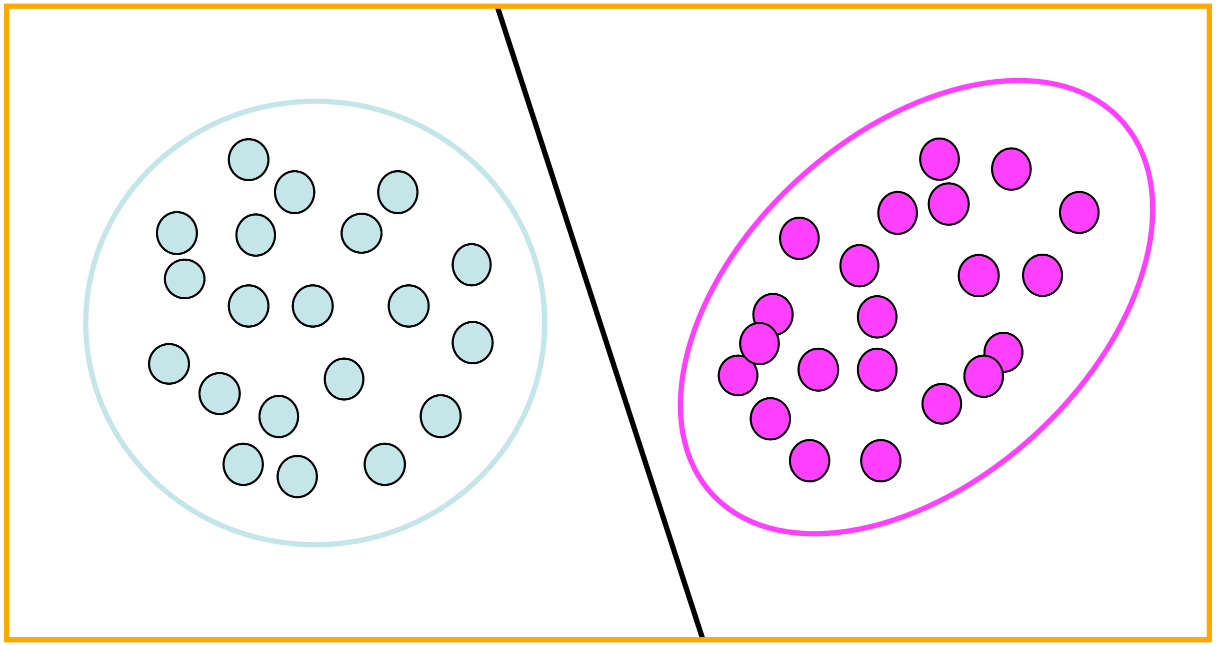
Multi-Manifold Learning

René Vidal
Center for Imaging Science
Institute for Computational Medicine
Johns Hopkins University

Data segmentation and clustering

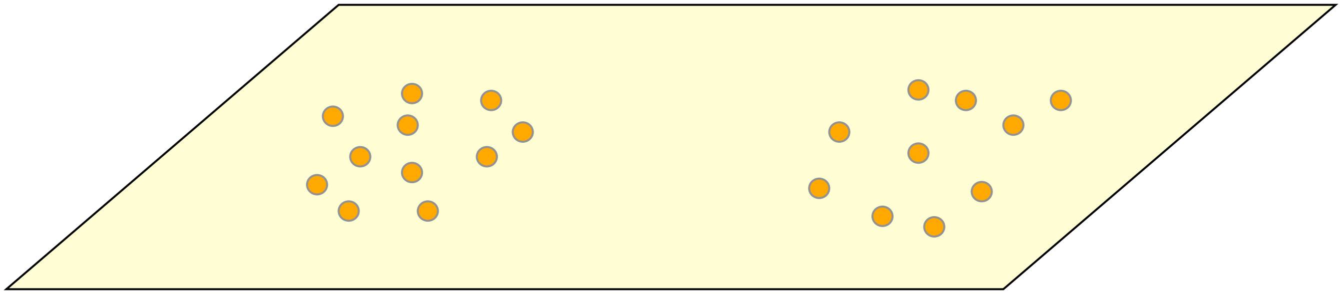
- Given a set of points, separate them into multiple groups



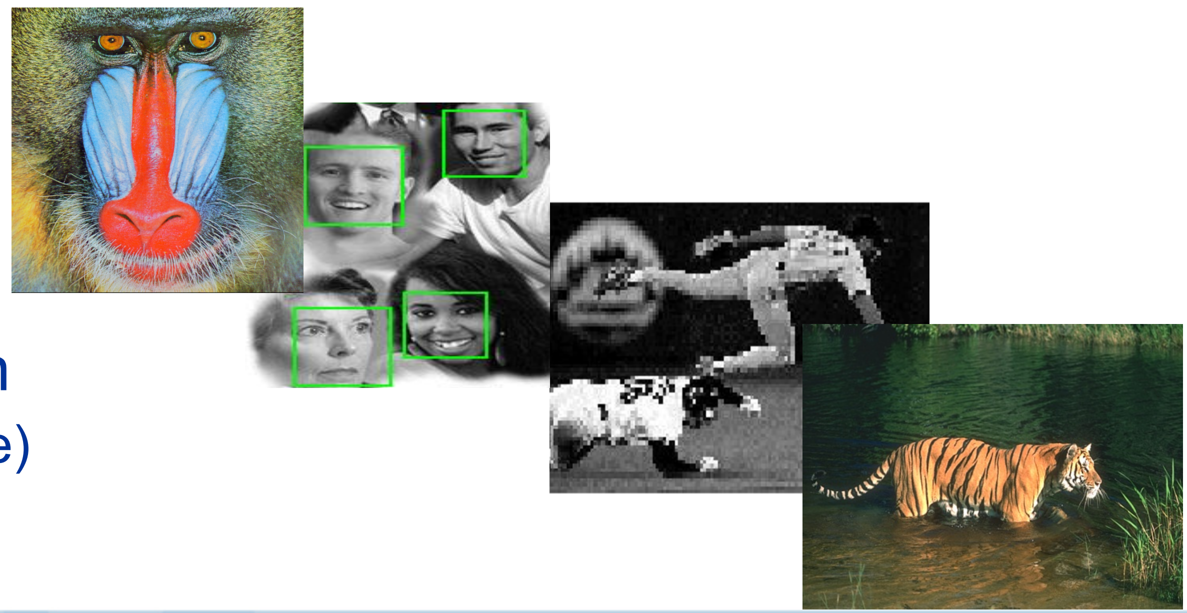
- Discriminative methods: learn boundary
- Generative methods: learn mixture model, using, e.g. Expectation Maximization

Dimensionality reduction and clustering

- In many problems data is high-dimensional: can reduce dimensionality using, e.g. Principal Component Analysis

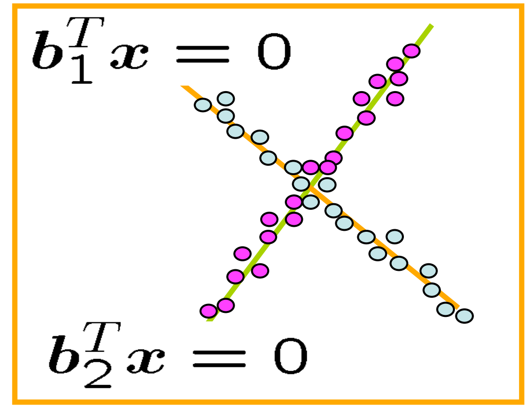


- Image compression
- Recognition
  - Faces (Eigenfaces)
- Image segmentation
  - Intensity (black-white)
  - Texture

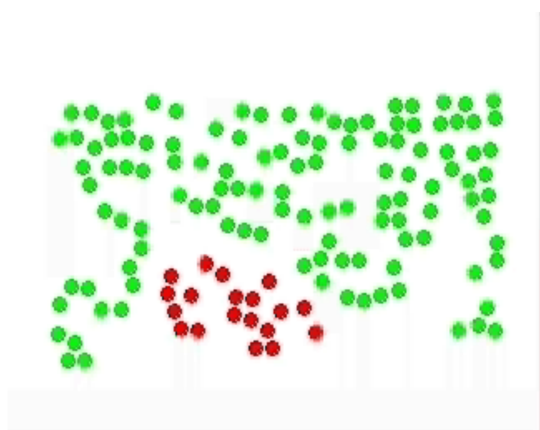


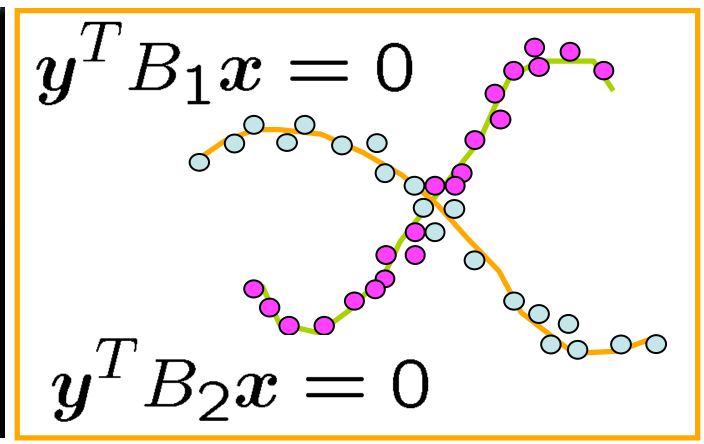
Segmentation problems in dynamic vision

- Segmentation of video and dynamic textures



- Segmentation of rigid-body motions

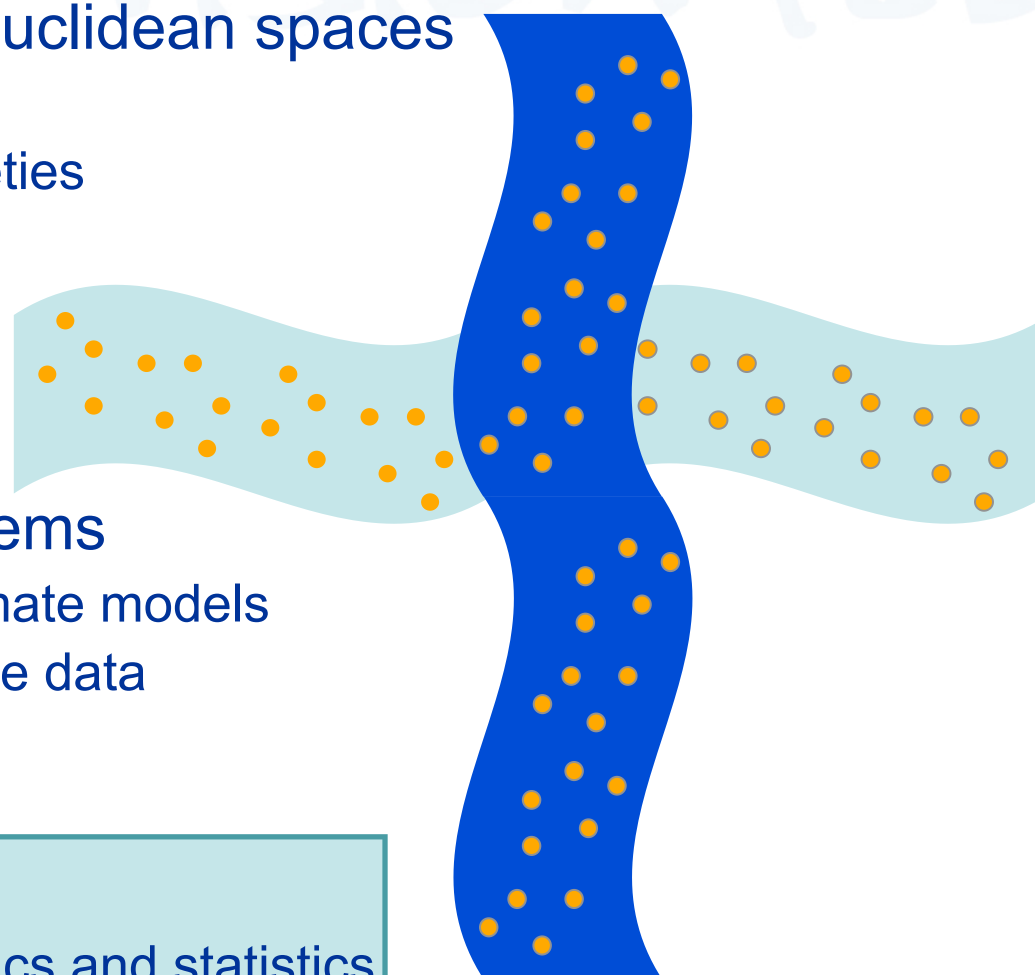




Center for IMAGING SCIENCE

Clustering data on non Euclidean spaces

- Clustering data on non Euclidean spaces
  - Mixtures of linear spaces
  - Mixtures of algebraic varieties
  - Mixtures of Lie groups



- “Chicken-and-egg” problems
  - Given segmentation, estimate models
  - Given models, segment the data
  - Initialization?

