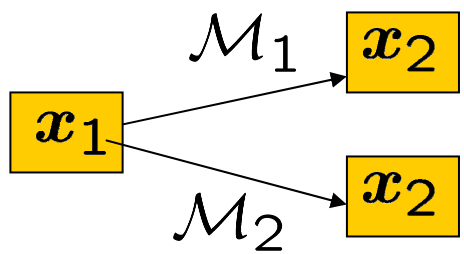




- Mathematics of the problem depends on
  - Number of frames (2, 3, multiple)
  - Projection model (affine, perspective)
  - Motion model (affine, translational, planar motion, rigid motion)
  - 3-D structure (planar or not)

A unified approach to motion segmentation

- Estimation of multiple motion models equivalent to estimation of one multibody motion model



$$f(\boldsymbol{x}_1, \boldsymbol{x}_2, \mathcal{M}_1) = 0$$

or → chicken-and-egg

$$f(\boldsymbol{x}_1, \boldsymbol{x}_2, \mathcal{M}_2) = 0$$

  - Eliminate feature clustering: multiplication

$$f(\boldsymbol{x}_1, \boldsymbol{x}_2, \mathcal{M}_1) f(\boldsymbol{x}_1, \boldsymbol{x}_2, \mathcal{M}_2) = 0$$

  - Estimate a single multibody motion model: polynomial fitting

$$f(\boldsymbol{x}_1, \boldsymbol{x}_2, \mathcal{M}_1) f(\boldsymbol{x}_1, \boldsymbol{x}_2, \mathcal{M}_2) = g(\boldsymbol{x}_1, \boldsymbol{x}_2, \mathcal{M}) = 0$$

  - Segment multibody motion model: polynomial differentiation

$$\mathcal{M} \mapsto \{\mathcal{M}_i\}_{i=1}^{n} \qquad \mathcal{M}_i = Dg|_{\boldsymbol{x}_1, \boldsymbol{x}_2}$$

A unified approach to motion segmentation

- Applies to most motion models in computer vision

Motion models	Model equations	Equivalent to clustering
2-D translational	$\boldsymbol{x}_2 = \boldsymbol{x}_1 + T_i$	Hyperplanes in $\mathbb{C}^2$
2-D similarity	$\boldsymbol{x}_2 = \lambda_i R_i \boldsymbol{x}_1 + T_i$	Hyperplanes in $\mathbb{C}^3$
2-D affine	$\boldsymbol{x}_2 = A_i \begin{bmatrix} \boldsymbol{x}_1 \\ 1 \end{bmatrix}$	Hyperplanes in $\mathbb{C}^4$
3-D translational	$0 = \boldsymbol{x}_2^T \widehat{T_i} \boldsymbol{x}_1$	Hyperplanes in $\mathbb{R}^3$
3-D fundamental matrix	$0 = \boldsymbol{x}_2^T F_i \boldsymbol{x}_1$	Bilinear forms in $\mathbb{R}^{3\times 3}$
3-D homography	$\boldsymbol{x}_2 \sim H_i \boldsymbol{x}_1$	Bilinear forms in $\mathbb{C}^{2\times 3}$
3-D trifocal tensor	$0 = \boldsymbol{x}_1 \boldsymbol{\ell}_2 \boldsymbol{\ell}_3 T_i$	Trilinear forms in $\mathbb{R}^{3\times 3\times 3}$
3-D multiframe affine	$\boldsymbol{x}_{fp} = A_{fp} X_p$	Subspaces in $\mathbb{R}^5$

- All motion models can be segmented algebraically by
  - Fitting multibody model: real or complex polynomial to all data
  - Fitting individual model: differentiate polynomial at a data point

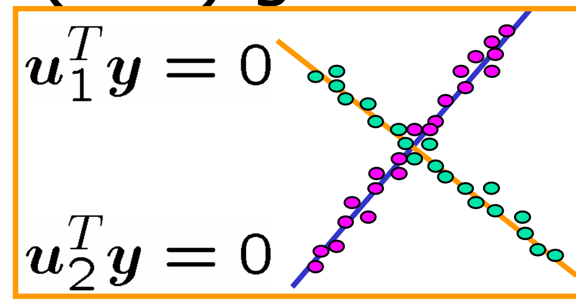
Taxonomy of problems

- 2-D Layered representation
  - Probabilistic approaches: Jepson-Black’93, Ayer-Sawhney ’95, Darrel-Pentland’95, Weiss-Adelson’96, Weiss’97, Torr-Szeliski-Anandan ’99
  - Initialization: Wang-Adelson ’94, Irani-Peleg ’92, Shi-Malik ‘98
- Multiple rigid motions in two perspective views
  - Two-body fundamental matrix: Wolf-Shashua CVPR’01
  - Multibody fundamental matrix: Vidal et al. ECCV’02, IJCV’04, CVPR’03
- Multiple 2-D and 3-D motion models in two perspective views
  - Two transparent motions: Izawa’92
  - Complex polynomial differentiation: Vidal-Ma ECCV’04
- Multiple rigid motions in three perspective views
  - Multibody trifocal tensor: Hartley-Vidal: CVPR’04
- Multiple rigid motions in multiple affine views
  - Factorization-based: Costeira-Kanade’98, Gear’98 Han-Kanade’01, Kanatani’01-02-04
  - PowerFactorization + GPCA: Vidal-Hartley: CVPR’04
- Multiple rigid motions in multiple perspective views

Segmentation of 2-D translational motions

- Scene having multiple optical flows $\{\boldsymbol{u}_i \in \mathbb{P}^2\}_{i=1}^n$
- Brightness constancy constraint (BCC) gives GPCA problem with K=3

