JHU vision lab

Part II
Applications of GPCA in Computer Vision

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

Part II: Applications in computer vision

- Applications to motion & video segmentation (10.30-11.20)
  - 2-D and 3-D motion segmentation
  - Temporal video segmentation
  - Dynamic texture segmentation

- Applications to image representation and segmentation (11.20-12.10)
  - Multi-scale hybrid linear models for sparse image representation
  - Hybrid linear models for image segmentation

JHU vision lab

Applications to motion and and video segmentation

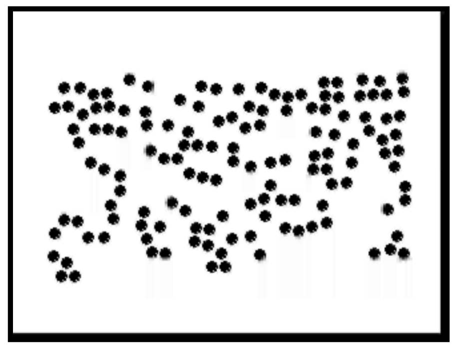
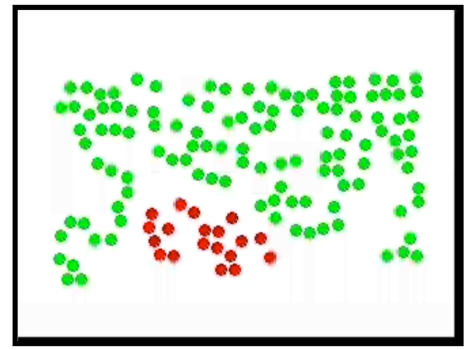
René Vidal
Center for Imaging Science
Institute for Computational Medicine
Johns Hopkins University

THE DEPARTMENT OF BIOMEDICAL ENGINEERING
The Whitaker Institute at Johns Hopkins

Center for IMAGING SCIENCE

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

- Mathematics of the problem depends on
  - Number of frames (2, 3, multiple)
  - Projection model (affine, perspective)
  - Motion model (affine, translational, homography, fundamental matrix, etc.)
  - 3-D structure (planar or not)

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
  - Variational approaches: Cremers-Soatto ICCV'03
  - Initialization: Wang-Adelson'94, Irani-Peleg'92, Shi-Malik'98, Vidal-Singaraju'05-'06
- Multiple rigid motions in two perspective views
  - Probabilistic approaches: Feng-Perona'98, Torr'98
  - Particular cases: Izawa-Mase'92, Shashua-Levin'01, Sturm'02,
  - Multibody fundamental matrix: Wolf-Shashua CVPR'01, Vidal et al. ECCV'02, CVPR'03, IJCV'06
  - Motions of different types: Vidal-Ma-ECCV'04, Rao-Ma-ICCV'05
- 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, Wu et al.'01, Kanatani' et al.'01-02-04
  - Algebraic: Yan-Pollefeys-ECCV'06, Vidal-Hartley-CVPR'04
- Multiple rigid motions in multiple perspective views
  - Schindler et al. ECCV'06, Li et al. CVPR'07

A unified approach to motion segmentation

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

$$
\begin{tikzcd}
 & \boxed{x_2} \\
\boxed{x_1} \arrow[ur, "\mathcal{M}_1"] \arrow[dr, "\mathcal{M}_2"'] & \\
 & \boxed{x_2}
\end{tikzcd}
\qquad
\begin{array}{c}
f(x_1, x_2, \mathcal{M}_1) = 0 \\
\boxed{\text{or}} \longrightarrow \boxed{\text{chicken-and-egg}} \\
f(x_1, x_2, \mathcal{M}_2) = 0
\end{array}
$$

  - Eliminate feature clustering: multiplication

$$f(x_1, x_2, \mathcal{M}_1) f(x_1, x_2, \mathcal{M}_2) = 0$$

  - Estimate a single multibody motion model: polynomial fitting

$$f(x_1, x_2, \mathcal{M}_1) f(x_1, x_2, \mathcal{M}_2) = g(x_1, x_2, \mathcal{M}) = 0$$

  - Segment multibody motion model: polynomial differentiation

$$\mathcal{M} \mapsto \{\mathcal{M}_i\}_{i=1}^{n} \qquad \mathcal{M}_i = Dg|_{x_1, x_2}$$

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

Segmentation of 3-D translational motions

- Multiple epipoles (translation)

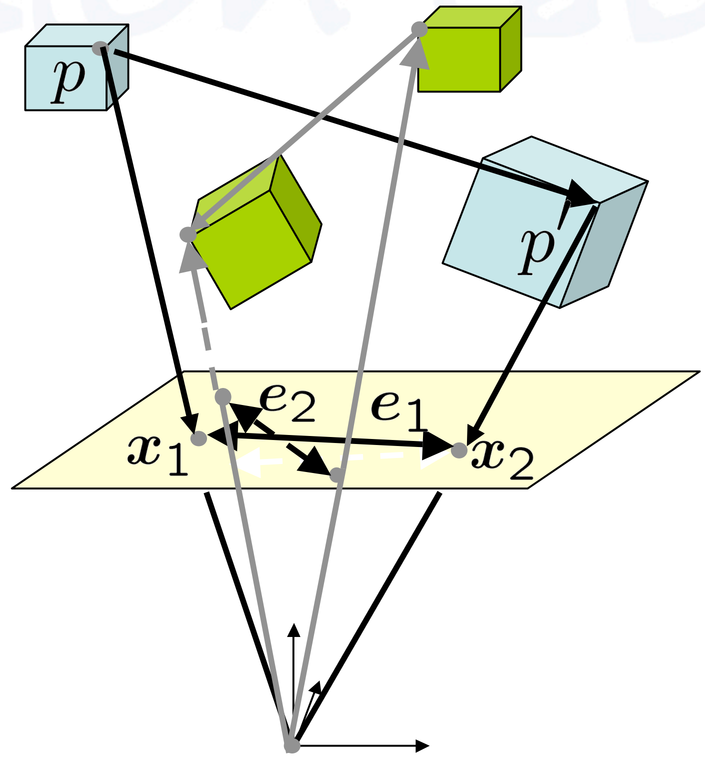
$$\{\boldsymbol{e}_i \in \mathbb{R}^3\}_{i=1}^n$$