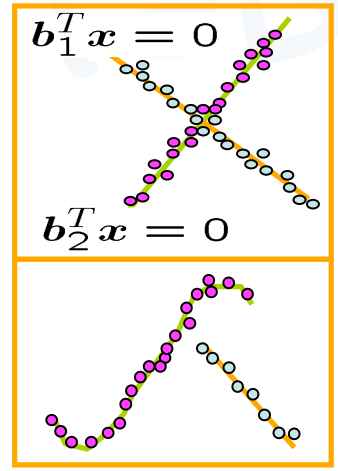
- Need to combine
  - Algebra/geometry, dynamics and statistics

Dimensionality reduction & clustering

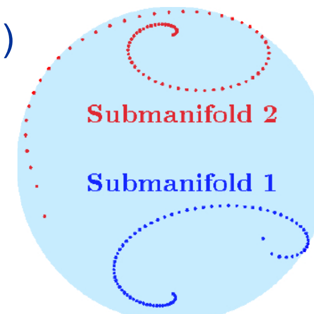
- Global techniques
  - Isomap (Tenenbaum et al. ‘00)
  - Kernel PCA (Schölkopf-Smola’98)
- Local techniques
  - Locally Linear Embedding (LLE) (Roweis-Saul ’00)
  - Laplacian Eigenmaps (LE) (Belkin-Niyogi ‘02)
  - Hessian LLE (HLLE) (Donoho-Grimes ‘03)
  - Local Tangent Space Alignment (Zha-Zhang’05)
  - Maximum Variance Unfolding (Weinberger-Saul ‘04)
  - Conformal Eigenmaps (Sha-Saul’05)
  - Structure Preserving Embedding (Shaw-Jebara’09)
- Clustering based on geometry
  - LLE+Spectral clustering (Polito-Perona ’02)
  - Spectral embedding and clustering (Brand-Huang’03)
  - Isomap+EM (Souvenir-Pless’05)
- Clustering based on dimension
  - Fractal dimension (Barbara-Chen’00)
  - Tensor voting (Mordohai-Medioni’05)
  - Dimension induced clustering (Gionis et al. ’05)
  - Translated Poisson mixtures (Haro et al.’08)

Talk outline

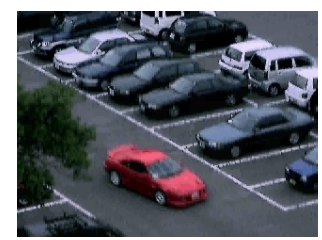
- Part I: Clustering Linear Manifolds
  - Generalized Principal Component Analysis (GPCA) (Vidal-Ma-Sastry '03, '04, '05)
  - Sparse Subspace Clustering (SCC) (Elhamifar-Vidal '09)

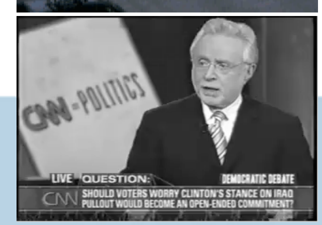


- Part II: Clustering Nonlinear Manifolds
  - Linear/nonlinear manifolds of Euclidean space (Goh-Vidal '07)
  - Submanifolds of a Riemannian manifold (Goh-Vidal '08)



- Part III: Applications
  - Segmentation of rigid body motions (Vidal-Tron-Hartley'08)
  - Segmentation of dynamic textures (Ghoreyshi-Vidal'06)
  - Segmentation of video shots (Vidal'08)
  - Segmentation of diffusion tensor images (Goh-Vidal'08)
  - Segmentation of probability density functions (Goh-Vidal'08)





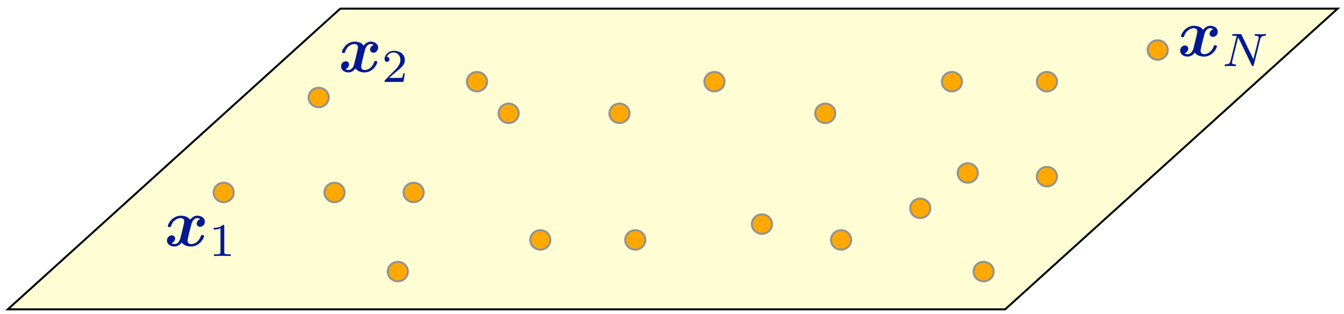
JHU vision lab

Part I
Generalized Principal Component Analysis

René Vidal
Center for Imaging Science
Institute for Computational Medicine
Johns Hopkins University

Principal Component Analysis (PCA)

- Given a set of points $x_1, x_2, \ldots, x_N$
  - Geometric PCA: find a subspace S passing through them
  - Statistical PCA: find projection directions that maximize the variance



- Solution (Beltrami'1873, Jordan'1874, Hotelling'33, Eckart-Householder-Young'36)

$$U \Sigma V^\top = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$$

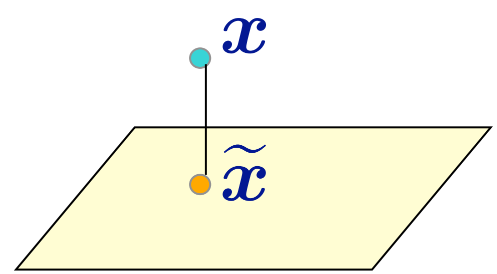
Basis for S

$\boxed{\dim(S) = \mathrm{rank}(U)}$

- Applications: data compression, regression, computer vision (eigenfaces), pattern recognition, genomics

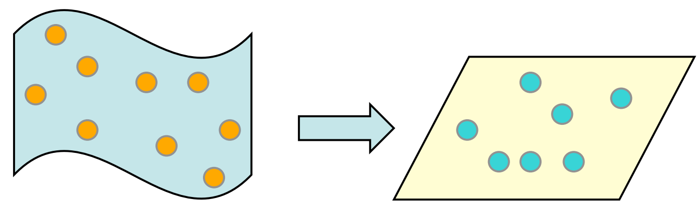
Extensions of PCA

- Higher order SVD (Tucker'66, Davis'02)
- Independent Component Analysis (Common '94)
- Probabilistic PCA (Tipping-Bishop '99)
  - Identify subspace from noisy data
  - Gaussian noise: standard PCA
  - Noise in exponential family (Collins et al.'01)

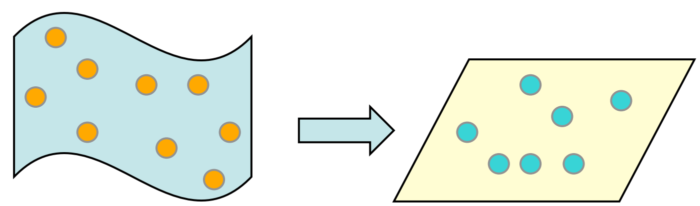


$$\boldsymbol{x} = \widetilde{\boldsymbol{x}} + \text{noise}$$

- Nonlinear dimensionality reduction
  - Multidimensional scaling (Torgerson'58)
  - Locally linear embedding (Roweis-Saul '00)
  - Isomap (Tenenbaum '00)



- Nonlinear PCA (Scholkopf-Smola-Muller '98)
  - Identify nonlinear manifold by applying PCA to data embedded in high-dimensional space

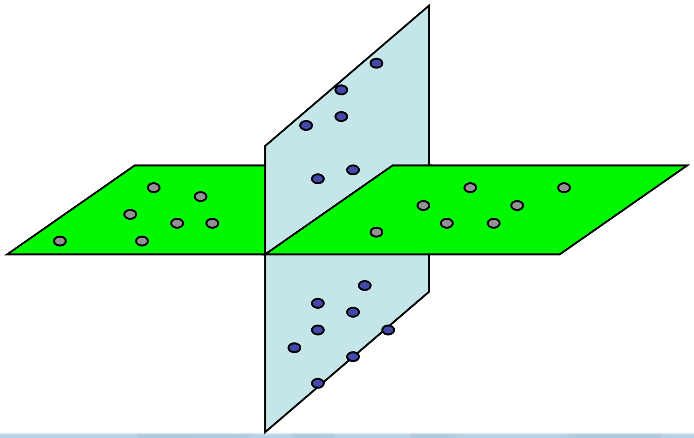


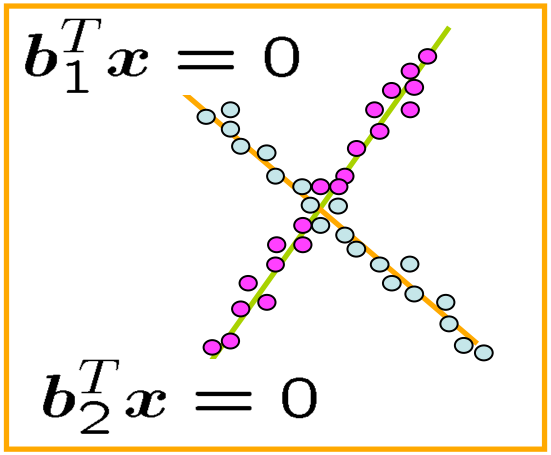
- Principal Curves and Principal Geodesic Analysis (Hastie-Stuetzle'89, Tishbirany '92, Fletcher '04)

Generalized Principal Component Analysis

- Given a set of points lying in 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

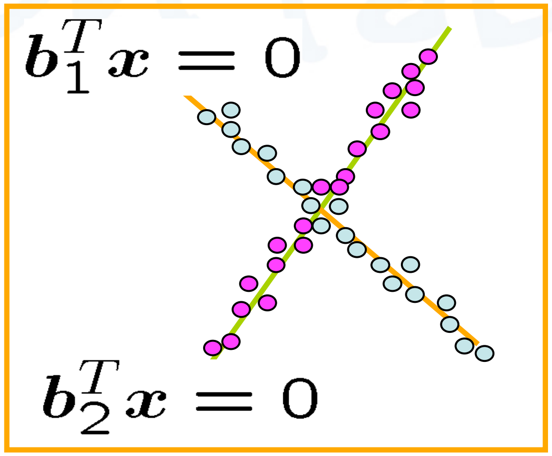




Prior work on subspace clustering

- Iterative algorithms
  - RANSAC (Leonardis et al. '02, Haralik-Harpaz '07)
  - K-subspaces (Kamhalta-Leen '94, Ho et al. '03)

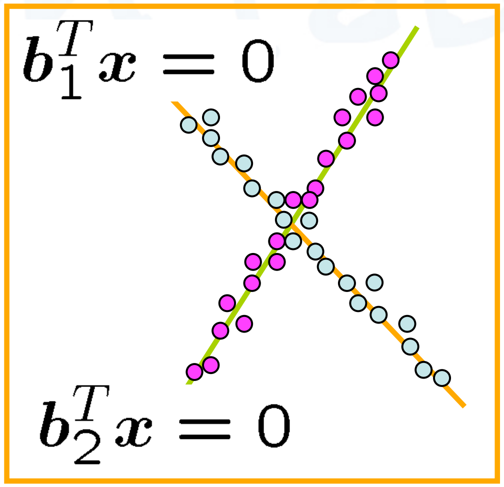
- Probabilistic approaches
  - Mixtures of PPCA (Tipping-Bishop '99, Grubber-Weiss '04)
  - Multi-Stage Learning (Kanatani '04)
  - Agglomerative Lossy Compression (Ma et al. '07)



- Initialization
  - Geometric approaches: 2 planes in R^3 (Shizawa-Maze '91)
  - Factorization-based approaches for independent subspaces of equal dimension (Boult-Brown '91, Costeira-Kanade '98, Gear '08, Kanatani '01)
  - Spectral clustering based approaches (Zelnik-Manor '03, Yan-Pollefeys '06, Govindu'05, Agarwal et al. '05, Fan-Wu '06, Chen-Lerman'08)

Spectral clustering-based approaches

- 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 subspace to be far



- Local subspace affinity (LSA) (Yan-Pollefeys ‘06)
  - Use the angles between locally fitted subspaces as similarity
  - Has problems with intersecting subspaces
- Spectral curvature clustering (SCC) (Chen-Lerman ’08)
  - Define multiway similarity as normalized volume of d+1 points
  - Suffers from curse of dimensionality

Basic ideas behind GPCA

- Towards an analytic solution to subspace clustering
  - 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?