$$\boldsymbol{y}^T\boldsymbol{u} = I_x u + I_y v + I_t = 0$$



- Multibody brightness constancy constraint

$$p_n(\boldsymbol{y}) = (u_1^T\boldsymbol{y})\cdots(u_n^T\boldsymbol{y}) = 0$$

Segmentation of 2-D translational motions

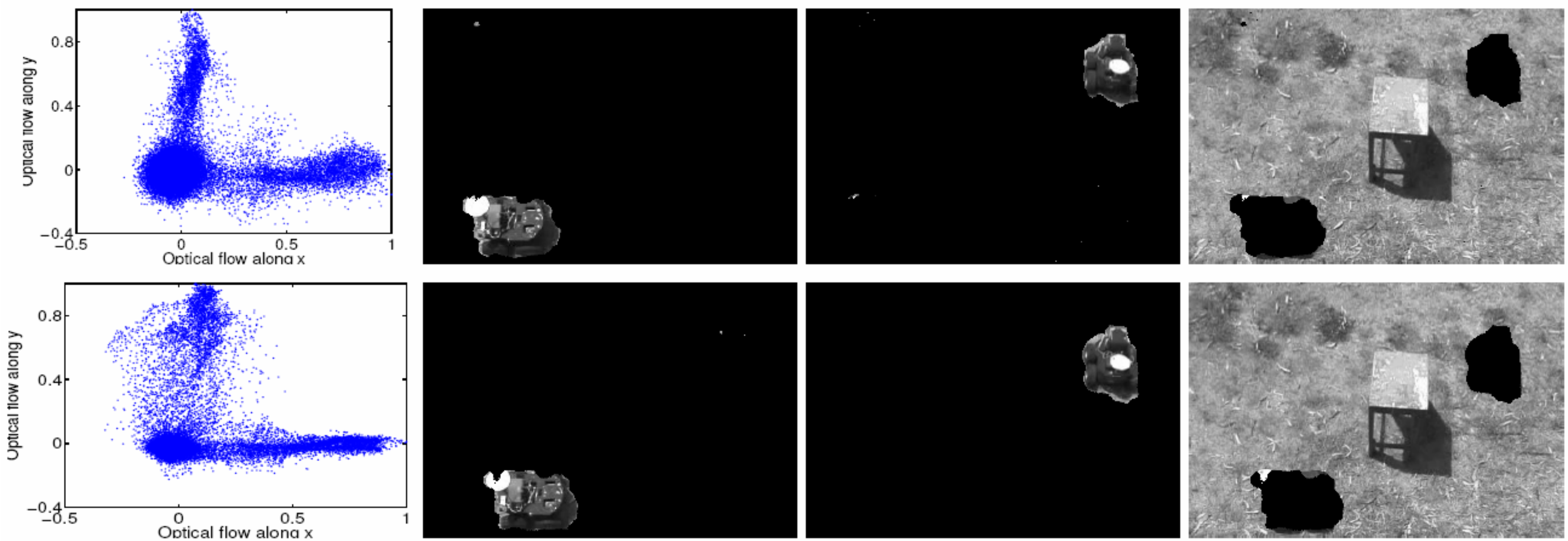


Fig. 1. Segmenting the optical flow of the two-robot sequence by clustering lines in $\mathbb{C}^2$.

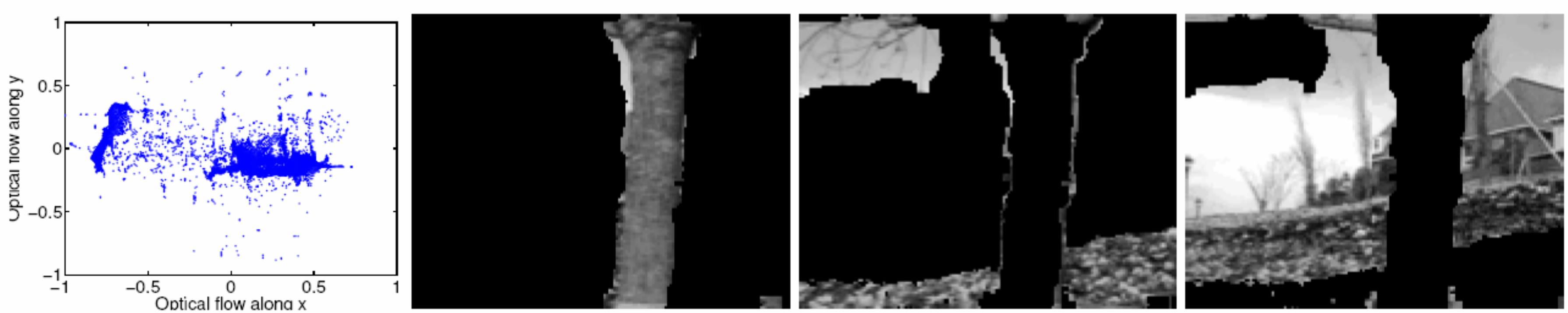


Fig. 2. Segmenting the optical flow of the flower-garden sequence by clustering lines in $\mathbb{C}^2$.

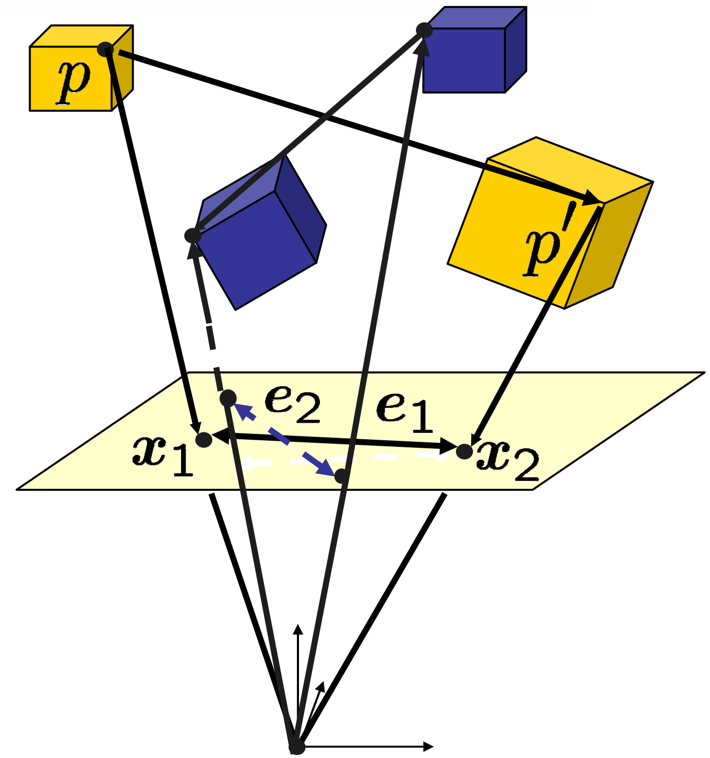
Segmentation of 3-D translational motions

- Multiple epipoles (translation)

$$\{\boldsymbol{e}_i \in \mathbb{R}^3\}_{i=1}^n$$

- Epipolar constraint: plane in $\mathbb{R}^3$
  - Plane normal = epipoles
  - Data = epipolar lines

$$\boldsymbol{e}_i^T \underbrace{(\boldsymbol{x}_1 \times \boldsymbol{x}_2)}_{\boldsymbol{\ell}=\text{epipolar line}} = 0$$