- 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

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

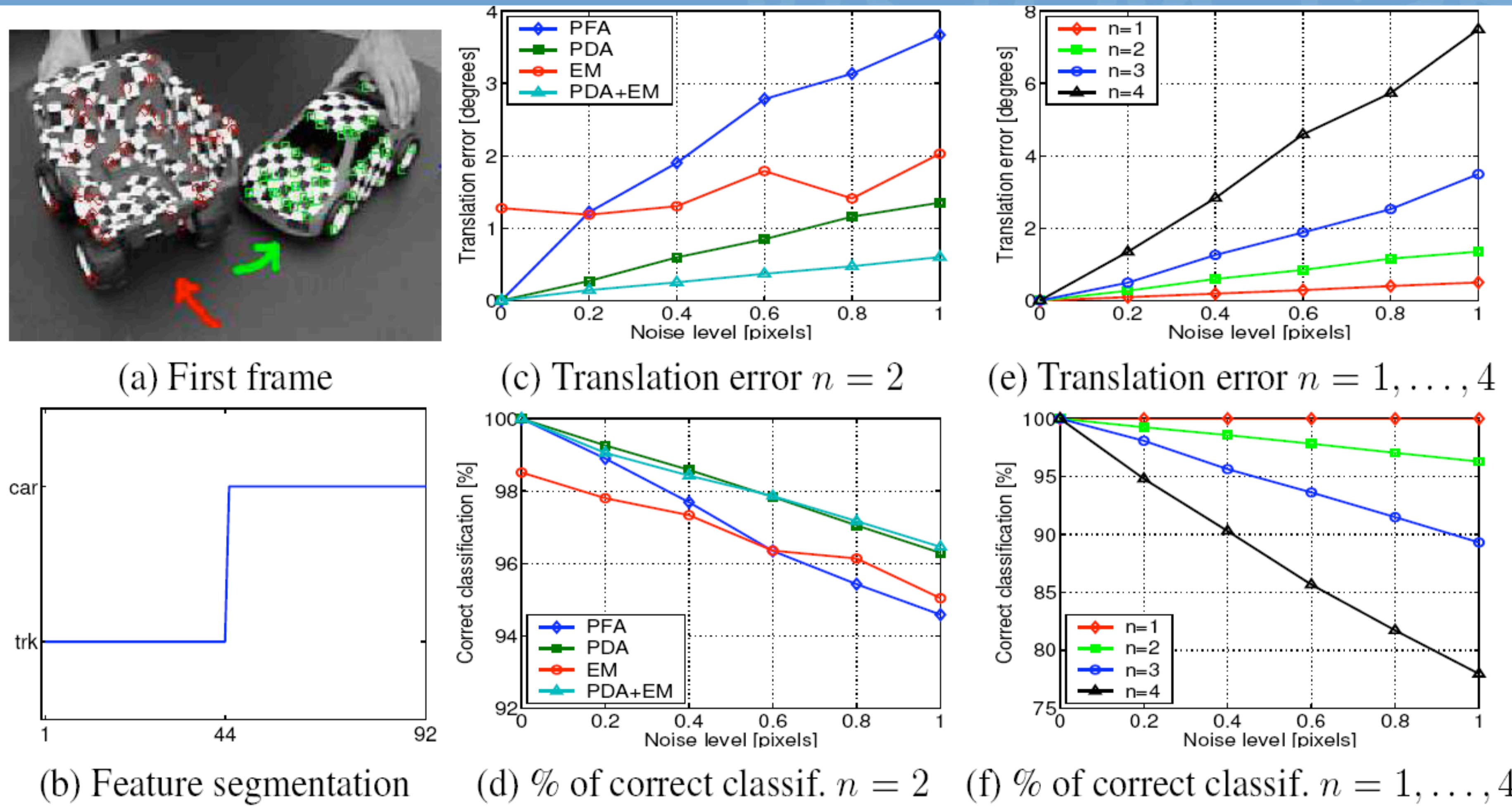


- Epipoles are derivatives of $p_n(\boldsymbol{\ell})$ at epipolar lines

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

Center for IMAGING SCIENCE

Segmentation of 3-D translational motions



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

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

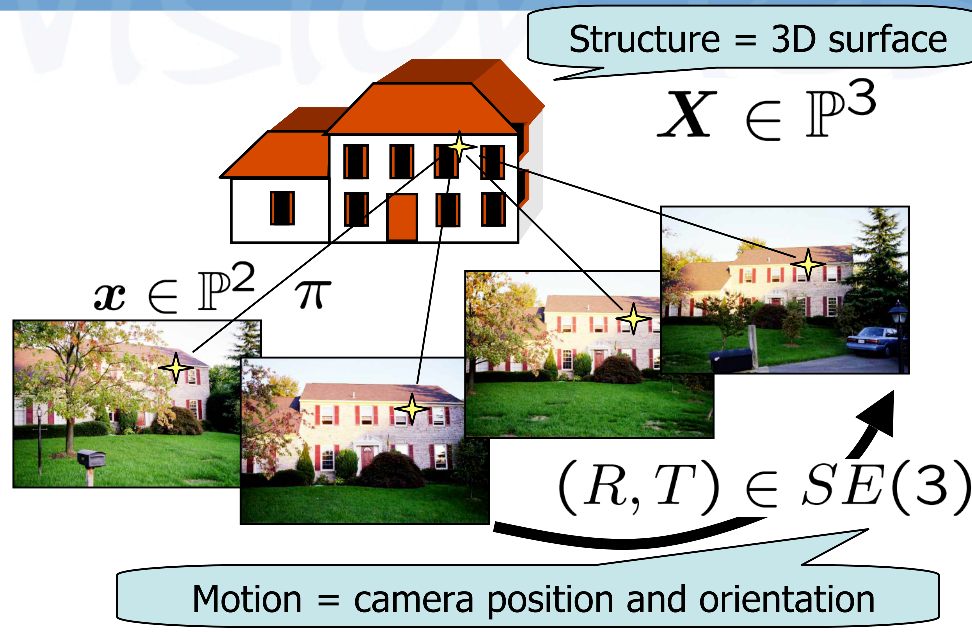
Fig. 3. Segmenting 3-D translational motions by clustering planes in $\mathbb{R}^3$. Left: segmenting a real sequence with 2 moving objects. Center: comparing our algorithm with PFA and EM as a function of noise in the image features. Right: performance of PFA as a function of the number of motions.

Single-body factorization

- Affine camera model

$$\boxed{\begin{aligned} \boldsymbol{x}_{fp} &= \begin{bmatrix} R_f & T_f \end{bmatrix} \boldsymbol{X}_p \\ &= \mathtt{A}_f \boldsymbol{X}_p \end{aligned}}$$

  - 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

$$\begin{array}{ccccc} \mathtt{W} & = & \mathtt{M} & & \mathtt{S}^T \\ \underbrace{\begin{bmatrix} \boldsymbol{x}_{11} \cdots \boldsymbol{x}_{1P} \\ \vdots \qquad \vdots \\ \boldsymbol{x}_{F1} \cdots \boldsymbol{x}_{FP} \end{bmatrix}}_{2F \times P} & = & \underbrace{\begin{bmatrix} \mathtt{A}_1 \\ \vdots \\ \mathtt{A}_F \end{bmatrix}}_{2F \times 4} & & \underbrace{\begin{bmatrix} \boldsymbol{X}_1 \cdots \boldsymbol{X}_P \end{bmatrix}}_{4 \times P} \end{array}$$

Multi-body factorization

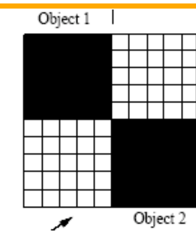
- Given n rigid motions

$$\mathtt{W}_i = \mathtt{M}_i \mathtt{S}_i^T \qquad \mathtt{M}_i \in \mathbb{R}^{2F\times 4}$$
$$i = 1, \dots, n \quad \mathtt{S}_i \in \mathbb{R}^{P_i \times 4}$$

$$\begin{aligned}\mathtt{W} &= \begin{bmatrix}\mathtt{W}_1 \cdots \mathtt{W}_n\end{bmatrix} \\ &= \begin{bmatrix}\mathtt{M}_1 \cdots \mathtt{M}_n\end{bmatrix} \begin{bmatrix} \mathtt{S}_1^T & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathtt{S}_n^T \end{bmatrix}\end{aligned}$$

- Motion segmentation is obtained from
  - Leading singular vector of $\mathtt{W}$ (Boult and Brown '91)
  - Shape interaction matrix $\mathtt{Q}$ (Costeira & Kanade '95, Gear '94)

$$\mathtt{Q} = \mathtt{V}\mathtt{V}^T \qquad \mathtt{Q}_{ij} = 0 \text{ if } i \text{ and } j$$
$$\mathtt{W} = \mathtt{U}\mathtt{S}\mathtt{V}^T \qquad \text{belong to different motions}$$