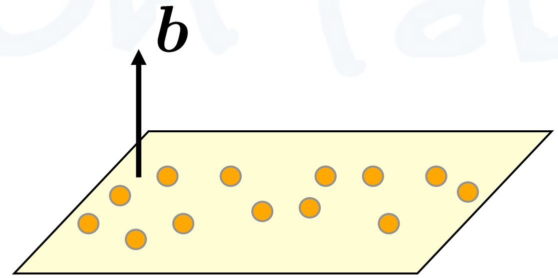
- Will consider the most general case
  - Subspaces of unknown and possibly different dimensions
  - Subspaces may intersect arbitrarily (not only at the origin)

- GPCA is 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

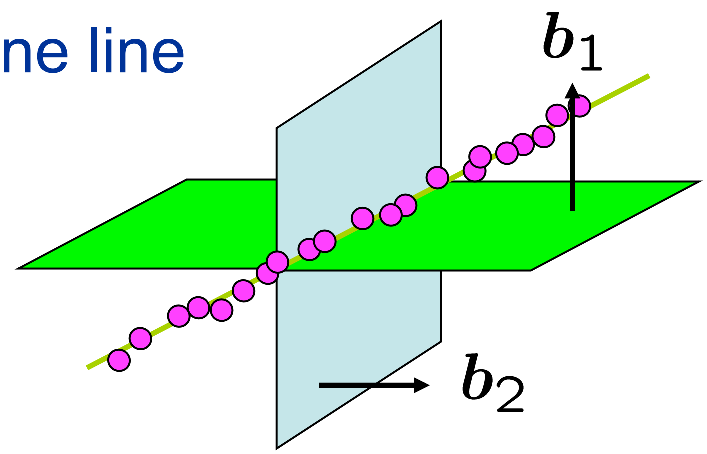
Representing one subspace

- 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
  - Set of polynomials of degree 1

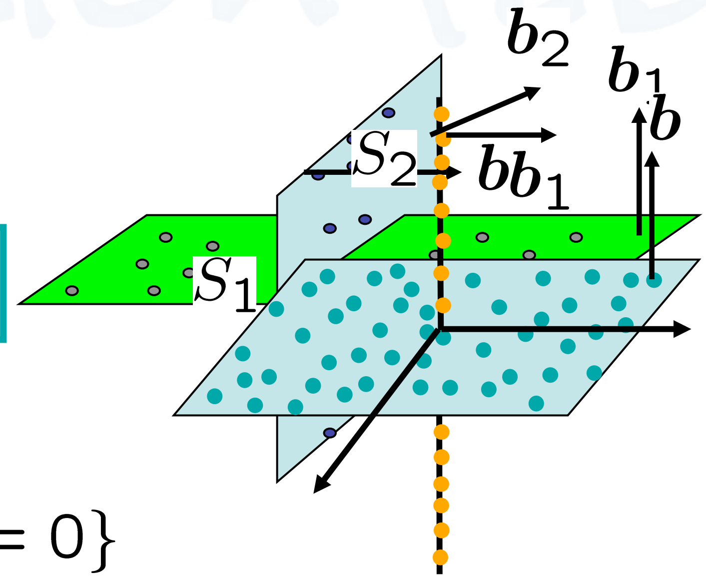
$$S = \{\boldsymbol{x} : B^T\boldsymbol{x} = 0\}$$

Representing *n* subspaces

- Two planes

$$(\boldsymbol{b}_1^T \boldsymbol{x} = 0) \; \boxed{\text{or}} \; (\boldsymbol{b}_2^T \boldsymbol{x} = 0)$$

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



- One plane and one line
  - Plane: $S_1 = \{\boldsymbol{x} : \boldsymbol{b}^T \boldsymbol{x} = 0\}$
  - Line: $S_2 = \{\boldsymbol{x} : \boldsymbol{b}_1^T \boldsymbol{x} = \boldsymbol{b}_2^T \boldsymbol{x} = 0\}$

$$S_1 \cup S_2 = \{\boldsymbol{x} : (\boldsymbol{b}^T \boldsymbol{x} = 0) \; \boxed{\text{or}} \; (\boldsymbol{b}_1^T \boldsymbol{x} = \boldsymbol{b}_2^T \boldsymbol{x} = 0\}$$

De Morgan's rule

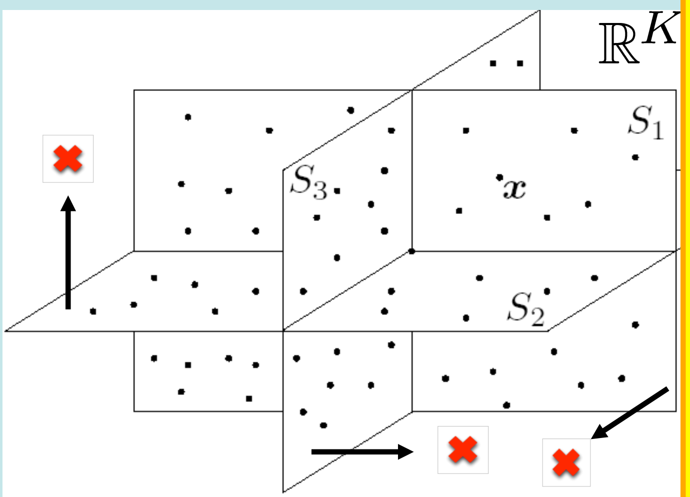
$$S_1 \cup S_2 = \{\boldsymbol{x} : \boxed{(\boldsymbol{b}^T \boldsymbol{x})(\boldsymbol{b}_1^T \boldsymbol{x}) = 0} \; \boxed{\text{and}} \; \boxed{(\boldsymbol{b}^T \boldsymbol{x})(\boldsymbol{b}_2^T \boldsymbol{x}) = 0}\}$$

- 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

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

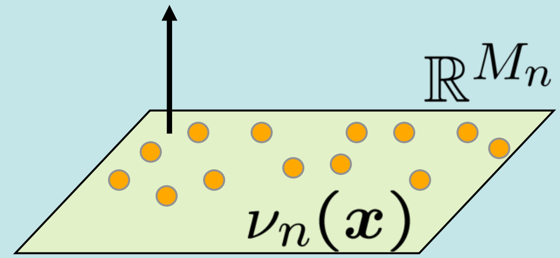


Veronese map

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

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

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



$$M_n = \begin{pmatrix} n+K-1 \\ n \end{pmatrix}$$

- 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
  - Number of subspaces
  - Basis are normal vectors

$$c^T \nu_n(x) = (b_1^T x) \cdots (b_n^T x)$$

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

$$b_1, b_2, \cdots b_n$$

$$c \in \mathbb{R}^{M_n}$$

$$b_1 \quad b_2 \quad \cdots \quad b_n$$

Polynomial Factorization (GPCA-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

$$\boldsymbol{c} \in \mathbb{R}^{M_n}$$

$$\boldsymbol{b}_1 \quad \boldsymbol{b}_2 \quad \ldots \quad \boldsymbol{b}_n$$

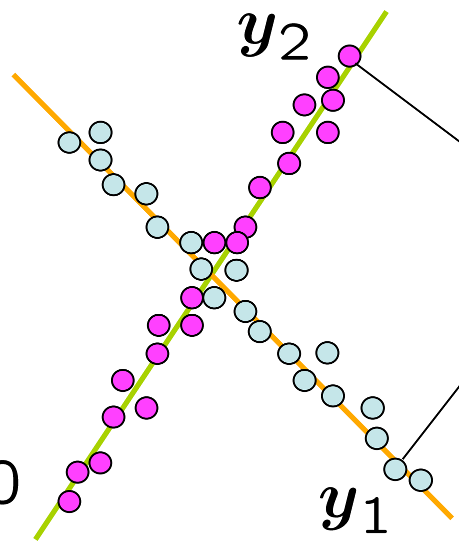
Polynomial Differentiation (GPCA-PDA) [CVPR'04]

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

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

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

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



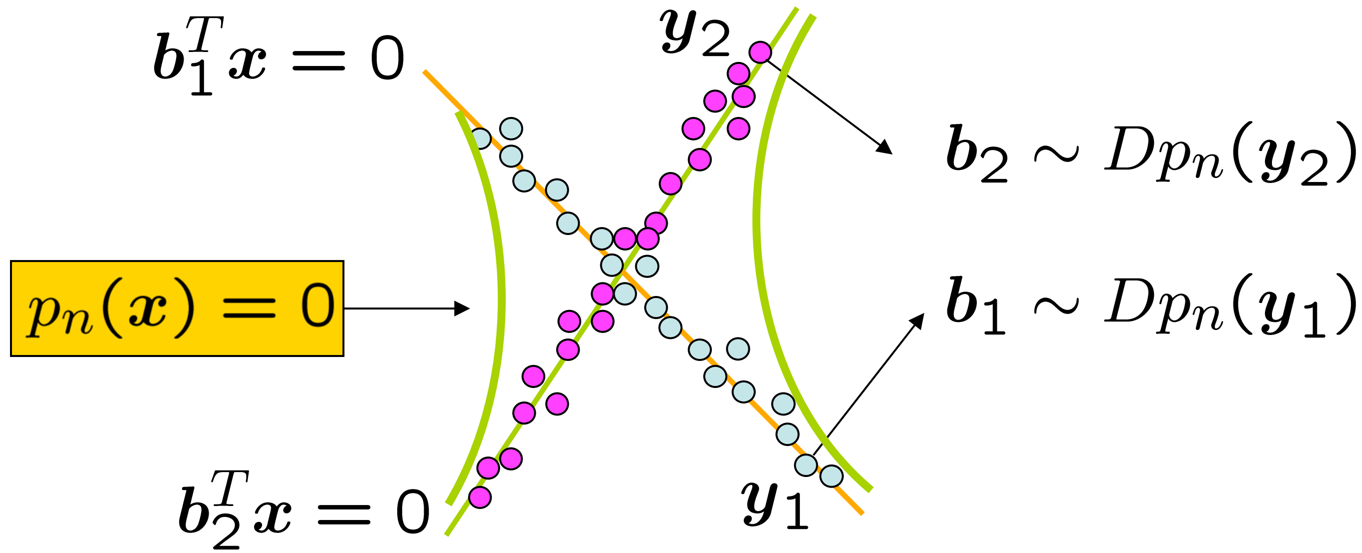
$$\boldsymbol{b}_2 \sim Dp_n(\boldsymbol{y}_2)$$

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

- 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?