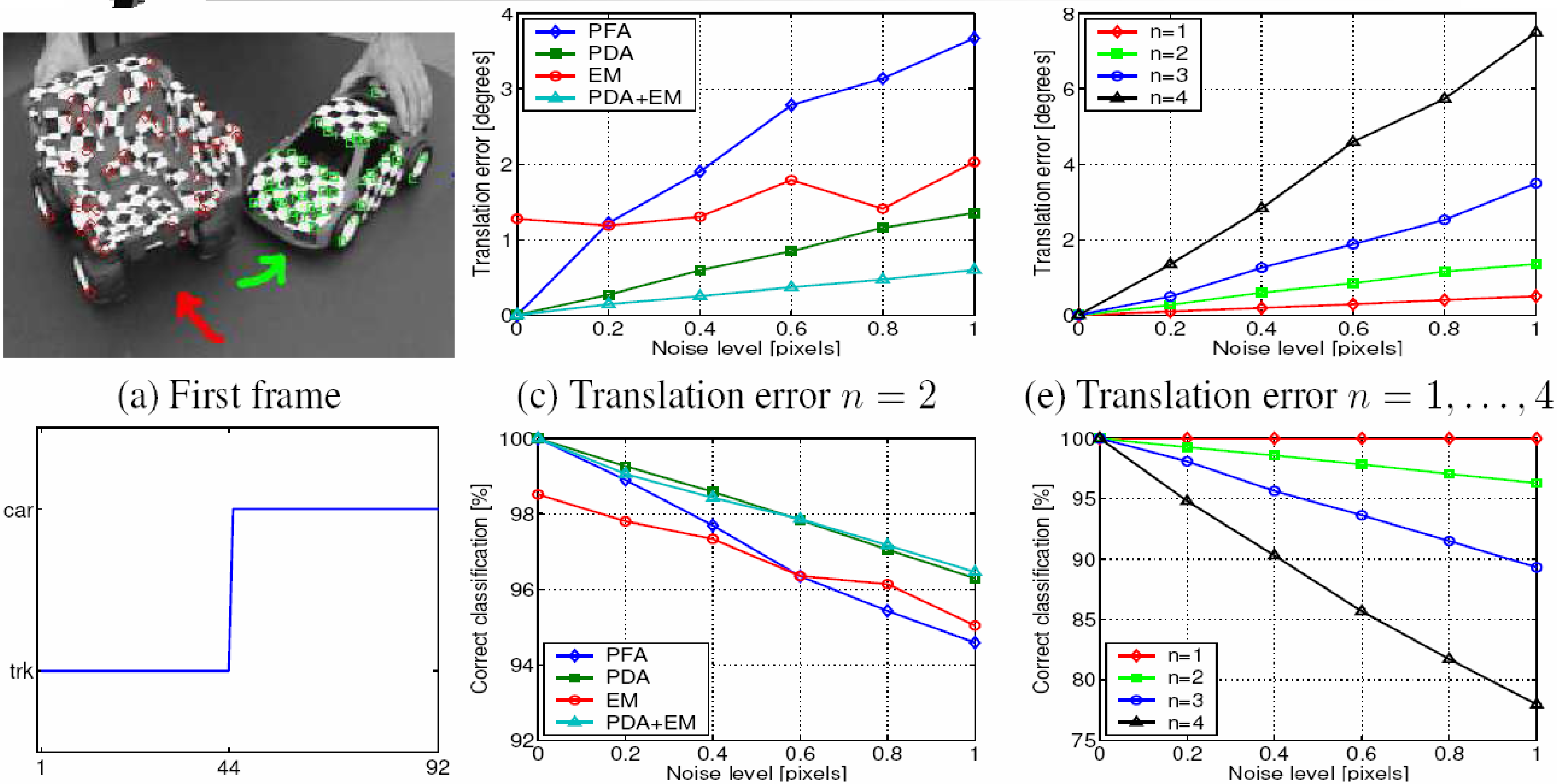
- Multibody epipolar constraint

$$p_n(\boldsymbol{\ell}) = \prod_{i=1}^n (\boldsymbol{e}_i^T \boldsymbol{\ell}) = 0$$

- Epipoles are derivatives of $p_n(\boldsymbol{\ell})$ at epipolar lines

$$\boldsymbol{e}_i = \frac{\partial\big(p_n(\boldsymbol{\ell})\big)}{\partial \boldsymbol{\ell}}\bigg|_{\boldsymbol{\ell}=\boldsymbol{\ell}_i}$$

Segmentation of 3-D translational motions

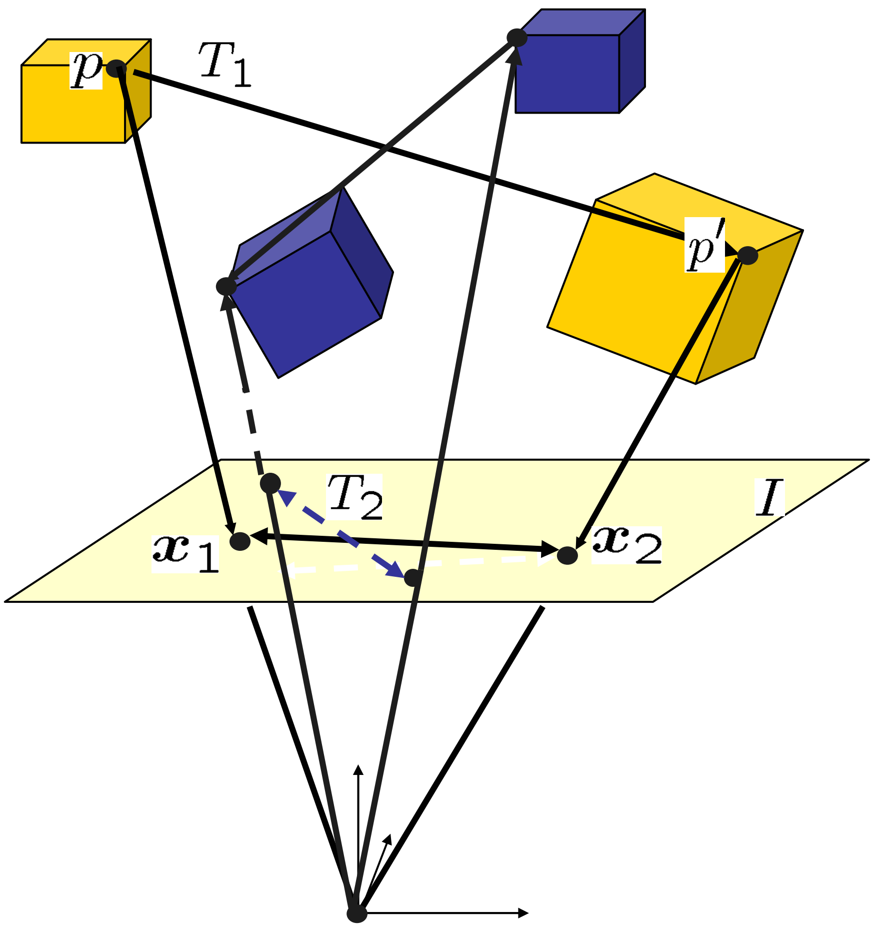


(a) First frame (c) Translation error $n = 2$ (e) Translation error $n = 1, \ldots, 4$

(b) Feature segmentation (d) % of correct classif. $n = 2$ (f) % of correct classif. $n = 1, \ldots, 4$

Fig. 3. Segmenting 3-D translational motions by clustering planes in $\mathbb{R}^3$. Left: segmenting a real sequence with 2 moving objects. Center: comparing our algorithm with PFA and EM as a function of noise in the image features. Right: performance of PFA as a function of the number of motions.