  - Number of motions (if fully-dimensional)

$$n = \frac{1}{4}\text{rank}(\mathtt{W})$$

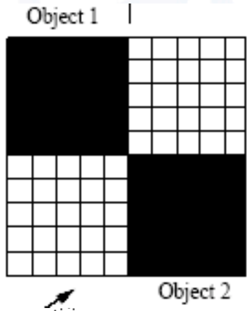
- Motion subspaces need to be independent (Kanatani '01)

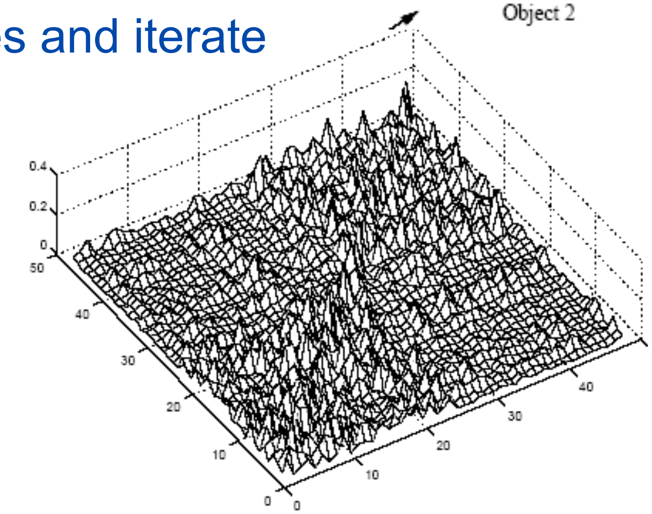
$$\text{rank}([\mathtt{W}_i\ \mathtt{W}_j]) = \text{rank}(\mathtt{W}_i) + \text{rank}(\mathtt{W}_j)$$

Multi-body factorization

- Sensitive to noise

  $Q_{ij} = 0$ if i and j belong to different motions
  - Kanatani (ICCV '01): use model selection to scale Q
  - Wu et al. (CVPR'01): project data onto subspaces and iterate
- Fails with partially dependent motions
  - Zelnik-Manor and Irani (CVPR'03)
    - Build similarity matrix from normalized Q
    - Apply spectral clustering to similarity matrix
  - Yan and Pollefeys (ECCV'06)
    - Local subspace estimation + spectral clustering
  - Kanatani (ECCV'04)
    - Assume degeneracy is known: pure translation in the image
    - Segment data by multi-stage optimization (multiple EM problems)
- Cannot handle missing data
  - Gruber and Weiss (CVPR'04)
    - Expectation Maximization



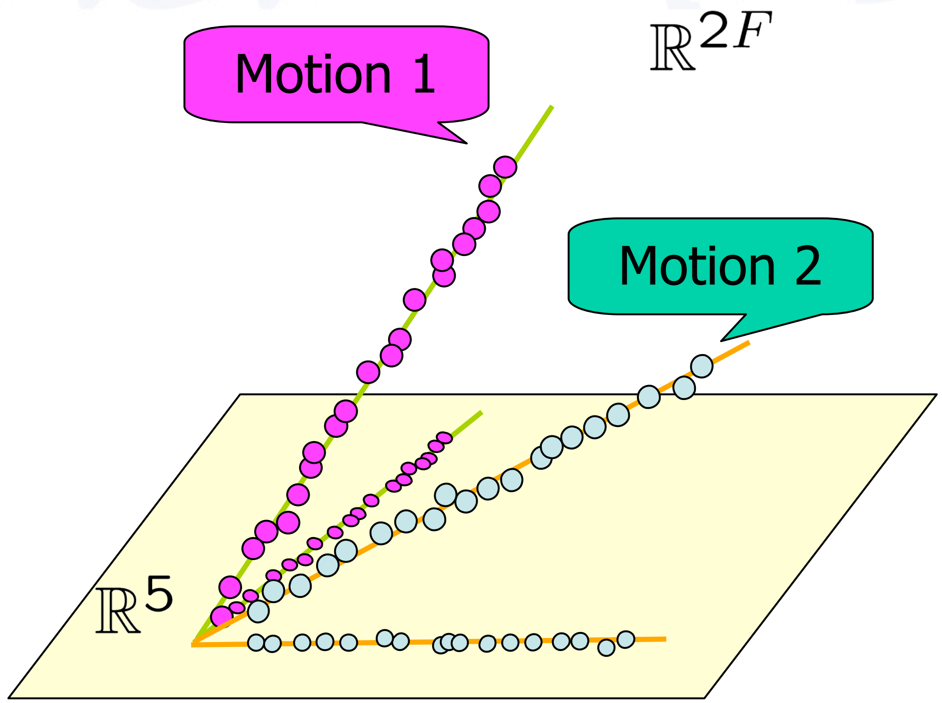


PowerFactorization+GPCA

- A motion segmentation algorithm that
  - Is provably correct with perfect data
  - Handles both independent and degenerate motions
  - Handles both complete and incomplete data
- Project trajectories onto a 5-D subspace of $\mathbb{R}^{2F}$
  - Complete data: PCA or SVD
  - Incomplete data: PowerFactorization
- Cluster projected subspaces using GPCA
  - Handles both independent and degenerate motions
  - Non-iterative: can be used to initialize EM

Projection onto a 5-D subspace

- Motion of one rigid-body lives in 4-D subspace of $\mathbb{R}^{2F}$
- By projecting onto a 5-D subspace of $\mathbb{R}^{2F}$
  - Number and dimensions of subspaces are preserved
  - Motion segmentation is equivalent to clustering subspaces of dimension 2, 3 or 4 in $\mathbb{R}^5$
  - Minimum #frames = 3 (CK needs a minimum of 2n frames for n motions)
  - Can remove outliers by robustly fitting the 5-D subspace using Robust SVD (DeLaTorre-Black)



- What projection to use?
  - PCA: 5 principal components
  - RPCA: with outliers

Projection onto a 5-D subspace

PowerFactorization algorithm:

Given $\mathtt{W}$, factor it as $\mathtt{W} = \mathtt{AB}^T$

- Complete data

$$\min_{\mathtt{A},\mathtt{B}} \sum_{(i,j)} (\mathtt{W}_{ij} - (\mathtt{AB}^T)_{ij})^2$$

  - Given A solve for B

$$\boxed{\mathtt{B}_k = \mathtt{W}^T \mathtt{A}_{k-1}}$$

  - Orthonormalize B
  - Given B solve for A

$$\boxed{\mathtt{A}_k = \mathtt{W}\mathtt{B}_k}$$

  - Iterate
- Converges to rank-r approximation with rate

$$(s_{r+1}/s_r)^k$$

- Incomplete data

$$\min_{\mathtt{A},\mathtt{B}} \sum_{(i,j)\in\mathcal{I}} (\mathtt{W}_{ij} - (\mathtt{AB}^T)_{ij})^2$$