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

GPCA for hyperplane segmentation

- Coefficients of the polynomial can be computed from null space of embedded data matrix
  - Solve using least squares
  - N = #data points

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

- Number of subspaces can be computed from the rank of embedded data matrix

$$n = \min\{i : \mathrm{rank}(L_i) = M_i - 1\}$$

- Normal to the subspaces $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots \boldsymbol{b}_n$ can be computed from the derivatives of the polynomial

$$\boldsymbol{c} \in \mathbb{R}^{M_n}$$

$$\boldsymbol{b}_1 \quad \boldsymbol{b}_2 \quad \ldots \quad \boldsymbol{b}_n$$

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

GPCA for subspaces of different dimensions

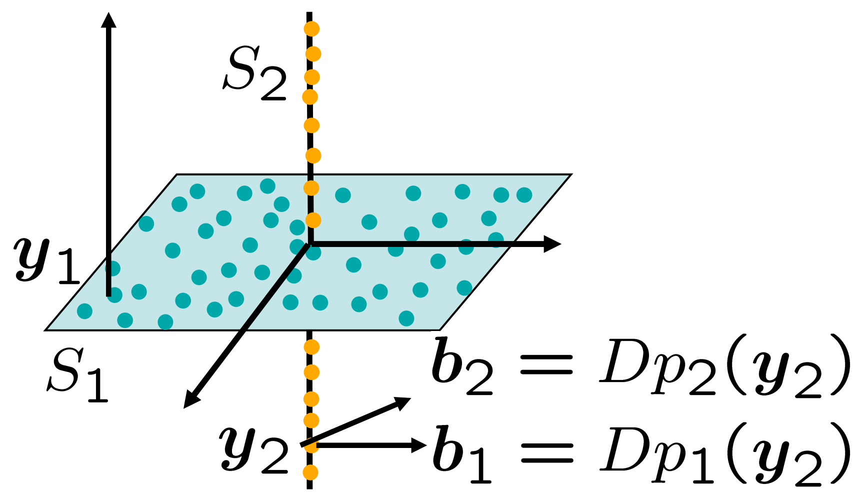
- There are multiple polynomials fitting the data

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

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

- The derivative of each polynomial gives a different normal vector

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



- 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

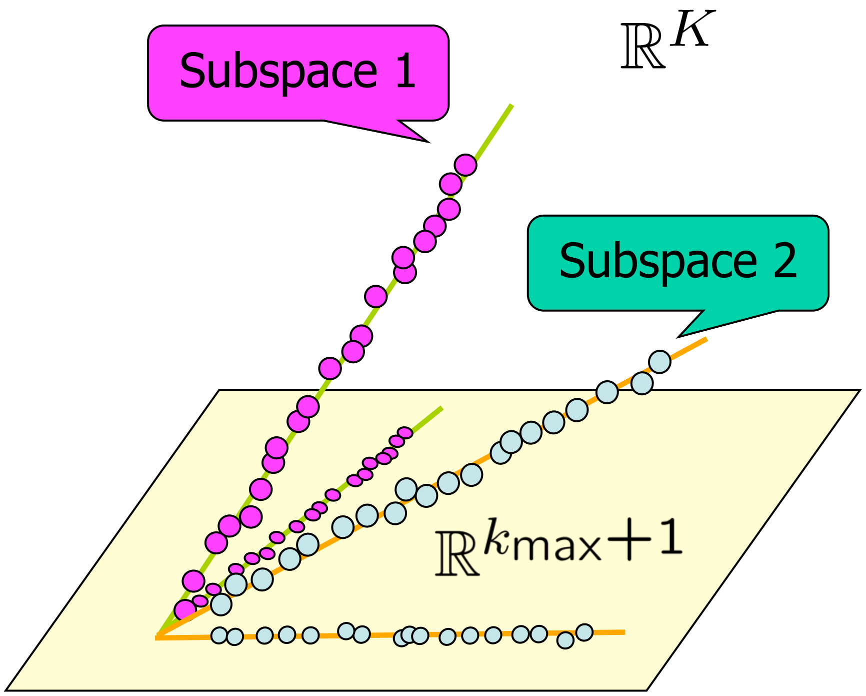
- Minimum number of points
  - K = dimension of ambient space
  - n = number of subspaces

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

- In practice the dimension of each subspace ki 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

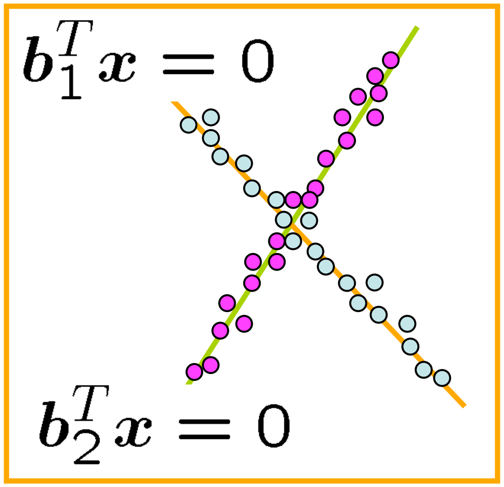


- Open problem: how to choose projection?
  - PCA?

GPCA 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 subspace 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 $\mathcal{D}_{ij} \doteq \langle B_i, B_j \rangle$

Extensions of GPCA to nonlinear manifolds

- Segmentation of quadratic forms (Rao-Yang-Ma '05, '06)

$$\prod_{i=1}^{n}(\boldsymbol{x}^\top A\boldsymbol{x}) = \nu_n(\boldsymbol{x})^\top \mathcal{A}\nu_n(\boldsymbol{x}) = 0$$

- Segmentation of bilinear surfaces (Vidal-Ma-Soatto-Sastry '03, '06)

$$\prod_{i=1}^{n}(\boldsymbol{x}_2^\top F_i\boldsymbol{x}_1) = \nu_n(\boldsymbol{x}_2)\mathcal{F}\nu_n(\boldsymbol{x}_1) = 0 \qquad F_i \in so(3)\times SO(3)$$

- Segmentation of mixed linear and bilinear (Singaraju-Vidal '06)

$$\prod_{i=1}^{n}(\boldsymbol{b}_i^\top \boldsymbol{x}_1)\prod_{j=1}^{m}(\boldsymbol{x}_2^\top A_j\boldsymbol{x}_1) = \nu_n(\boldsymbol{x}_2)^\top \mathcal{A}\nu_{n+m}(\boldsymbol{x}_1) = 0$$

- Segmentation of trilinear surfaces (Hartley-Vidal '05, '08)

$$\prod_{i=1}^{n}(\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3 T_i) = \nu_n(\boldsymbol{x}_1)\nu_n(\boldsymbol{x}_2)\nu_n(\boldsymbol{x}_3)\mathcal{T} = 0$$

Center for IMAGING SCIENCE

JHU vision lab

Part II: Sparse Subspace Clustering (SCC)

René Vidal
Center for Imaging Science
Institute for Computational Medicine
Johns Hopkins University

THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins

Center for IMAGING SCIENCE

Basic ideas behind SSC

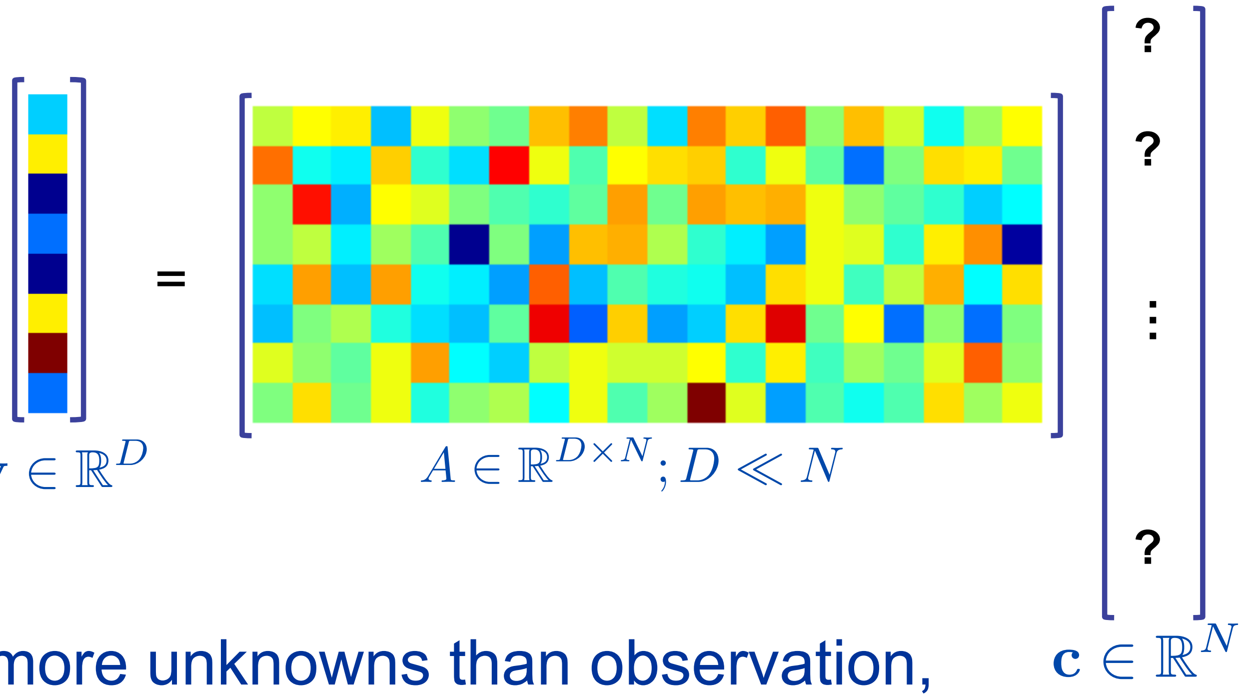
- We propose a subspace clustering algorithm based on *sparse representation theory*
  - Obtain a sparse representation of a point using *convex optimization*
  - Use sparse representation to define *similarity matrix*
  - Obtain segmentation using *spectral clustering*

- Main contribution
  - Extend sparse representation results from one to multiple subspaces

- The Sparse Subspace Clustering (SSC) algorithm
  - Is *provably correct* with perfect and noisy data
  - Can handle data corrupted by *noise*, *outliers* and *missing entries*
  - Significantly *outperforms state-of-the-art* algorithms for segmenting videos with multiple moving objects

Sparse representation: motivation

- Underdetermined system of linear equations: $\mathbf{y} = A\mathbf{c}$



- Many more unknowns than observation, thus the solution is not unique

- Classical solution: ℓ_2 norm, $\min \|\mathbf{c}\|_2 \text{ subject to } \mathbf{y} = A\mathbf{c}$

Sparse representation: L0 versus L1

- What if we know that the solution is sparse?
  - Look for the sparsest solution:

$$(\mathrm{P}_0) \quad \min \|\mathbf{c}\|_0 \quad \text{subject to} \quad \mathbf{y} = A\mathbf{c} \qquad \text{Intractable!}$$

  - $\|\mathbf{c}\|_0$: number of nonzero elements

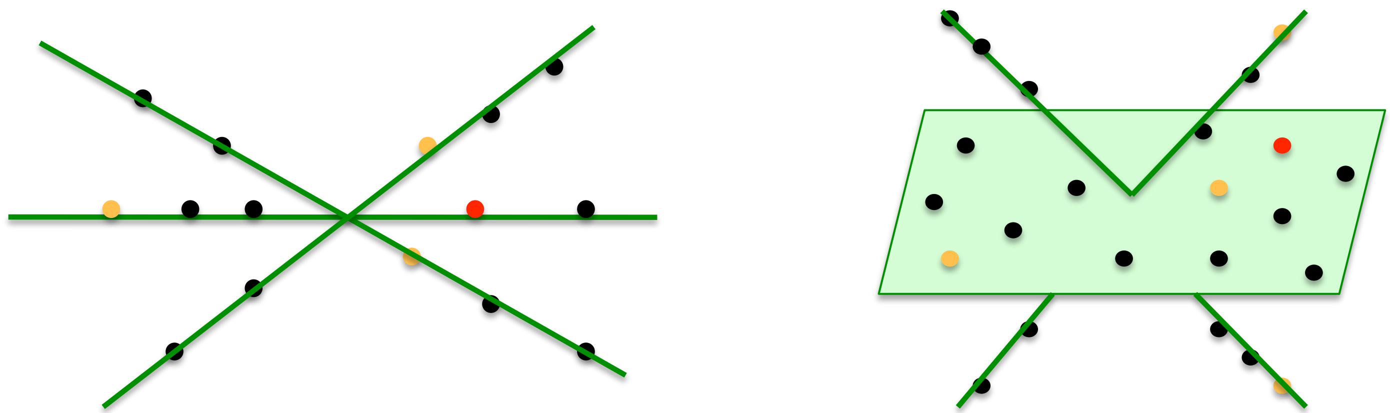
- Convex relaxation

$$(\mathrm{P}_1) \quad \min \|\mathbf{c}\|_1 \quad \text{subject to} \quad \mathbf{y} = A\mathbf{c} \qquad \text{Efficient!}$$

- P_0 and P_1 are equivalent under some conditions on A
  - mutual coherence (Tropp'04),
  - cumulative coherence (Tropp'04),
  - restricted isometry constant (Candes & Tao'05), …

Sparse subspace clustering: intuition

- Idea: a point $\mathbf{y} \in \mathbb{R}^D$ from subspace S of dimension $d \ll D$ can write itself as a linear combination of d points in the same subspace $\longrightarrow$ sparse representation! (Ma et al.)