Segmentation of 3-D fundamental matrices



- Rotation: $R_1 \in SO(3)$
- Translation: $\widehat{T_1} \in so(3)$
- Epipolar constraint

$$x_2^T \underbrace{\widehat{T}_1 R_1}_{F_1 \in \mathbb{R}^{3\times 3}} x_1 = 0$$

- Multiple motions

$\{(R_i, T_i)\}_{i=1}^n \quad \{F_i \doteq \widehat{T_i} R_i\}_{i=1}^n$

$$\prod_{i=1}^n (x_2^T F_i x_1) = 0$$

$$\nu_n(x_2)^T F \nu_n(x_1) = 0$$

Multibody epipolar constraint

Segmentation of 3-D fundamental matrices

- Multibody epipolar constraint

$$E_n(\boldsymbol{x}_1, \boldsymbol{x}_2) = \prod_{i=1}^{n} (\boldsymbol{x}_2^T F_i \boldsymbol{x}_1) = 0$$

- Multibody fundamental matrix

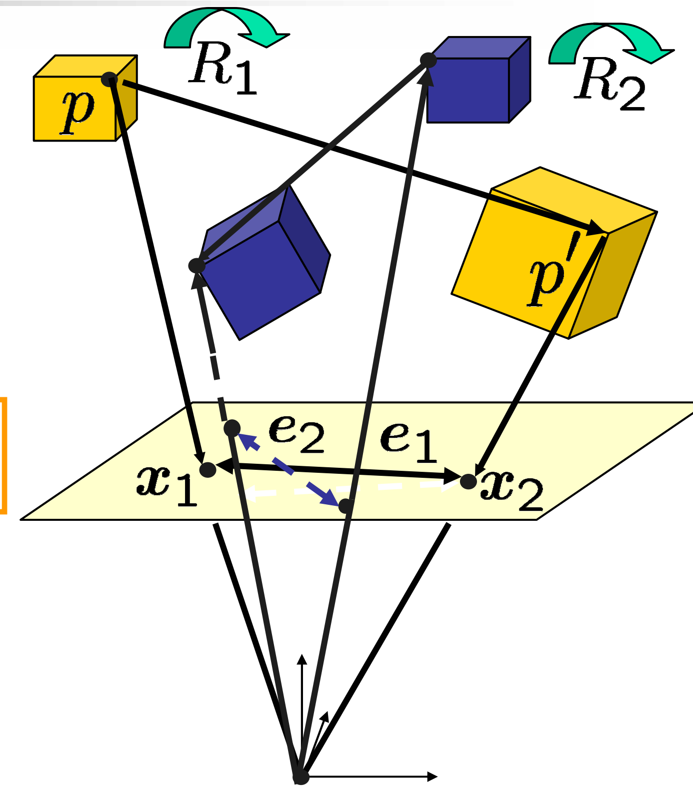
$$E_n(\boldsymbol{x}_1, \boldsymbol{x}_2) = \nu_n(\boldsymbol{x}_2)^T \mathcal{F} \nu_n(\boldsymbol{x}_1) = 0$$

- Epipolar lines: derivatives of $E_n(\boldsymbol{x}_1, \boldsymbol{x}_2)$ at a correspondence

$$\boldsymbol{\ell} = \frac{\partial\big(E_n(\boldsymbol{x}_1, \boldsymbol{x}_2)\big)}{\partial \boldsymbol{x}_2}$$

- Epipoles are derivatives of $p_n(\boldsymbol{\ell})$ at epipolar lines

$$\boldsymbol{e}_i = \frac{\partial\big(p_n(\boldsymbol{\ell})\big)}{\partial \boldsymbol{\ell}}\bigg|_{\boldsymbol{\ell}=\boldsymbol{\ell}_i}$$

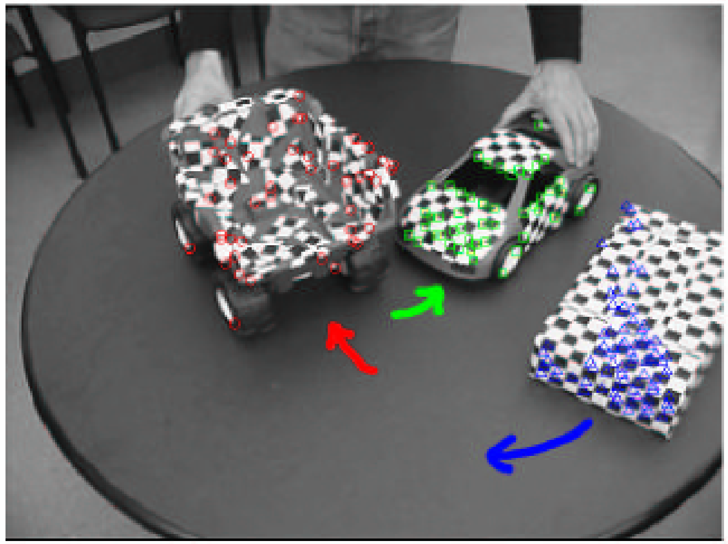


$$\{F_i \doteq \widehat{T_i} R_i\}_{i=1}^{n}$$

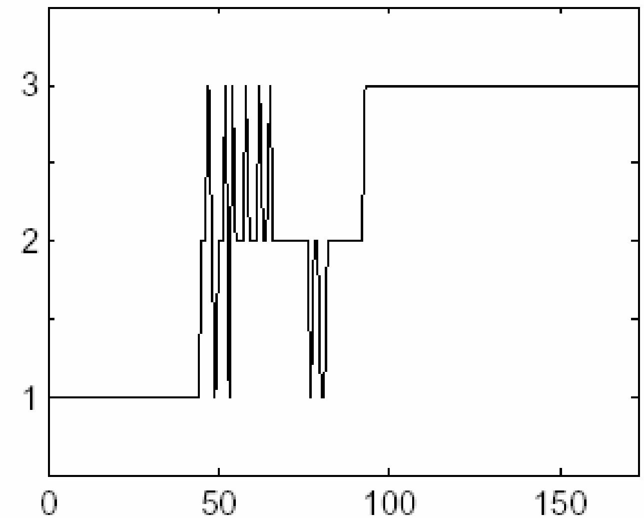
- Fundamental matrices

$$F_i = \frac{\partial^2\big(E_n(\boldsymbol{x}_1, \boldsymbol{x}_2)\big)}{\partial \boldsymbol{x}_1 \partial \boldsymbol{x}_2}\bigg|_{\substack{\boldsymbol{x}_1 = \boldsymbol{e}_i \\ \boldsymbol{x}_2 = \boldsymbol{e}_i}}$$