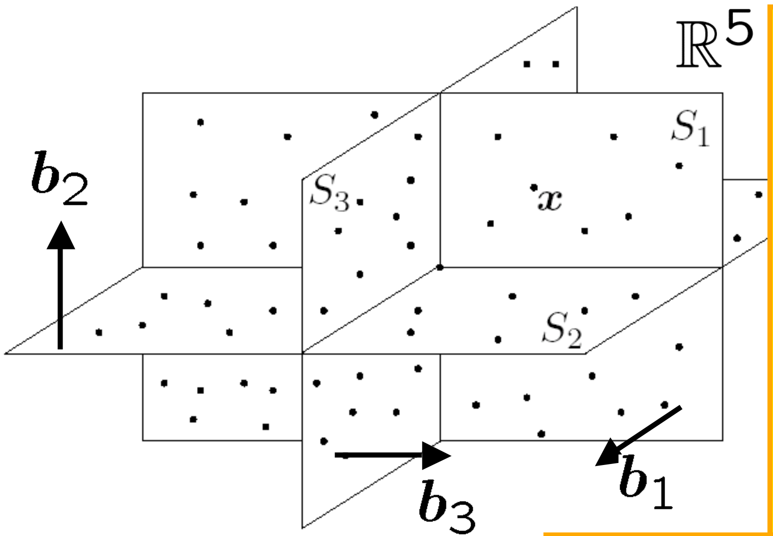
$$\mathcal{I} = (i,j) : \mathtt{W}_{ij} \text{ is known}$$

Linear problem

- It diverges in some cases
- Works well with up to 30% of missing data

Motion segmentation using GPCA

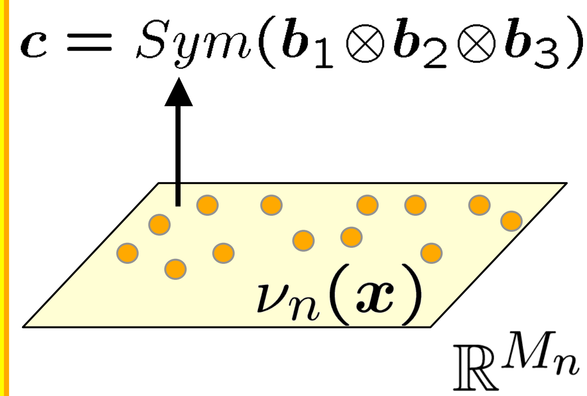
- Apply polynomial embedding to 5-D points



Veronese map

$$\nu_n : \mathbb{R}^5 \to \mathbb{R}^{M_n}$$

$$\nu_2 \begin{pmatrix} x_1 \\ \vdots \\ x_5 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_1 x_2 \\ \vdots \\ x_5^2 \end{pmatrix}$$



$$M_n = \frac{(n+1)(n+2)(n+3)(n+4)}{24}$$

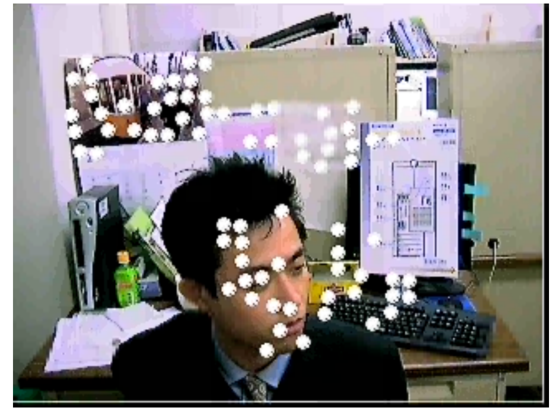
Minimum #points	1	2	3	4	n
	4	14	34	69	$O(n^4)$

Experimental results: Kanatani sequences

- Sequence A

Sequence B

Sequence C

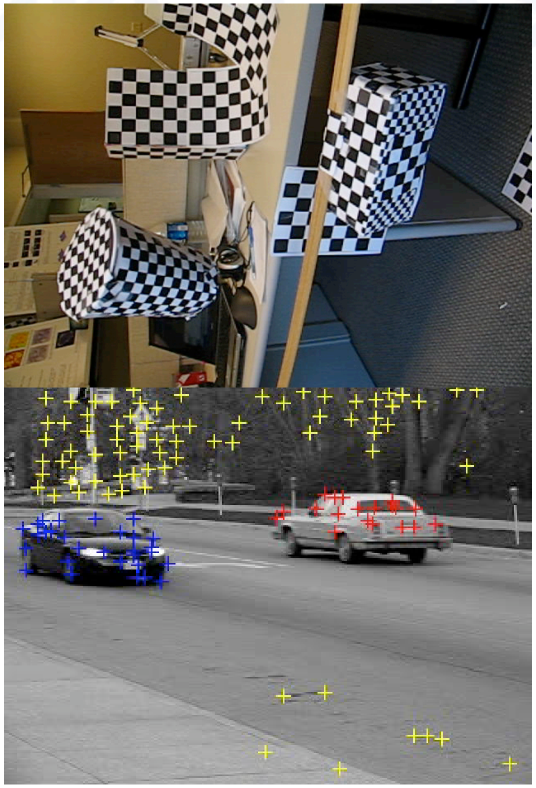


- Percentage of correct classification

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%**
PowerFactorization + GPCA	**100%**	**100%**	**100%**

Experimental results: Hopkins 155 database

- Collected 155 sequences
  - 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



Experimental results: Hopkins 155 database

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

Algorithm	error	time
GPCA	4.59%	0.03 s
LSA-5	6.73%	6.75 s
LSA-8	3.45%	7.58 s
MSL	4.14%	11.7 h
RANSAC	5.56%	0.02s

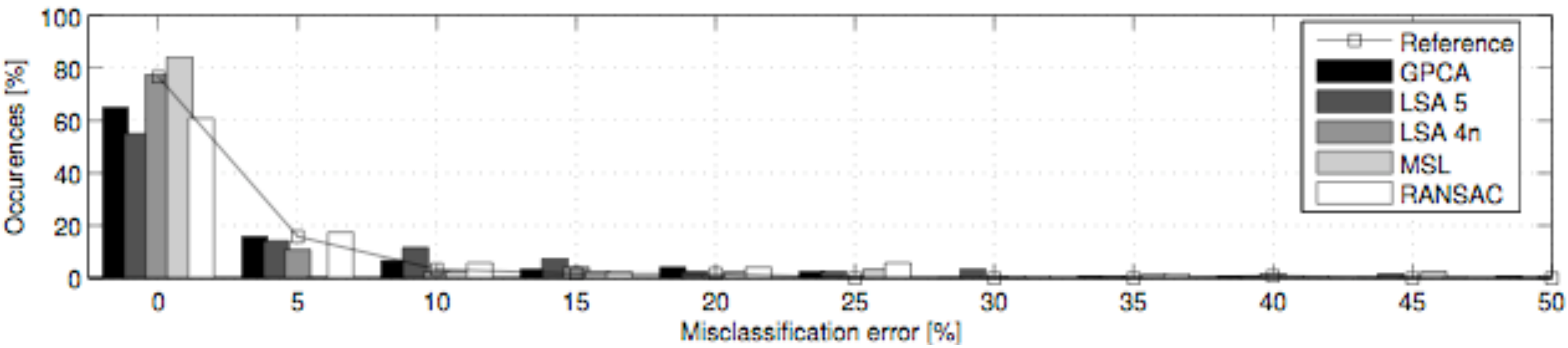
- 3 motions, 408 points, 27 frames

Algorithm	error	time
Kanatani	7.67%	1d 4 h
GPCA	23.8%	0.05 s
Spectral GPCA	26.6%	0.63 s