- Under what *conditions on the subspaces* does the sparsest representation of a point come from *points in the same subspace*?

Sparse subspace clustering

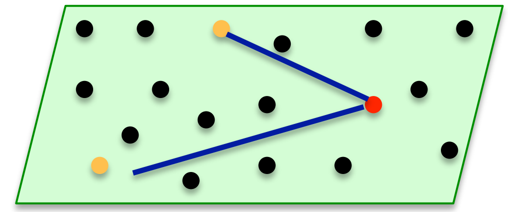
- *n* linear subspaces $\{S_i \subset \mathbb{R}^D\}_{i=1}^n$ are called *independent* if
$$\dim(\bigoplus_{i=1}^n S_i) = \sum_{i=1}^n \dim(S_i)$$

- *Theorem (Elhamifar & Vidal CVPR '09)*
For data points drawn from a union of independent linear subspaces, the sparsest representation of a point comes from points in the same subspace. This representation can be recovered by solving a convex program.

- Sketch of proof: Let $Y_{D\times N} = [Y_1, Y_2, \ldots, Y_n]\Gamma$ be the data matrix. Let $\mathbf{y} \in Y_1$. We can write $\mathbf{y} = Y_1\mathbf{c}_1 + [Y_2, \ldots, Y_n]\mathbf{c}'$. Now, since
$$\mathrm{Range}(Y_1) \cap \mathrm{Range}([Y_2, \ldots, Y_n]) = \{0\} \quad \Rightarrow \quad [Y_2, \ldots, Y_n]\mathbf{c}' = 0$$
$\mathbf{c}'$ does not contribute to $\mathbf{y}$

Sparse Subspace Clustering Algorithm

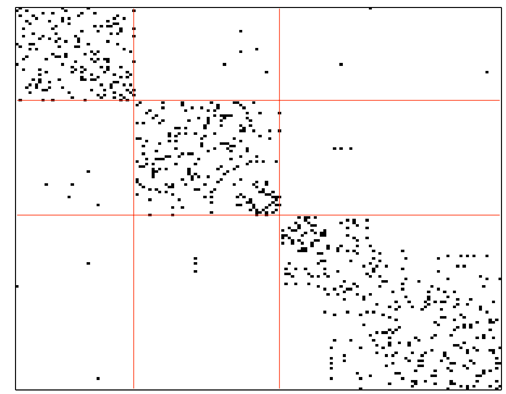
- Project D-dim data onto m-dim space, using random projection matrix with i.i.d. entries
  - Symmetric Bernoulli distribution
  - Zero-mean Normal distribution with variance m



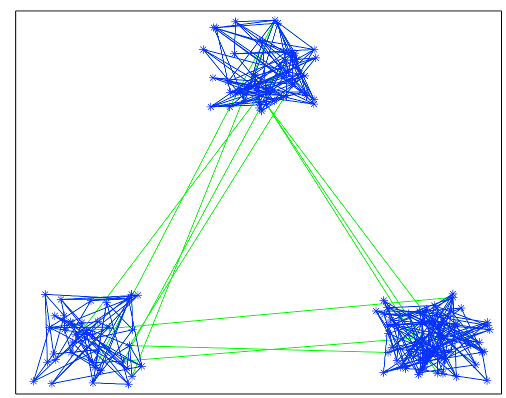
- Represent data points as nodes in graph G
  - Find the sparse coefficient vectors $\{\mathbf{c}_i\}_{i=1}^N$
  - Connect nodes i and j by an edge with weight

$$|c_{ij}| + |c_{ji}|$$

  - Each node connects itself to nodes in the same subspace => get a perfect block-diagonal matrix



- Spectral clustering: apply K-means to the smallest eigenvectors of the Laplacian of G



Extensions of SSC: affine, noise, missing

- Theorem (Elhamifar & Vidal'09)
  For data points drawn from a union of independent affine subspaces, the sparsest representation of a point comes from points in the same subspace. The SR can be found as

$$(\mathrm{P}_2) \quad \min \|\mathbf{c}_i\|_1 \quad \text{subject to} \quad \mathbf{y}_i = Y\mathbf{c}_i \text{ and } \mathbf{c}_i^\top 1 = 1$$

- When the data are *corrupted with noise*
  - $\min \|\mathbf{c}_i\|_1 \quad \text{subject to} \quad \|\mathbf{y}_i - Y\mathbf{c}_i\|_2 < \epsilon$.
  - $\min \|\mathbf{c}_i\|_1 + \mu \|\mathbf{y}_i - Y\mathbf{c}_i\|_2$ (LASSO)

- When the data have *missing entries* (Rao et. al '08)
  - Let $I \subset \{1, \dots, D\}$ be the indices of the missing entries in $\mathbf{y} \in \mathbb{R}^D$
  - Form $\tilde{\mathbf{y}} \in \mathbb{R}^{D-|I|}$ and $\tilde{Y} \in \mathbb{R}^{D-|I| \times N}$ by eliminating rows of $\mathbf{y}$ and Y indexed by I, and solve the same optimization problems

Extensions of SSC: outliers (Rao et. al '08)

- Let $\tilde{\mathbf{y}} = \mathbf{y} + \mathbf{e}$ be a corrupted vector with $\mathbf{e} \in \mathbb{R}^D$ being a sparse vector of outlying entries
- We can write: $\tilde{\mathbf{y}} = Y\mathbf{c} + \mathbf{e} = \begin{bmatrix} Y & I_D \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{e} \end{bmatrix}$
- The coefficient vector $\begin{bmatrix} \mathbf{c}^\top & \mathbf{e}^\top \end{bmatrix}^\top$ is still sparse!
- Perfect data: recover the sparse coefficients from

$$\min \left\| \begin{bmatrix} \mathbf{c} \\ \mathbf{e} \end{bmatrix} \right\|_1 \quad \text{subject to} \quad \tilde{\mathbf{y}} = \begin{bmatrix} Y & I_D \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{e} \end{bmatrix}$$

- Noisy data: recover the sparse coefficients from

$$\min \left\| \begin{bmatrix} \mathbf{c} \\ \mathbf{e} \end{bmatrix} \right\|_1 + \mu \left\| \tilde{\mathbf{y}} - \begin{bmatrix} Y & I_D \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{e} \end{bmatrix} \right\|_2$$

GPCA versus SCC

	GPCA	SCC
Type of subspaces	Arbitrary	Independent, disjoint
Number of subspaces	Can be estimated	Handled by spectral clustering
Subspace dimensions	Can be unknown and different, but … noise	Can be unknown and different, but …
Noise	Moderate	Yes
Outliers	No	Yes
Missing entries	No	Yes
Complexity	Exponential in the number of subspaces	One LASSO per point + Spectral clustering

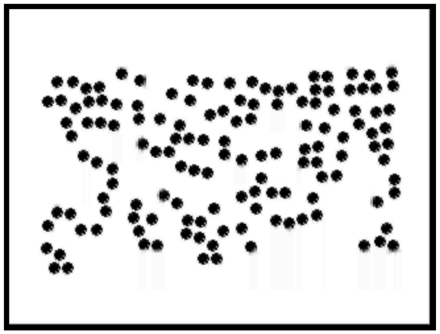
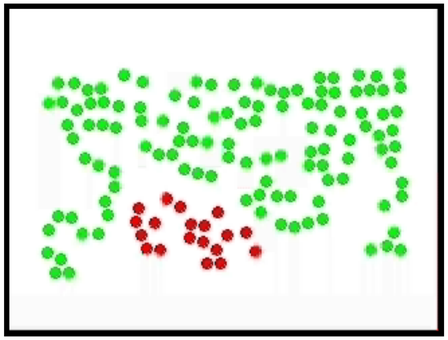
JHU vision lab

Part III: Applications in Computer Vision

René Vidal
Center for Imaging Science
Institute for Computational Medicine
Johns Hopkins University

3-D motion segmentation problem

- Given a set of point correspondences in multiple views, determine
  - Number of motion models
  - Motion model: affine, homography, fundamental matrix, trifocal tensor
  - Segmentation: model to which each pixel belongs

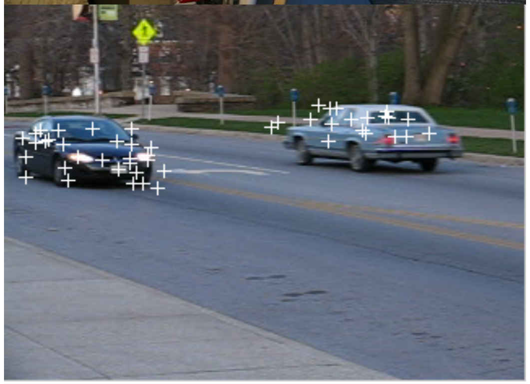




- Motion of one rigid-body lives in a 4-D linear subspace
  (Boult and Brown '91, Tomasi and Kanade '92)
  - P = #points
  - F = #frames

$$\mathtt{W} = \mathtt{M} \quad \mathtt{S}^T$$

$$\underbrace{\begin{bmatrix} \boldsymbol{x}_{11} \cdots \boldsymbol{x}_{1P} \\ \vdots \quad\quad \vdots \\ \boldsymbol{x}_{F1} \cdots \boldsymbol{x}_{FP} \end{bmatrix}}_{2F \times P} = \underbrace{\begin{bmatrix} \mathtt{A}_1 \\ \vdots \\ \mathtt{A}_F \end{bmatrix}}_{2F \times 4} \underbrace{\begin{bmatrix} \boldsymbol{X}_1 \cdots \boldsymbol{X}_P \end{bmatrix}}_{4 \times P}$$

Hopkins 155 motion segmentation database



- Collected 155 sequences (Tron-Vidal '07)
  - 120 with 2 motions
  - 35 with 3 motions
- Types of sequences
  - Checkerboard sequences: mostly full dimensional and independent motions
  - Traffic sequences: mostly degenerate (linear, planar) and partially dependent motions
  - Articulated sequences: mostly full dimensional and partially dependent motions
- Point correspondences
  - In few cases, provided by Kanatani & Pollefeys
  - In most cases, extracted semi-automatically with OpenCV