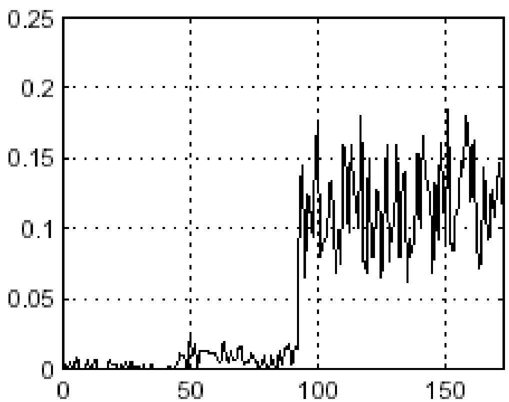
Segmentation of 3-D fundamental matrices


(a) First frame

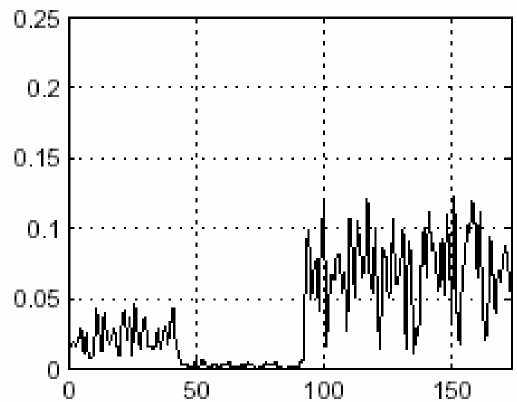
N = 44 + 48 + 81 = 173



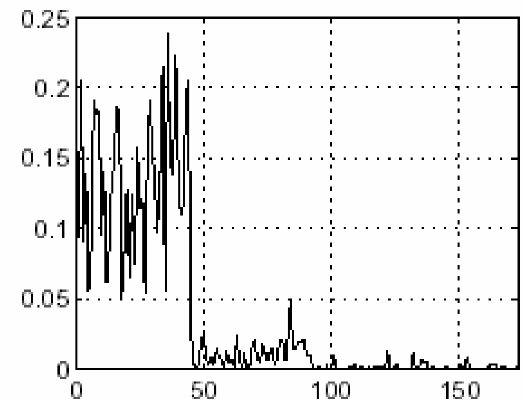
(b) Feature segmentation



(c) $E_1(\hat{F}_1)$

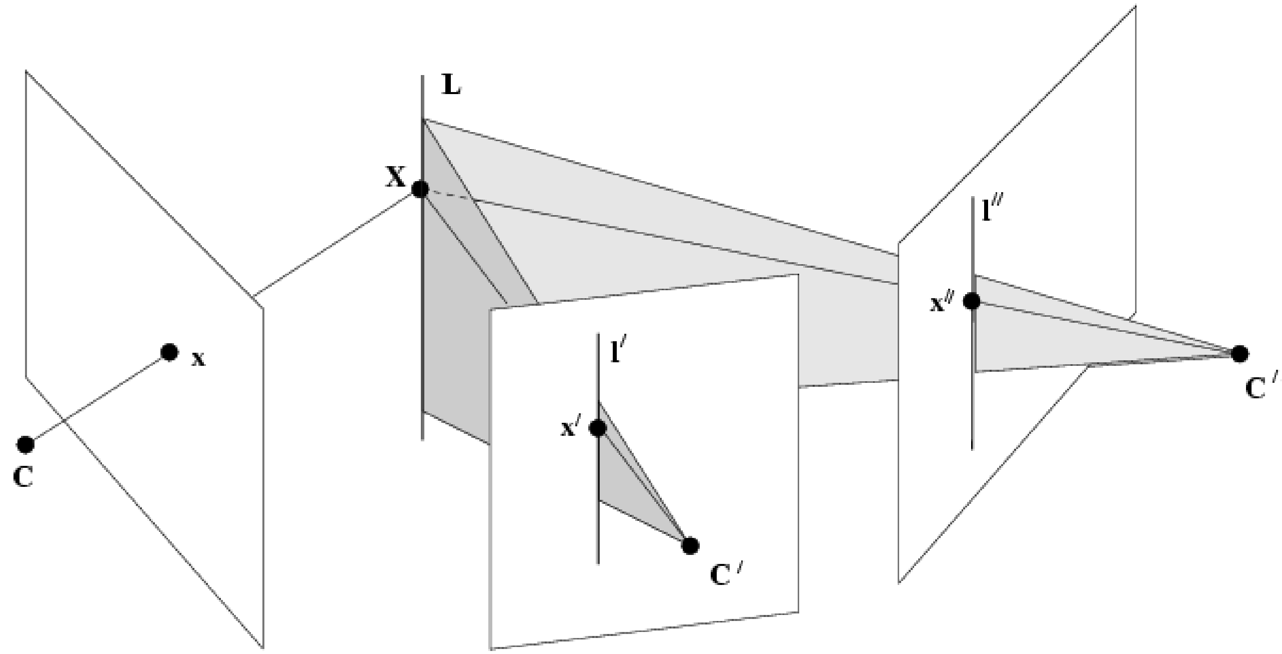


(d) $E_1(\hat{F}_2)$



(e) $E_1(\hat{F}_3)$

Segmentation of 3-D trifocal tensors



- Trilinear constraint when $\boldsymbol{x} \leftrightarrow \boldsymbol{\ell}' \leftrightarrow \boldsymbol{\ell}''$ correspond

$$\sum_{i,j,k} x^i l'_j l''_k \, T_i^{jk} = 0 \quad \Longleftrightarrow \quad \boldsymbol{x} \, \boldsymbol{\ell}' \, \boldsymbol{\ell}'' \, T_i = 0$$