Experimental results: Hopkins 155 database

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

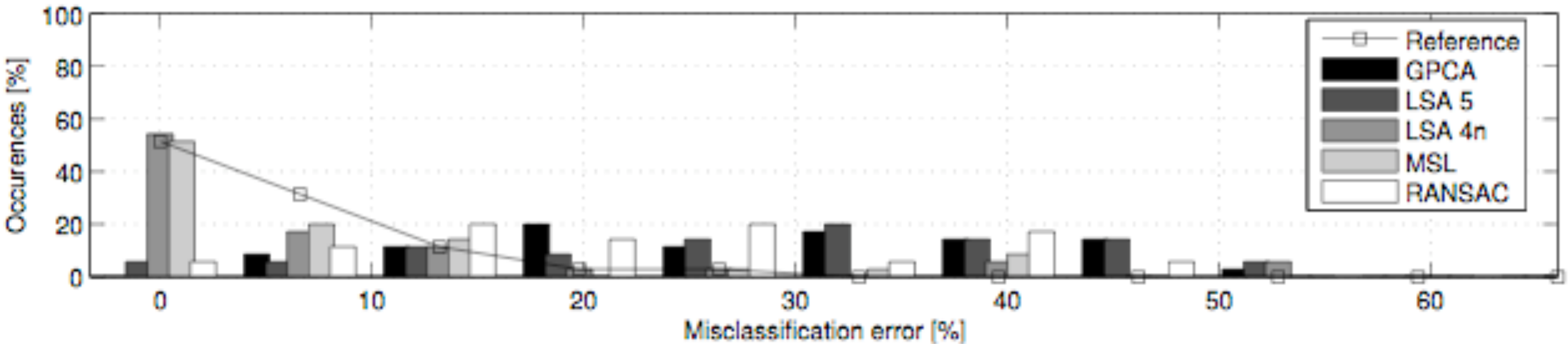
	REF	GPCA	LSA 5	LSA $4n$	MSL	RANSAC
Checkerboard	2.76%	6.09%	8.84%	2.57%	4.46%	6.52%
Traffic	0.30%	1.41%	2.15%	5.43%	2.23%	2.55%
Articulated	1.71%	2.88%	4.66%	4.10%	7.23%	7.25%
	REF	GPCA	LSA 5	LSA $4n$	MSL	RANSAC
Average	2.03%	4.59%	6.73%	3.45%	4.14%	5.56%
Time		0.32 s	6.75 s	7.58 s	11 h 4 m	0.18 s



Experimental results: Hopkins 155 database

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

	REF	GPCA	LSA 5	LSA $4n$	MSL	RANSAC
Checkerboard	6.28%	31.95%	30.37%	5.80%	10.38%	25.78%
Traffic	1.30%	19.83%	27.02%	25.07%	1.80%	12.83%
Articulated	2.66%	16.85%	23.11%	7.25%	2.71%	21.38%
	REF	GPCA	LSA 5	LSA $4n$	MSL	RANSAC
Average	5.08%	28.66%	29.28%	9.73%	8.23%	22.94%
Time		0.74 s	15.01 s	15.95 s	1 d 23 h	0.25 s



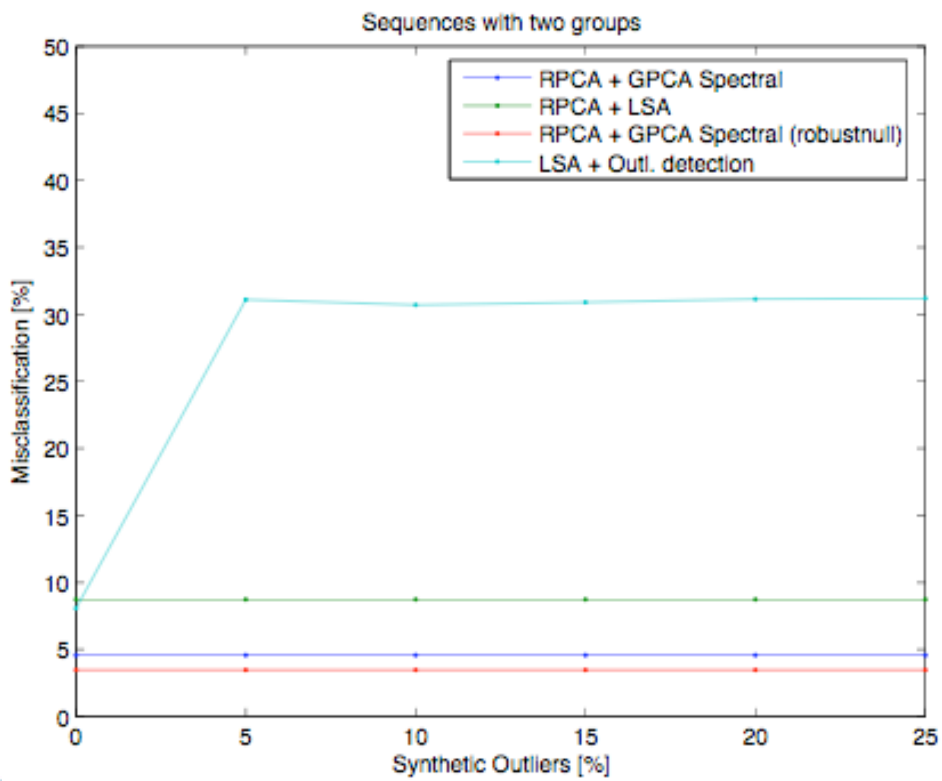
Experimental results: missing data sequences

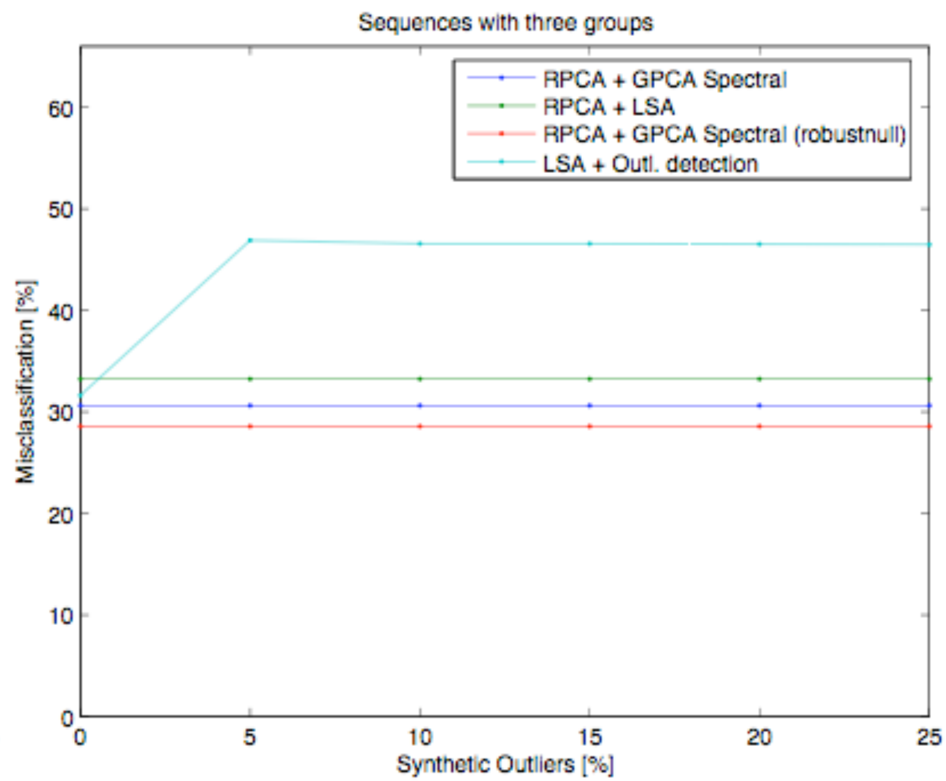
Sequence	P	F	n	missing data	PF+GPCA
oc1R2RC	686	40	3	8.98%	4.81%
oc1R2RC_g12	316	40	2	12.56%	0.00%
oc1R2RC_g13	520	40	2	11.46%	0.77%
oc1R2RC_g23	536	40	2	4.48%	2.24%
oc1R2RCT_g12	231	30	2	10.13%	3.46%
oc1R2RCT_g13	444	30	2	9.04%	11.49%
oc1R2RCT_g23	461	30	2	4.83%	7.81%
Average	456	35	2.1	8.78%	4.37%