Results on the Hopkins 155 database

- 2 motions, 120 sequences, 266 points, 30 frames

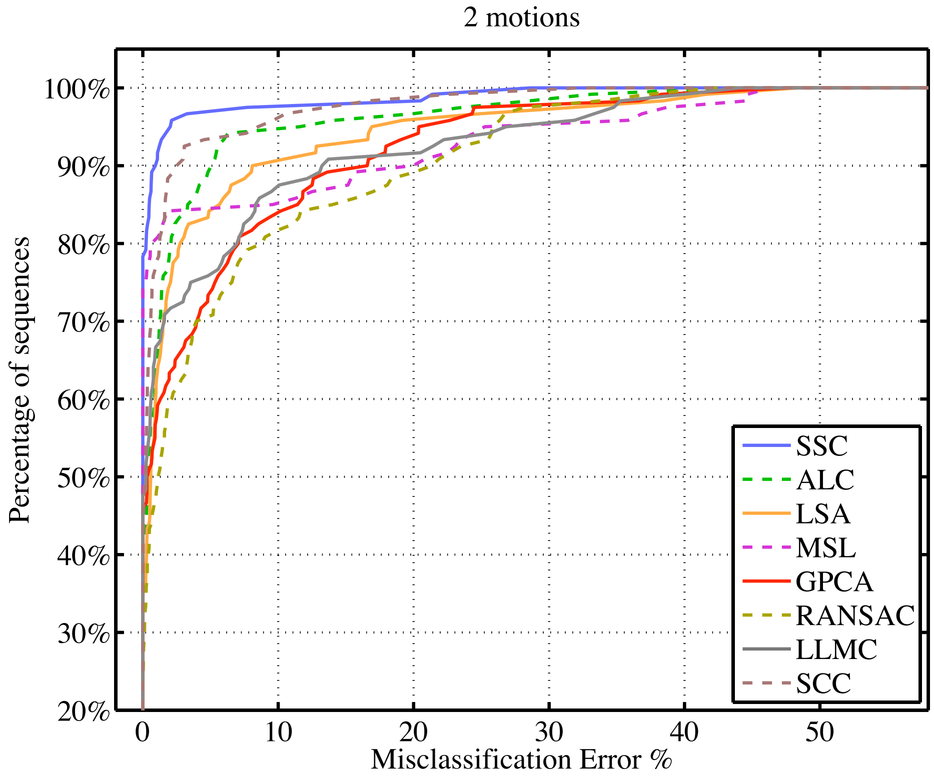
		GPCA	LLMC	LSA	RANSAC	MSL	SCC	ALC	SSC-B	SSC-N
78	*Checkerboard*	6.09	3.96	2.57	6.52	4.46	1.30	1.55	**0.83**	1.12
31	*Traffic*	1.41	3.53	5.43	2.55	2.23	1.07	1.59	0.23	**0.02**
11	*Articulated*	2.88	6.48	4.10	7.25	7.23	3.68	10.70	1.63	**0.62**
	All	4.59	4.08	3.45	5.56	4.14	1.46	2.40	**0.75**	0.82
	Time	0.32s		7.58s	0.18s	11h 4m			65s	

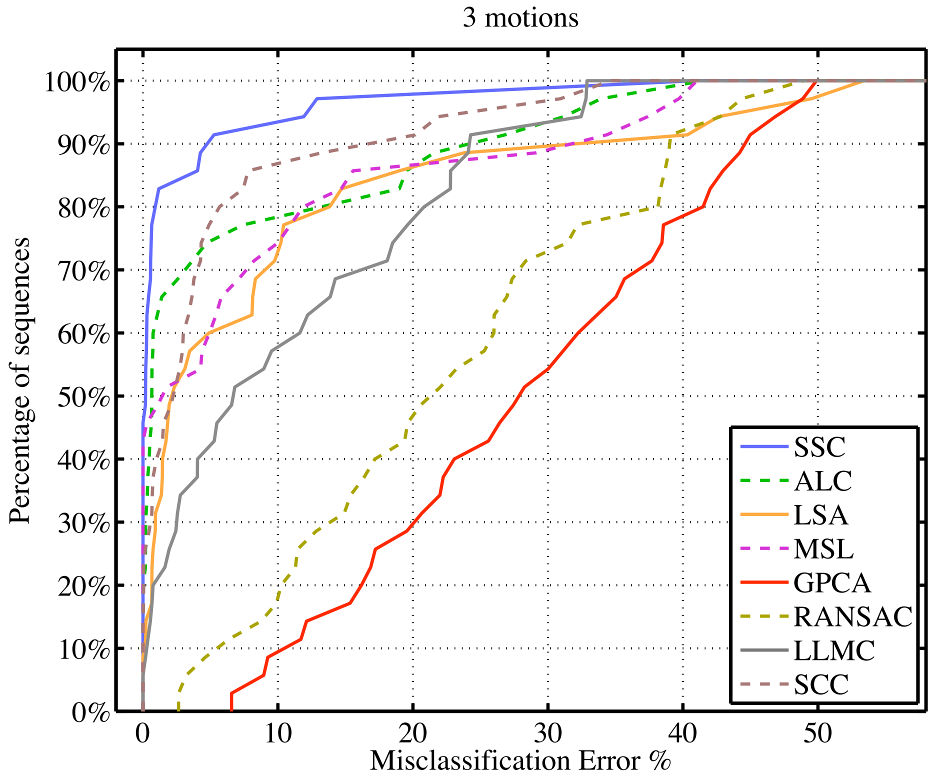
- 3 motions, 35 sequences, 398 points, 29 frames

	GPCA	LLMC	LSA	RANSAC	MSL	SCC	ALC	SSC-B	SSC-N
Checkerboard	31.95	8.48	5.80	25.78	10.38	5.68	5.20	4.49	**2.97**
Traffic	19.83	6.04	25.07	12.83	1.80	2.35	7.75	0.61	**0.58**
Articulated	16.85	9.38	7.25	21.38	2.71	10.94	21.08	1.60	**1.42**
All	28.66	8.04	9.73	22.94	8.23	5.31	6.69	3.55	**2.45**

Hopkins 155 database

- Misclassification rates for 2 and 3 motions





Results with missing entries & outliers

- Misclassifications rates on 12 motion sequences with missing data

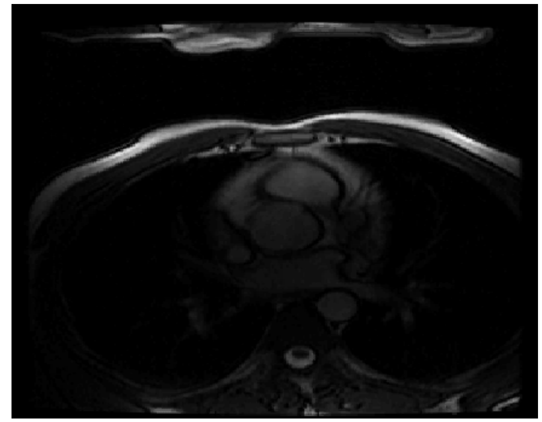
Method	PF+ ALC_5	PF+ALC_{sp}	ℓ^1+ALC_5	ℓ^1+ALC_{sp}	SSC-N
Average	1.89%	10.81%	3.81%	1.28%	**0.13%**
Median	0.39%	7.85%	0.17%	1.07%	0.00%

- Misclassifications rates on 12 motion sequences with corrupted data

Method	$\ell^1 + ALC_5$	$\ell^1 + ALC_{sp}$	SSC-N
Average	4.15%	3.02%	**1.05%**
Median	0.21%	0.89%	0.43%

Modeling dynamic textures

- Examples of dynamic textures:



- Model temporal evolution as the output of a linear dynamical system (LDS): Soatto et al. '01

dynamics

$$z_{t+1} = Az_t + v_t$$

images

$$y_t = Cz_t + w_t$$

appearance

Segmentation of dynamic textures

- Model intensity at each pixel as the output of an AR model

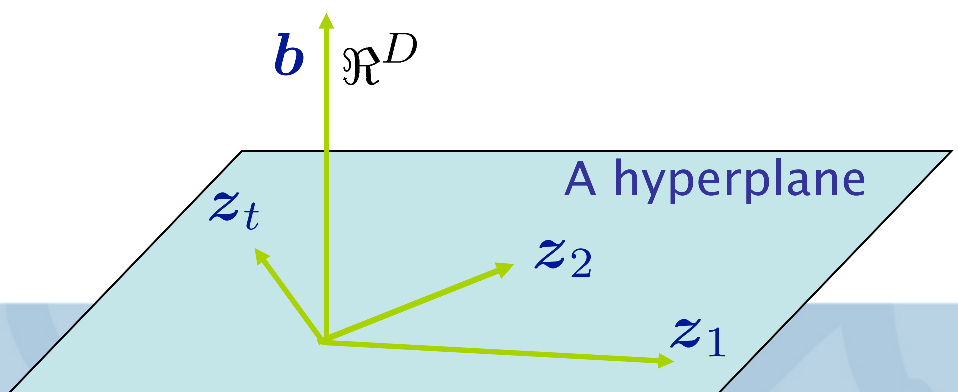
$$y_t(\boldsymbol{x}) = \sum_{j=1}^{n} a_j y_{t-j}(\boldsymbol{x}) + w_t(\boldsymbol{x})$$

- Define regressors & parameters

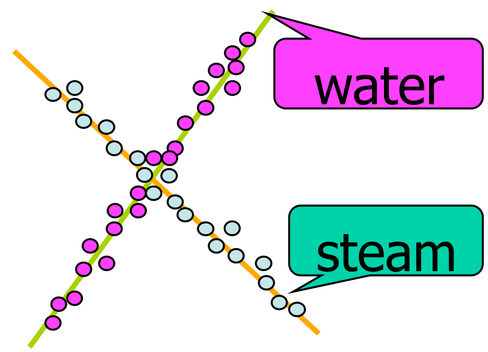
$$\boldsymbol{z}_t(\boldsymbol{x}) = [y_t(\boldsymbol{x}),\ y_{t-1}(\boldsymbol{x}), \cdots, y_{t-n}(\boldsymbol{x})]^\top$$

$$\boldsymbol{b} = [1,\ -a_1,\ -a_2, \cdots, -a_n]^\top$$

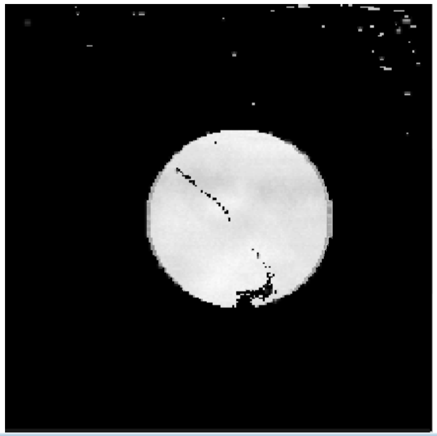
- Regressors with same texture live in a hyperplane



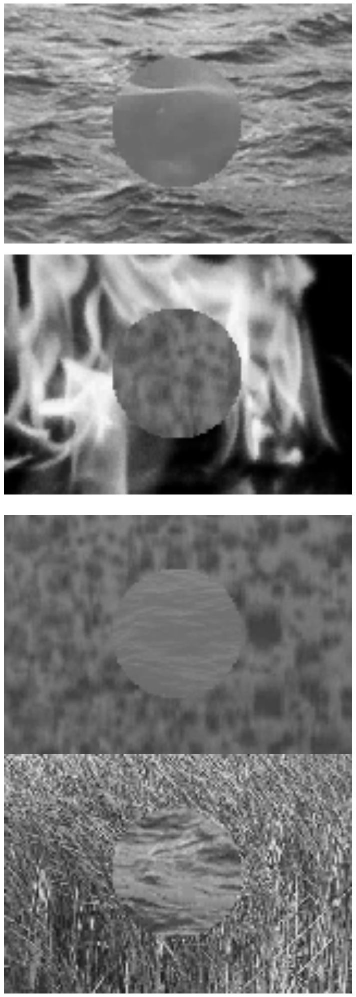
- Multiple dynamic textures live in multiple hyperplanes