- Multibody trilinear constraint and trifocal tensor

$$\prod_{i=1}^{n} (\boldsymbol{x} \, \boldsymbol{\ell}' \, \boldsymbol{\ell}'' \, T_i) = 0 \quad \Longleftrightarrow \quad \nu_n(\boldsymbol{x}) \nu_n(\boldsymbol{\ell}') \nu_n(\boldsymbol{\ell}'') \mathcal{T} = 0$$

Segmentation of 3-D trifocal tensors

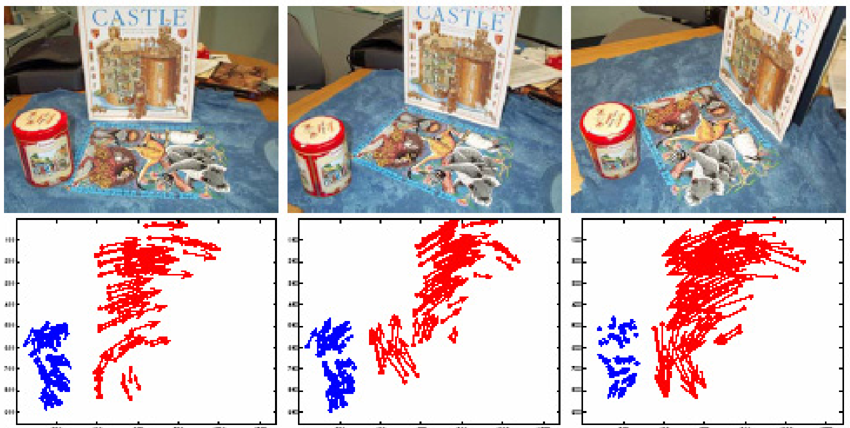


Figure 1: *Top: views 1-3 of a sequence with two rigid-body motions. Bottom: 2D displacements of the 140 correspondences from the current view ('o') to the next ('→').*

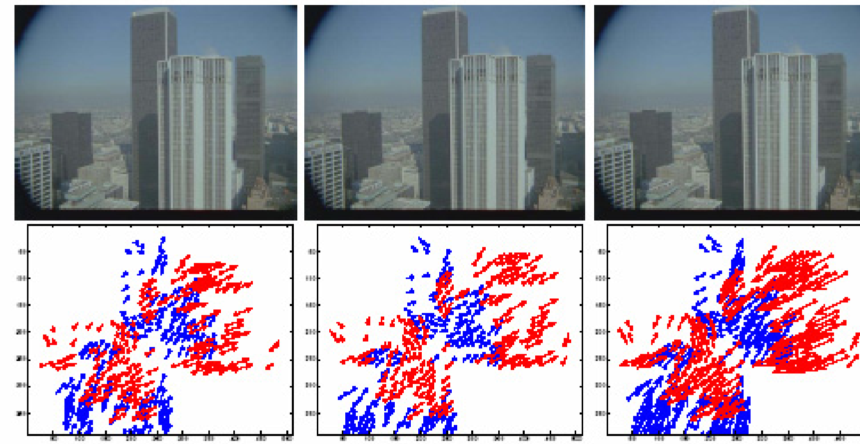


Figure 2: *Top: frames 1, 4 and 7 of the Wilshire sequence. Bottom: 2D displacement of the 328 original and flipped correspondences from current view ('○') to the next ('→').*

Table 1: *Percentage of misclassification of each algorithm.*

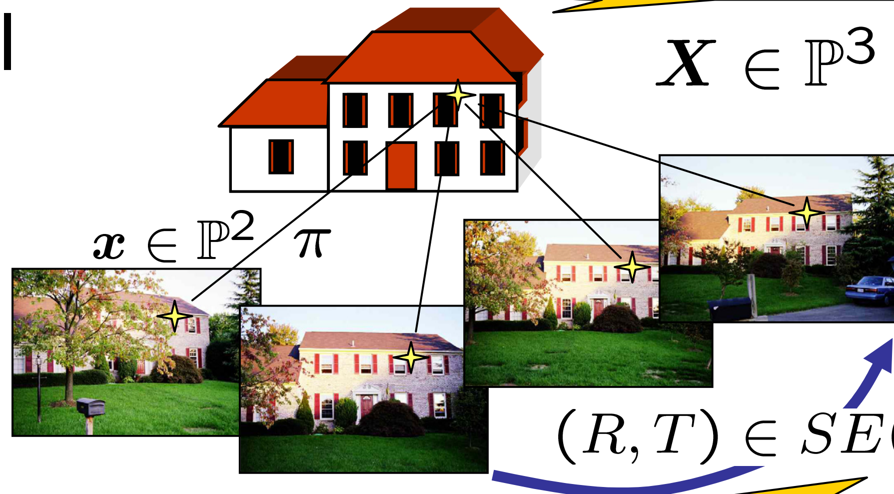
	K-means	Alg. I	Alg. II	Alg. II + K-means	Alg. II + K-means+EM
Tshirt-Book-Can	24.6%	24.3%	23.6%	7.1%	1.4%
Wilshire	39.5%	4.1%	2.5%	2.5%	0.0%

Segmentation of 3-D rigid-body motions

- Affine camera model

$$x_{fp} = \begin{bmatrix} R_f & T_f \end{bmatrix} X_p = \mathtt{A}_f X_p$$

  - p = point
  - f = frame



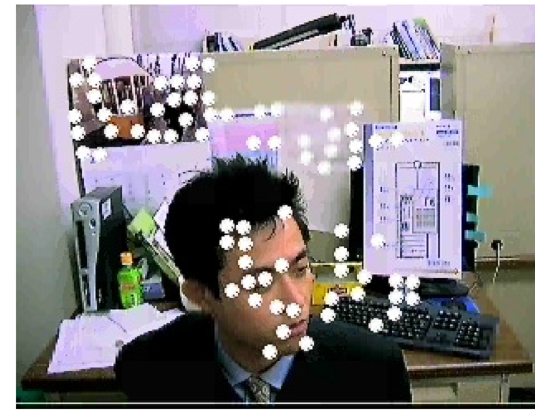
- Motion of one rigid-body lives in a 4-D subspace
(Boult and Brown '91, Tomasi and Kanade '92)

  - P = #points
  - F = #frames

$$\underbrace{\begin{bmatrix} x_{11} \cdots x_{1P} \\ \vdots \quad\quad \vdots \\ x_{F1} \cdots x_{FP} \end{bmatrix}}_{2F\times P}^{\mathtt{W}} = \underbrace{\begin{bmatrix} \mathtt{A}_1 \\ \vdots \\ \mathtt{A}_F \end{bmatrix}}_{2F\times 4}^{\mathtt{M}} \underbrace{\begin{bmatrix} X_1 \cdots X_P \end{bmatrix}}_{4\times P}^{\mathtt{S}^T}$$