- There is no clear correlation between amount of missing data and percentage of misclassification
- This could be because convergence of PF depends more on "where" missing data is located than on "how much" missing data there is

Experimental results: outliers

- For each sequence in the Hopkins 155 database, outlying points were drawn uniformly in [1,w]x[1,h] and used to generate outlying trajectories
- Robust SVD (De laTorre and Black'01) was used for projection, followed by GPCA or LSA for segmentation





Center for IMAGING SCIENCE

Conclusions

- For two motions
  - Algebraic methods (GPCA and LSA) are more accurate than statistical methods (RANSAC and MSL)
  - LSA performs better on full and independent sequences, while GPCA performs better on degenerate and partially dependent
  - LSA is sensitive to dimension of projection: d=4n better than d=5
  - MSL is very slow, RANSAC and GPCA are fast
- For three motions
  - GPCA is not very accurate, but is very fast
  - MSL is the most accurate, but it is very slow
  - LSA is almost as accurate as MSL and almost as fast as GPCA
- For data with outliers
  - Robust SVD combined with GPCA perform well

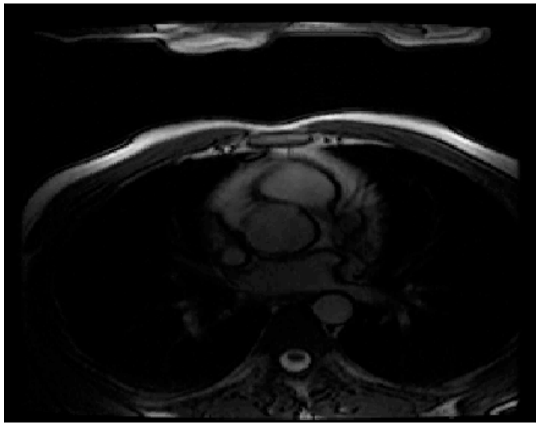
JHU vision lab

Segmentation of Dynamic Textures

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

Modeling a dynamic texture: fixed boundary

- 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

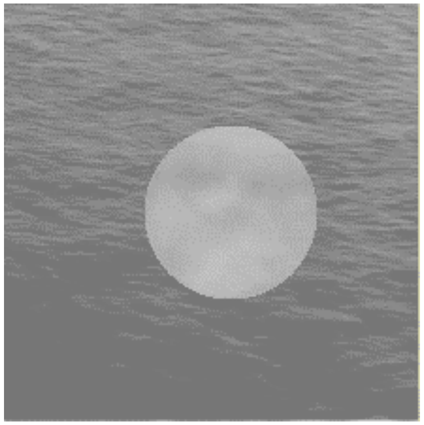
Modeling moving dynamic textures

- Can we recover the rigid motion of a camera looking at a moving dynamic texture?

DTCC: Dynamic Texture Constancy Constraint

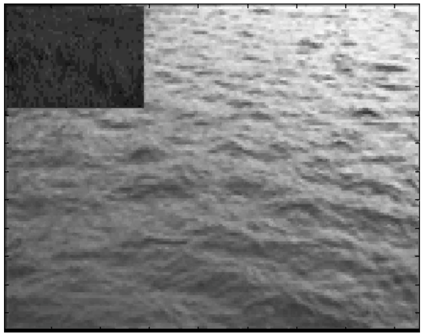
- Can we segment a scene containing multiple dynamic textures?

GPCA: Generalized Principal Component Analysis



- Can we segment a scene containing multiple moving dynamic textures?

GPCA + DTCC



Modeling moving dynamic textures

- A time invariant model cannot capture camera motion
  - A rigid scene is a 1st order LDS
  - $A = 1$, $z_t = 1$, $y_t = C =$ constant

$$z_{t+1} = A z_t + v_t$$
$$y_t = C z_t + w_t$$

- Camera motion can be modeled with a time varying LDS
  - A models nonrigid motion and C models rigid motion

dynamics

$$z_{t+1} = A z_t + v_t$$

images

$$y_t = C(t) z_t + w_t$$

appearance

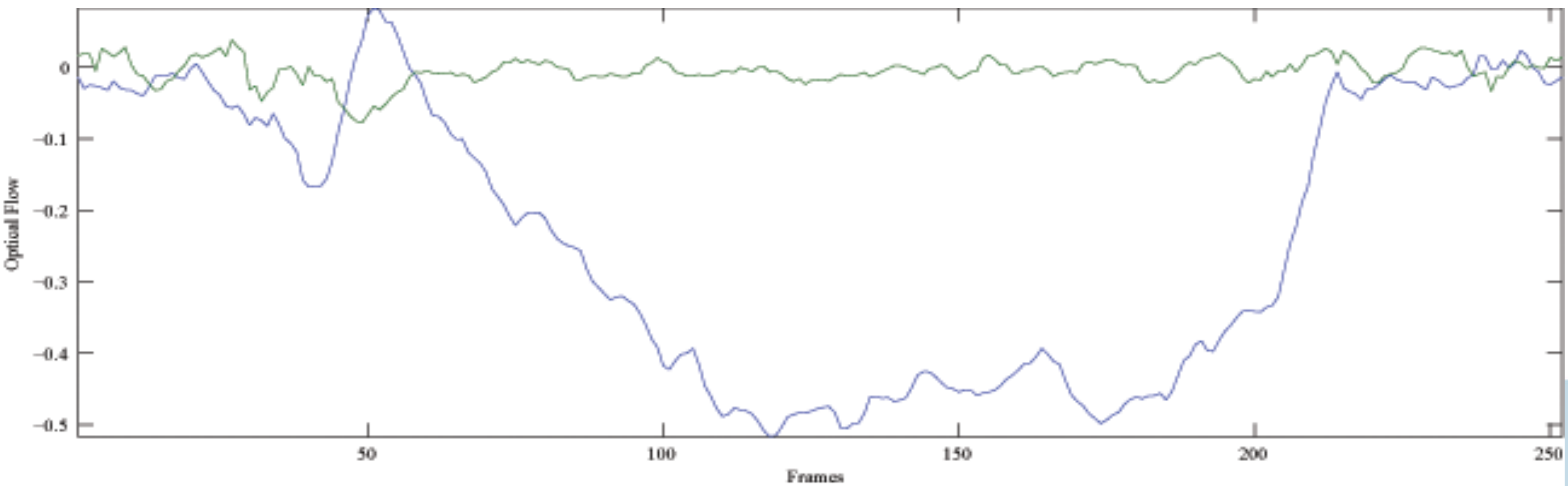
Optical flow of a moving dynamic texture

Static textures: optical flow from brightness constancy constraint (BCC)

$$I_x u + I_y v + I_t = 0$$

Dynamic textures: optical flow from dynamic texture constancy constraint (DTCC)

$$\boxed{\begin{array}{c} C_x u + C_y v + C_t = 0 \\ \Downarrow \\ I_x u + I_y v + C_t z_t = 0 \end{array}}$$