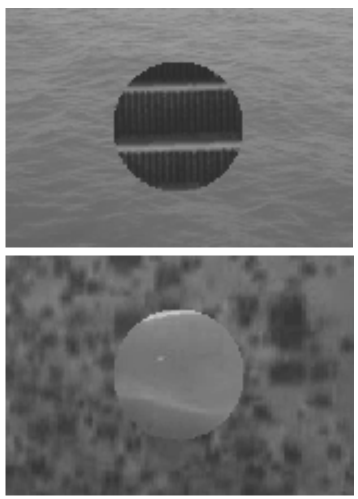


- Can cluster the regressors using GPCA



Segmentation of dynamic textures

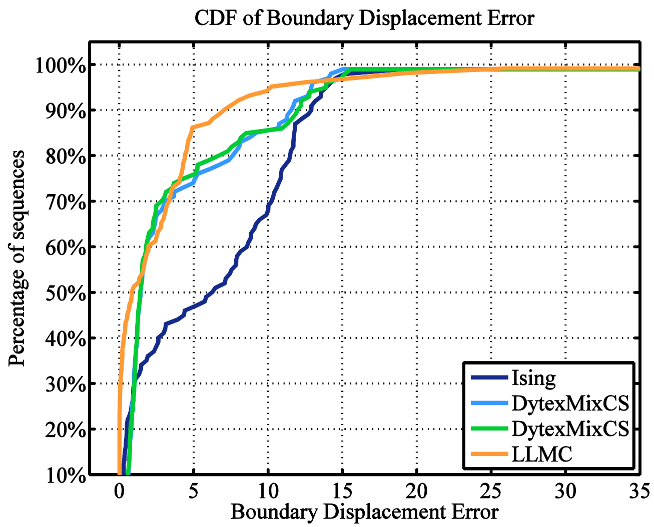


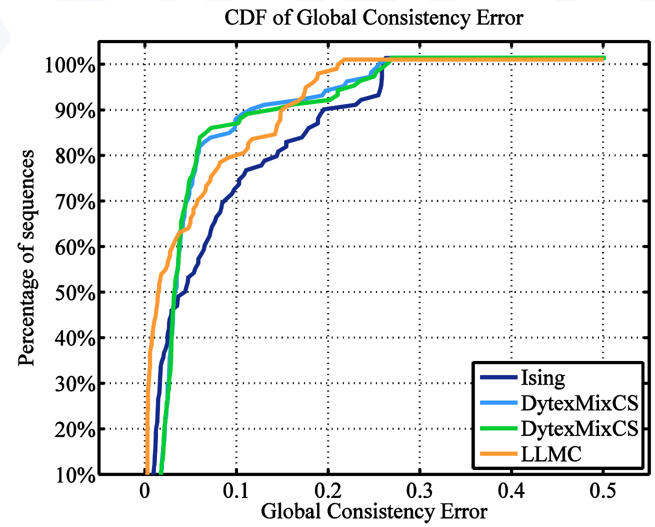


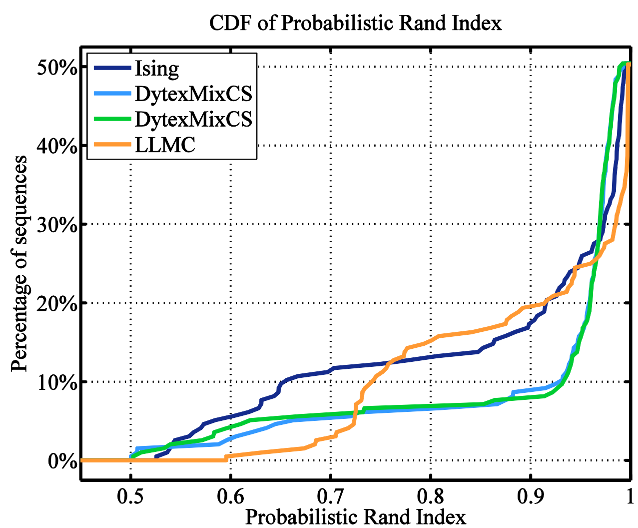
Boundary Displacement Error: ranges between $[0, \infty)$ *with* 0 *being the best*
Global Consistency Error: ranges between $[0, 1]$ *with* 0 *being the best*
Probabilistic Rand Index: ranges between $[0, 1]$ *with* 1 *being the best*
Variation of Information: ranges between $[0, \infty)$ *with* 0 *being the best*

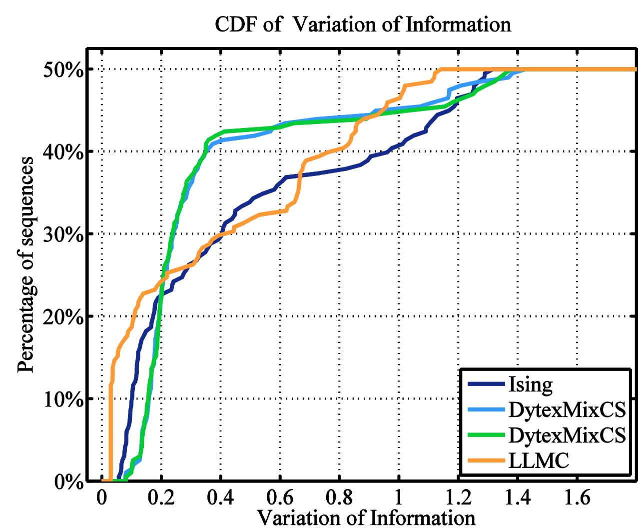
	BDE	GCE
DytexMixIC	3.46	0.05
DytexMixCS	3.52	0.06
Level Sets using Ising	6.06	0.08
LLMC	2.64	0.05
	PRI	VOI
DytexMixIC	0.92	0.35
DytexMixCS	0.92	0.35
Level Sets using Ising	0.88	0.45
LLMC	0.89	0.38

Segmentation of dynamic textures





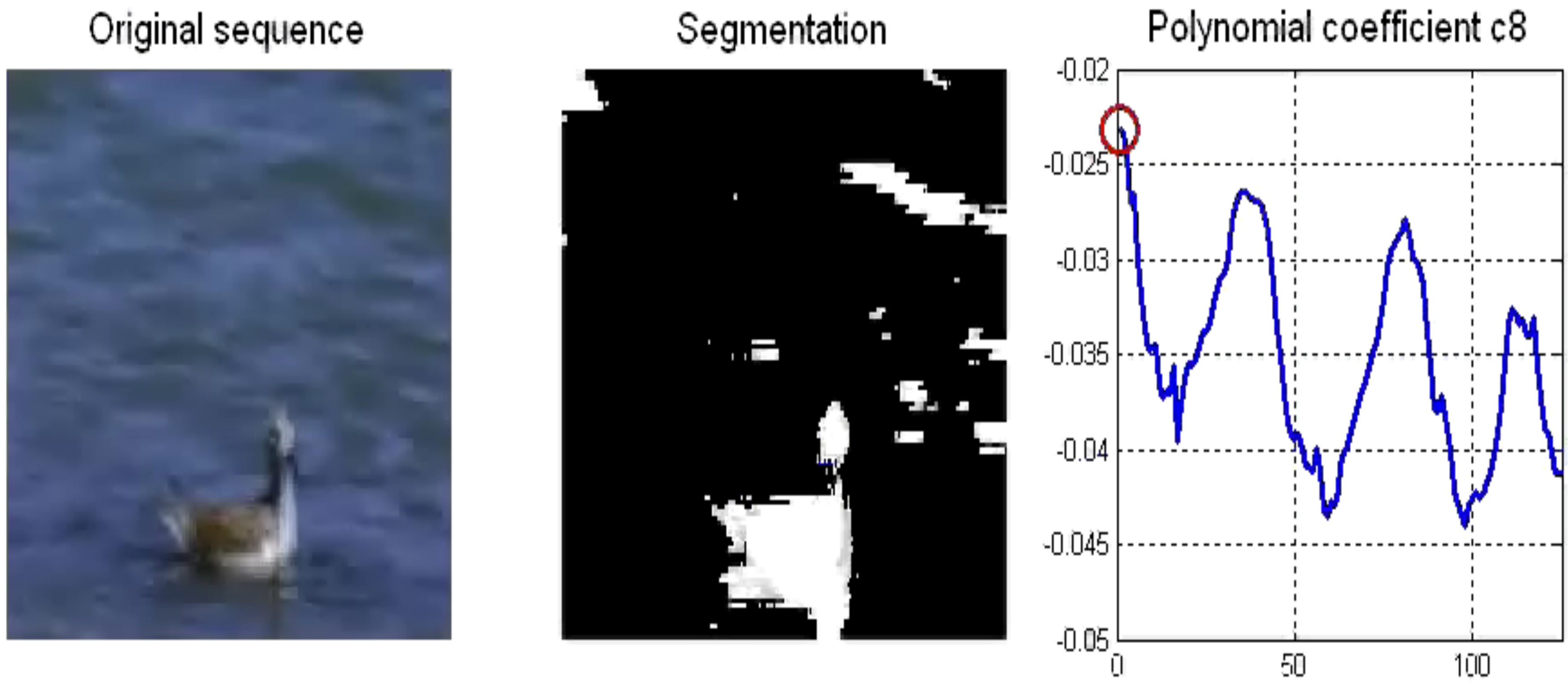




Boundary Displacement Error: ranges between $[0, \infty)$ *with* 0 *being the best*
Global Consistency Error: ranges between $[0, 1]$ *with* 0 *being the best*
Probabilistic Rand Index: ranges between $[0, 1]$ *with* 1 *being the best*
Variation of Information: ranges between $[0, \infty)$ *with* 0 *being the best*

Center for IMAGING SCIENCE

Segmentation of dynamic textures



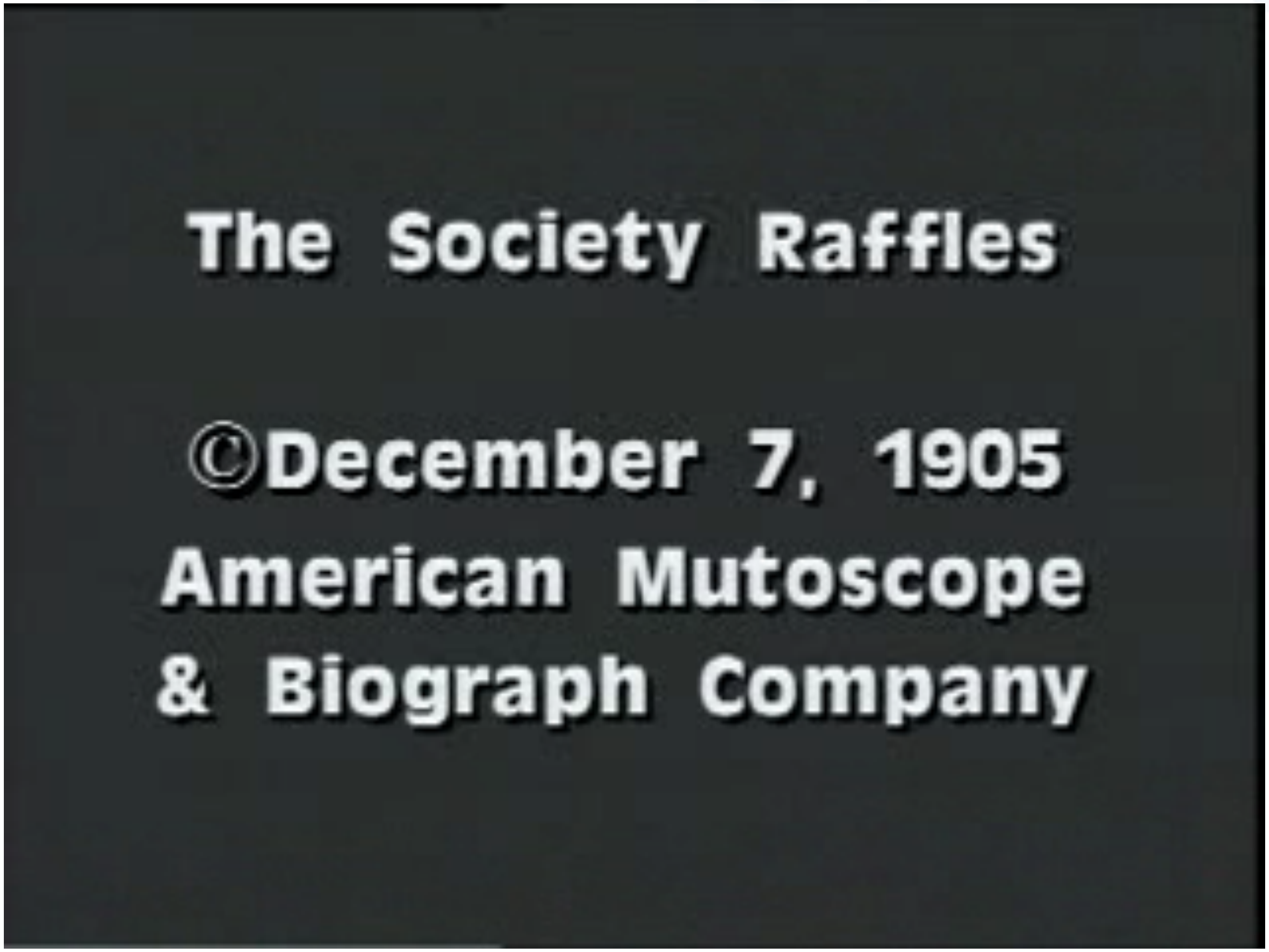
Variational segmentation of dynamic textures