Segmentation of 3-D rigid-body motions

- Sequence A Sequence B Sequence C

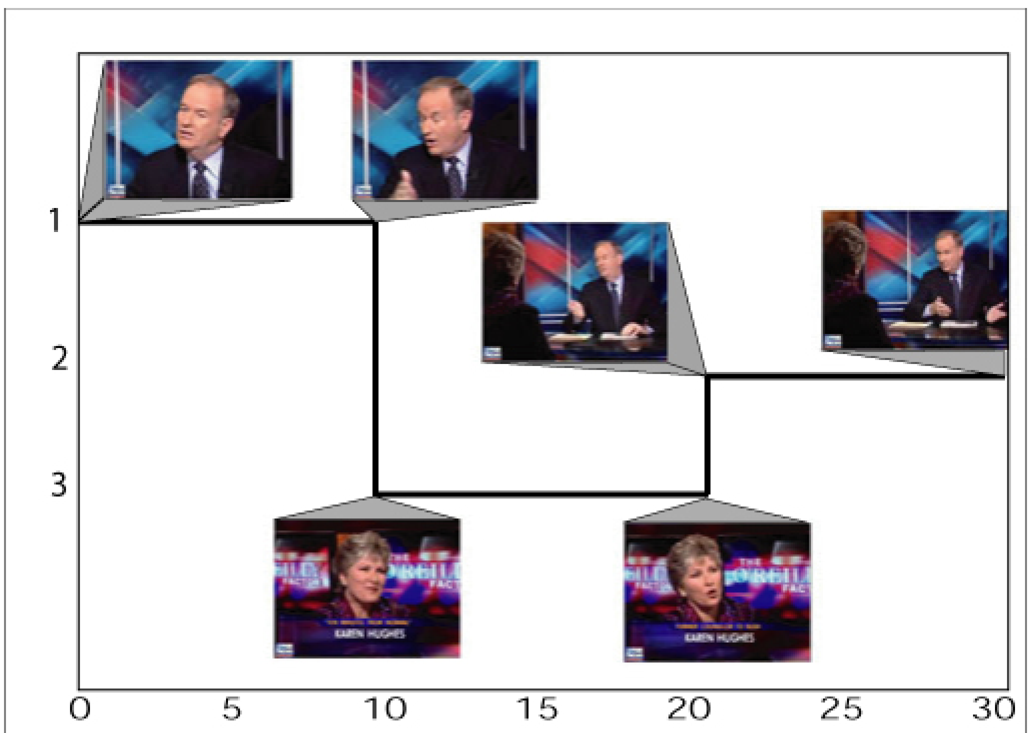


- Percentage of correct classification

Method	A	B	C
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 sep.	81.8%	99.7%	67.5%
Kanatani: multi-stage optimiz.	**100%**	**100%**	**100%**
Generalized PCA (GPCA)	**100%**	**100%**	**100%**

Temporal video segmentation

- Segmenting N=30 frames of a sequence containing n=3 scenes
  - Host
  - Guest
  - Both



- Image intensities are output of linear system

dynamics

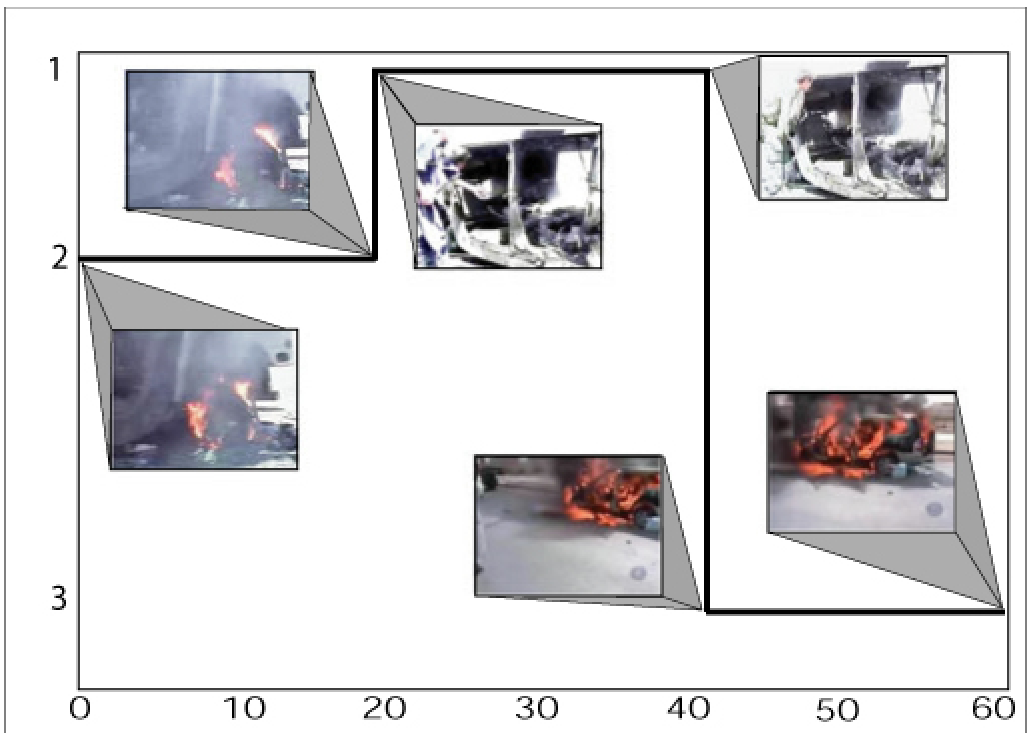
$$x_{t+1} = Ax_t + v_t$$
$$y_t = Cx_t + w_t$$

images, appearance

- Apply GPCA to fit n=3 observability subspaces

Temporal video segmentation

- Segmenting N=60 frames of a sequence containing n=3 scenes
  - Burning wheel
  - Burnt car with people
  - Burning car