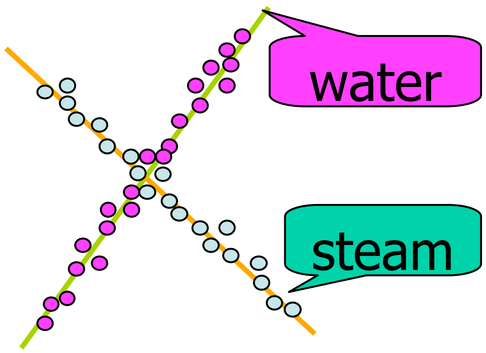


Segmenting non-moving dynamic textures

- One dynamic texture lives in the observability subspace

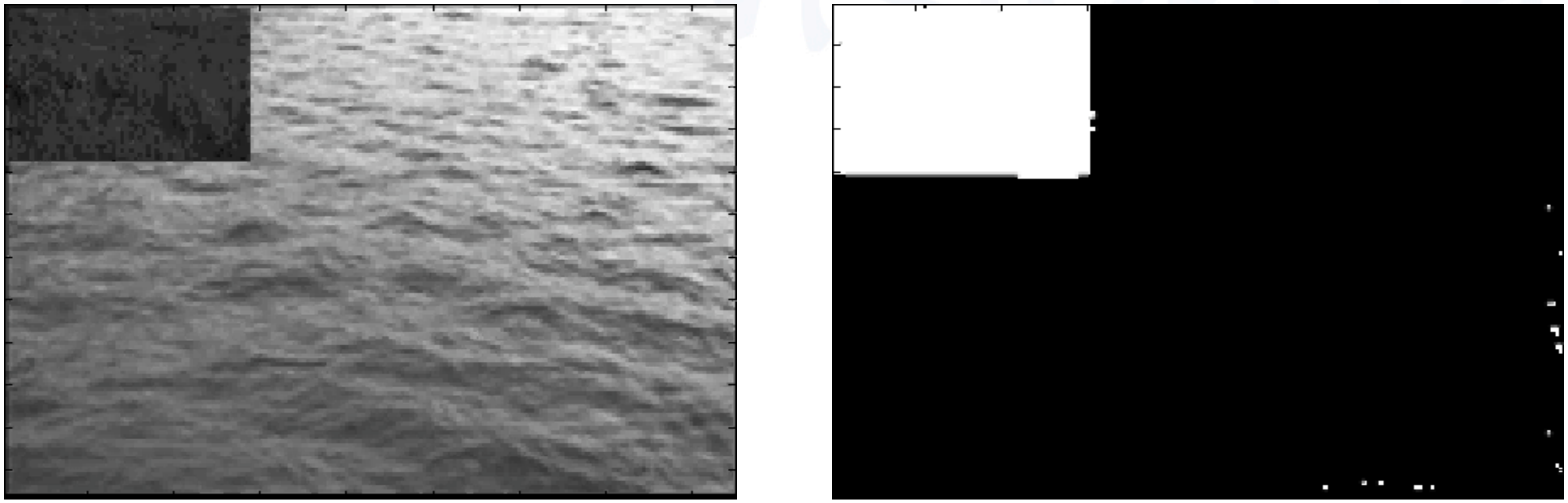
$$\begin{aligned} z_{t+1} &= A z_t + v_t \\ y_t &= C z_t + w_t \end{aligned} \qquad \begin{bmatrix} y_1 & y_2 & \cdots \\ y_2 & y_3 & \cdots \\ \vdots & & \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} \begin{bmatrix} z_1 & z_2 & \cdots \end{bmatrix}$$

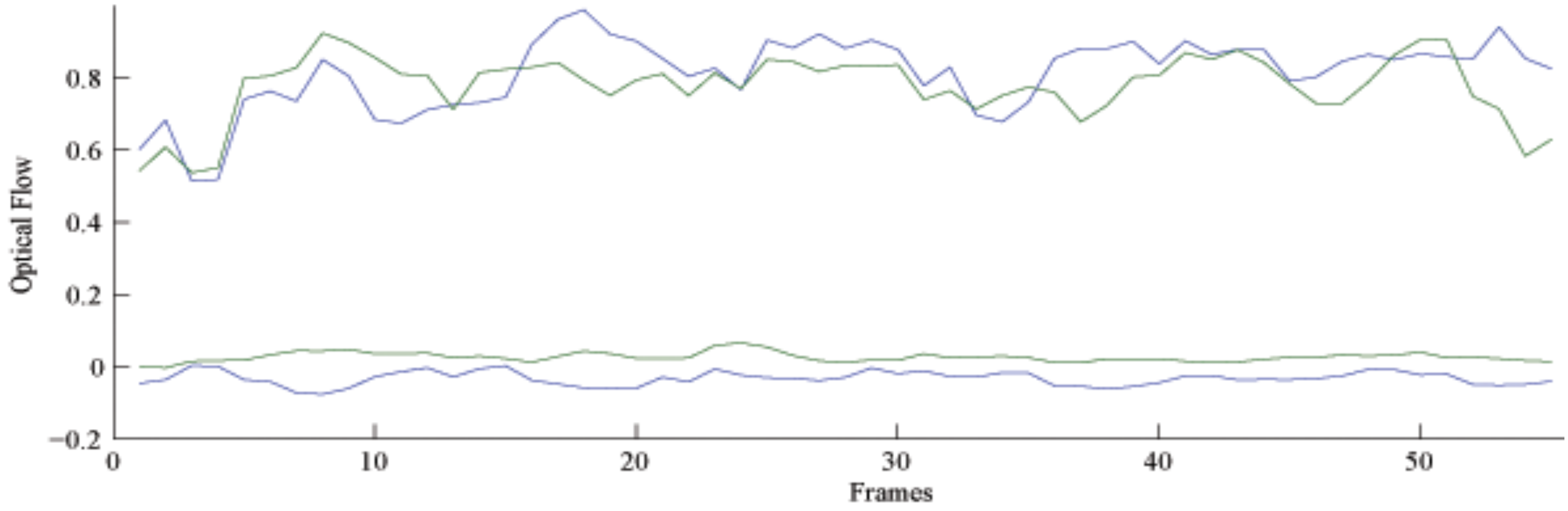
- Multiple textures live in multiple subspaces