- Moving boundary segmentation results

Raccoon on River

Ocean-fire

Temporal video segmentation



Temporal video segmentation by GPCA

- Empty living room
- Middle-aged man enters
- Woman enters
- Young man enters, introduces the woman and leaves
- Middle-aged man flirts with woman and steals her tiara
- Middle-aged man checks the time, rises and leaves
- Woman walks him to the door
- Woman returns to her seat
- Woman misses her tiara
- Woman searches her tiara
- Woman sits and dismays

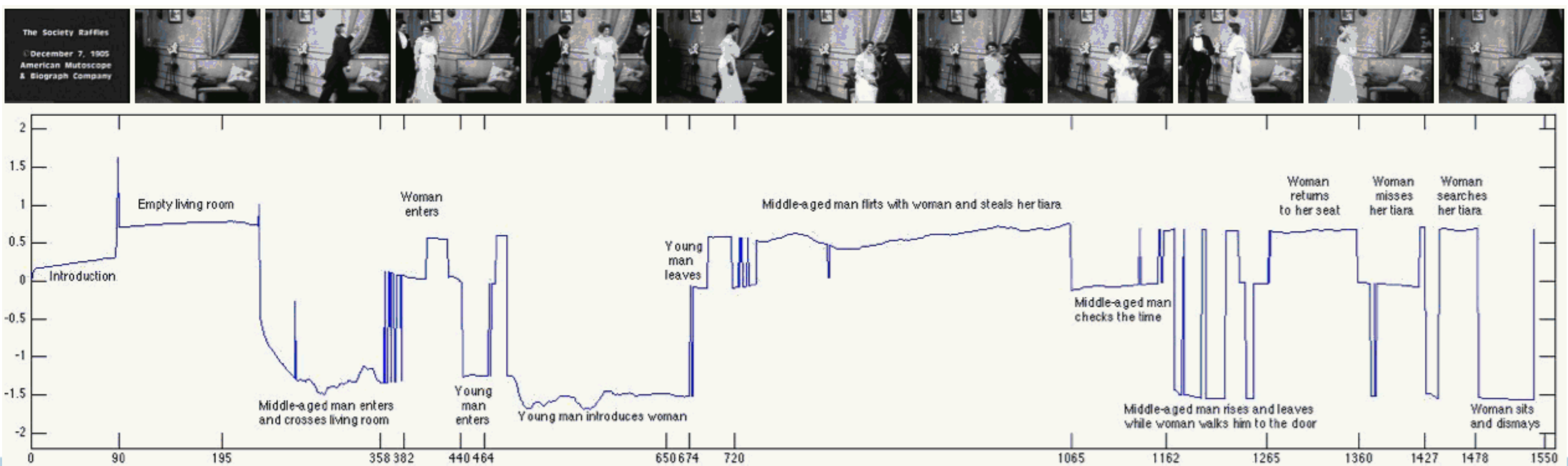
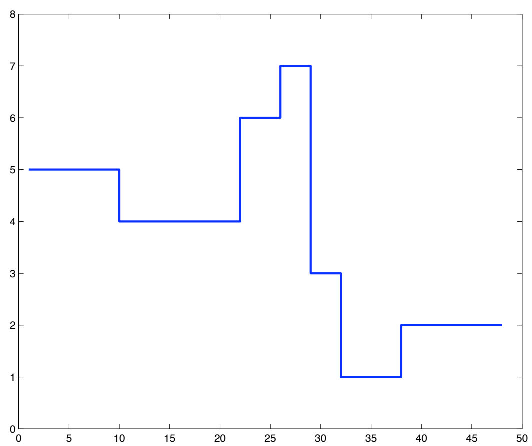
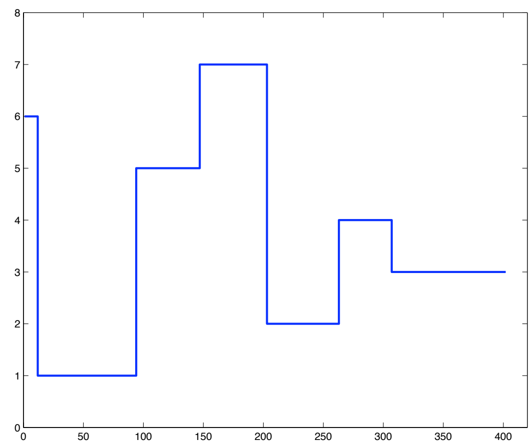
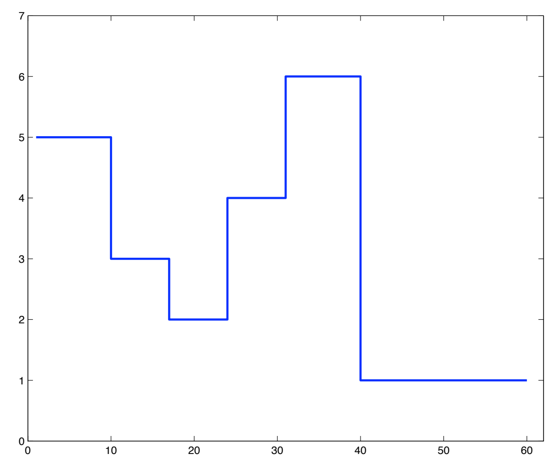


Fig. 5. Temporal segmentation of a scene from the movie *The society raffles*. The top row shows several key frames from the scene displaying different events. The bottom row shows the temporal evolution of the parameter $\hat{c}_t$ as a function of time.

Temporal video segmentation by SSC







Center for IMAGING SCIENCE

Conclusions

- Many computer vision problems can be posed as subspace clustering problems
  - 2-D and 3-D motion segmentation
  - Dynamic texture segmentation
  - Temporal video segmentation
- These problems can be solved using
  - Generalized Principal Component Analysis (GPCA): algebraic method based on polynomial fitting and differentiation
  - Sparse Subspace Clustering (SSC): algorithm based on sparse representation theory and spectral clustering
- Future work
  - Extending SCC to disjoint subspaces: what are the conditions on the subspace angles that allow for a sparse recovery?
  - Extending SCC to nonlinear manifolds

References: Springer-Verlag 2010

Generalized Principal Component Analysis

Estimation & Segmentation of Geometric Models

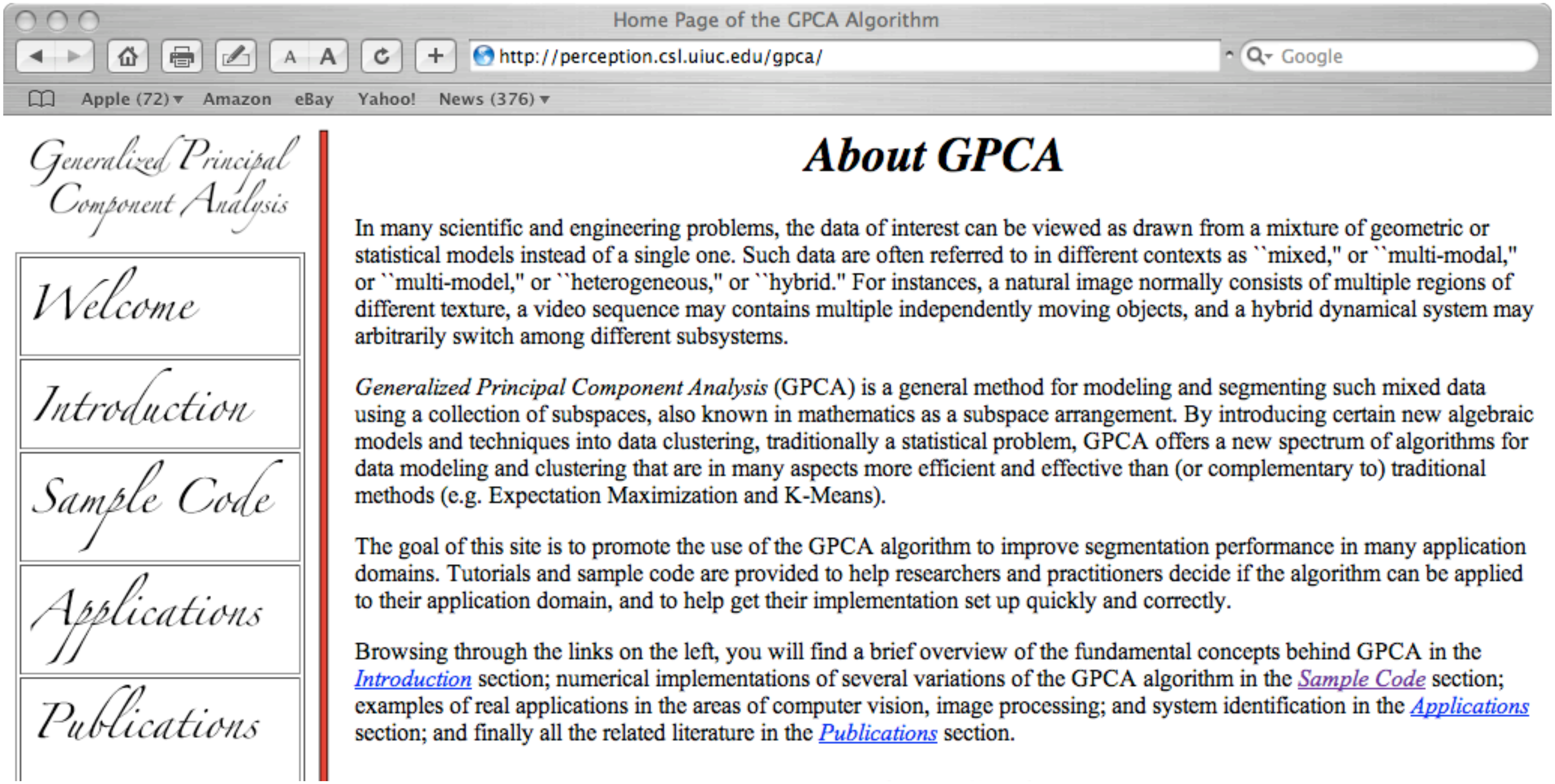
René Vidal (Biomedical Engineering, Johns Hopkins University)

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

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

Slides, MATLAB code, papers

http://perception.csl.uiuc.edu/gpca



About GPCA

In many scientific and engineering problems, the data of interest can be viewed as drawn from a mixture of geometric or statistical models instead of a single one. Such data are often referred to in different contexts as ``mixed," or ``multi-modal," or ``multi-model," or ``heterogeneous," or ``hybrid." For instances, a natural image normally consists of multiple regions of different texture, a video sequence may contains multiple independently moving objects, and a hybrid dynamical system may arbitrarily switch among different subsystems.

Generalized Principal Component Analysis (GPCA) is a general method for modeling and segmenting such mixed data using a collection of subspaces, also known in mathematics as a subspace arrangement. By introducing certain new algebraic models and techniques into data clustering, traditionally a statistical problem, GPCA offers a new spectrum of algorithms for data modeling and clustering that are in many aspects more efficient and effective than (or complementary to) traditional methods (e.g. Expectation Maximization and K-Means).

The goal of this site is to promote the use of the GPCA algorithm to improve segmentation performance in many application domains. Tutorials and sample code are provided to help researchers and practitioners decide if the algorithm can be applied to their application domain, and to help get their implementation set up quickly and correctly.

Browsing through the links on the left, you will find a brief overview of the fundamental concepts behind GPCA in the *Introduction* section; numerical implementations of several variations of the GPCA algorithm in the *Sample Code* section; examples of real applications in the areas of computer vision, image processing; and system identification in the *Applications* section; and finally all the related literature in the *Publications* section.

Center for IMAGING SCIENCE

For more information,

Vision Lab @ Johns Hopkins University

http://www.vision.jhu.edu

Thank You!

- Ehsan Elhamifar, JHU
- Roberto Tron, JHU
- Shankar Rao, UIUC
- Yi Ma, Microsoft Research China