- Image intensities are output of linear system

$$x_{t+1} = Ax_t + v_t$$
$$y_t = Cx_t + w_t$$

dynamics
images
appearance

- Apply GPCA to fit n=3 observability subspaces

Segmentation of dynamic textures

- Dynamic textures:
  - Output of a linear dynamical model (Soatto et al. 2001)

dynamics

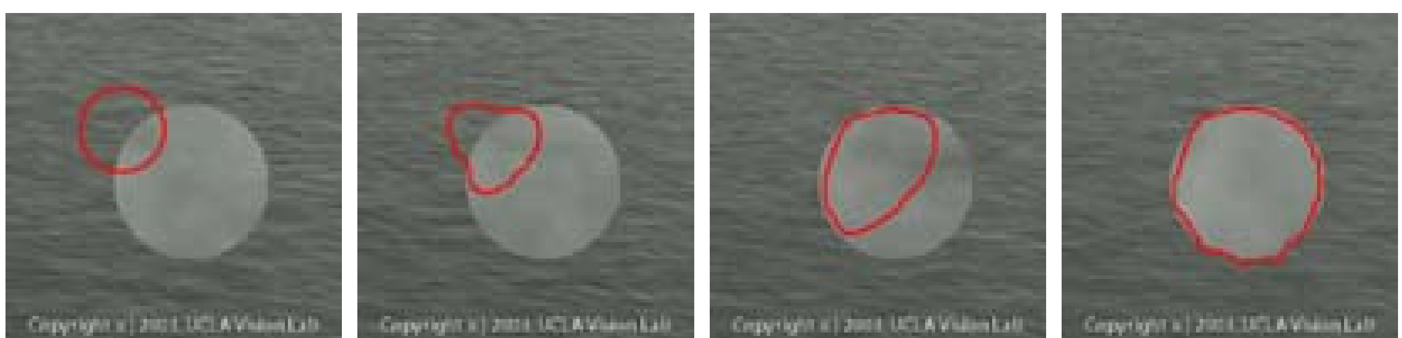
$$x_{t+1} = Ax_t + v_t$$

images

$$y_t = Cx_t + w_t$$

appearance

  - Dynamic texture segmentation using level-sets (Doretto et al. 2003)



Segmentation of dynamic textures

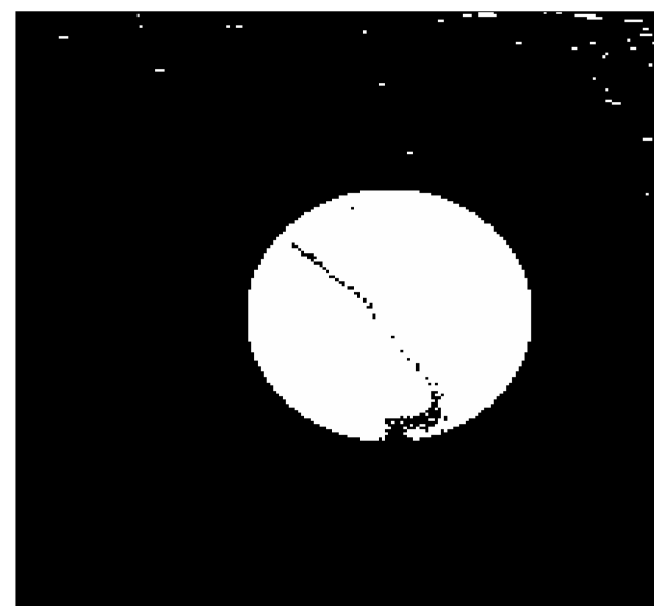
- One dynamic texture lives in the observability subspace

$$x_{t+1} = Ax_t + v_t$$
$$y_t = Cx_t + w_t$$

images

$$\begin{bmatrix} y_0 & \cdots & y_F \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{F-1} \end{bmatrix} \begin{bmatrix} x_0 & \cdots & x_F \end{bmatrix}$$

- Multiple textures live in multiple subspaces
  - Apply PCA to image intensities
  - Apply GPCA to projected data



Segmentation of moving dynamic textures

- Time varying model

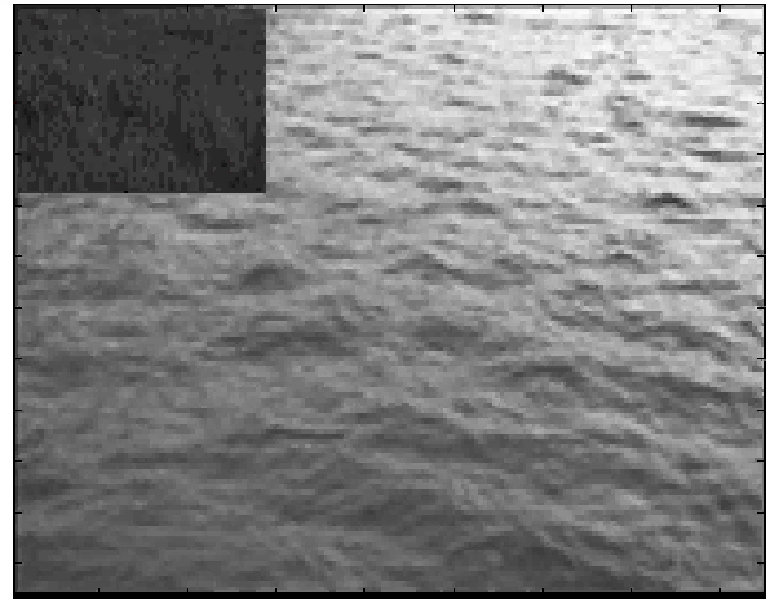
dynamics

$$x_{t+1} = Ax_t + v_t$$
$$y_t = C(t)x_t + w_t$$

images

appearance

- Segmentation of multiple moving subspaces
  - Apply PCA to intensities in a moving time window
  - Apply GPCA to projected data



Conclusions

- GPCA gives a unified algebraic approach for the analysis of dynamic scenes
  - Fit a polynomial to all image data
  - Differentiate the polynomial to obtain motion parameters

- Applies to most motion segmentation problems in vision
  - Two views
    - 2-D: translational, similarity, affine
    - 3-D: translational, fundamental matrices, homographies
  - Three views
    - Multibody trifocal tensor
  - Multiple views
    - Affine cameras

- Open problems
  - Multiple views perspective and central panoramic
  - Dealing with noise (bilinear and trilinear models)
  - Dealing with outliers (all models)

References

Generalized Principal Component Analysis

Estimation & Segmentation of Geometric Models

René Vidal (Biomedical Engineering, Johns Hopkins University)

Yi Ma (ECE, University of Illinois at Urbana-Champaign)

S. Shankar Sastry (EECS, University of California at Berkeley)

Thanks to collaborators and PhD students

- University of Illinois
  - Yi Ma
  - Kun Huang
- Johns Hopkins University
  - Jacopo Piazzi
  - Alvina Goh
  - Dheeraj Singaraju
  - Avinash Ravishandran
- Australian National University
  - Richard Hartley
- UCLA
  - Stefano Soatto
- UC Berkeley
  - Shankar Sastry