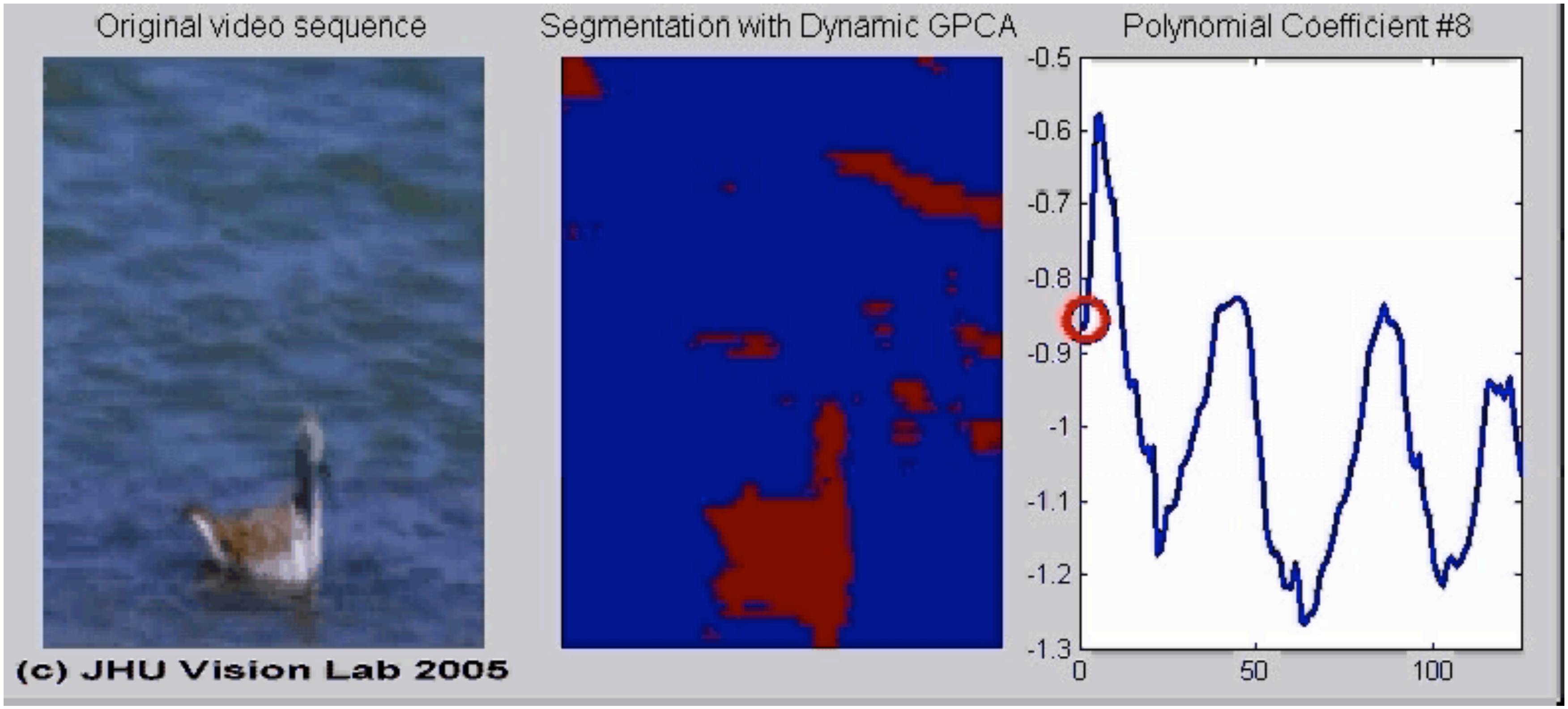
- Cluster the data using GPCA

Segmenting moving dynamic textures





Segmenting moving dynamic textures



Ocean-bird

Variational segmentation of dynamic textures

- How can we incorporate spatial coherence?
  - Model the dynamics with a mixture of AR models of order p

$$I(x,y,f) = a_0^j + \sum_{i=1}^{p} a_i^j I(x,y,f-i) + w(x,y,f)$$

  - Segment the scene by minimizing a spatial-temporal extension of the Chan-Vese energy functional

$$E = \mu|C| + \lambda_1 \int_{int(C)} \sum_{f=p+1}^{F} (I(x,y,f) - c_1(x,y,f))^2 dxdy$$
$$+\lambda_2 \int_{out(C)} \sum_{f=p+1}^{F} (I(x,y,f) - c_2(x,y,f))^2 dxdy$$

where

$$c_j(x,y,f) = a_0^j + \sum_{i=1}^{p} a_i^j I(x,y,f-i) \quad j = 1,2$$

Variational segmentation of dynamic textures

- Given the ARX parameters, we can solve for the implicit function φ by solving the PDE

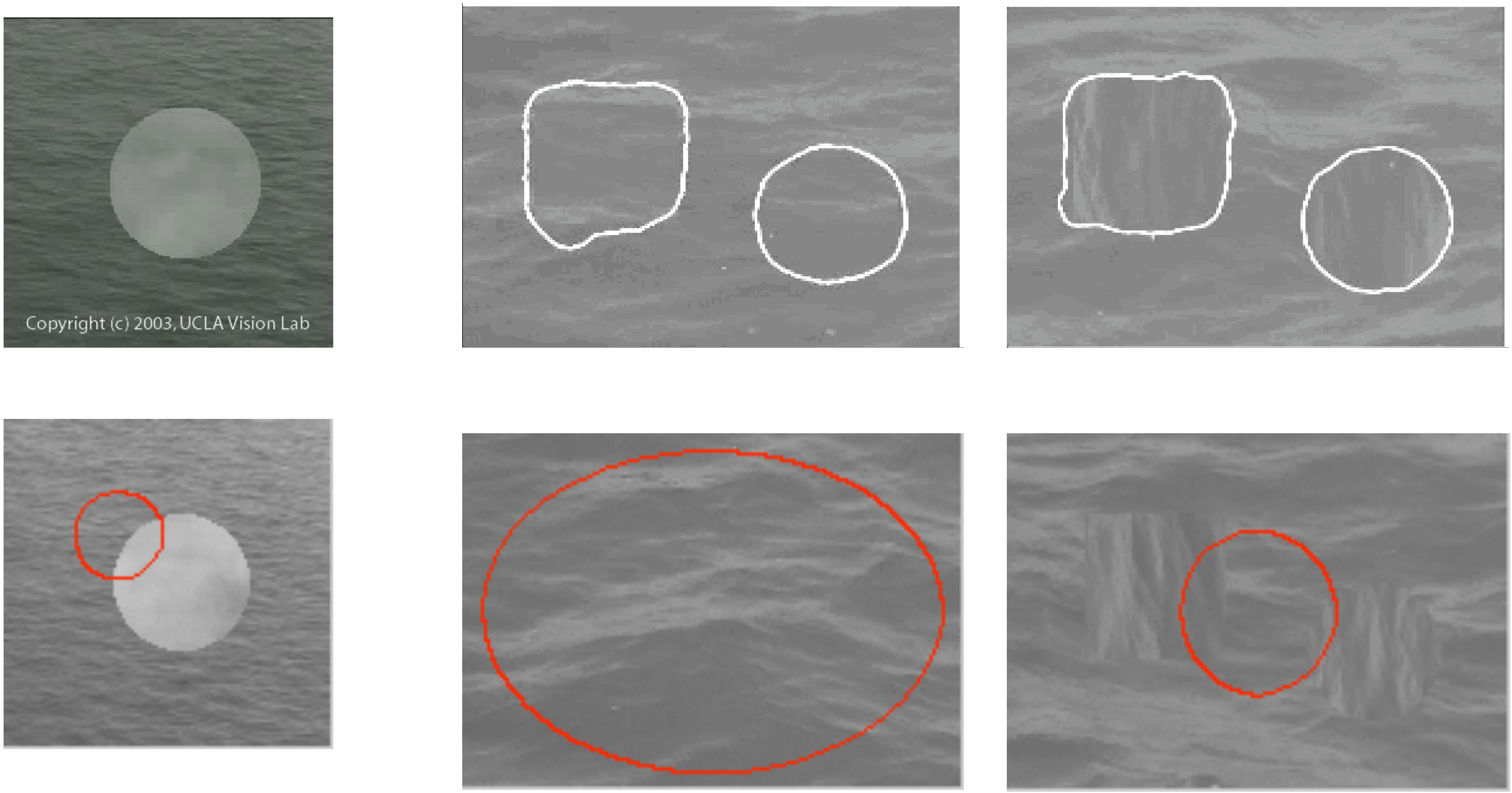
$$\frac{\partial \phi}{\partial t} = \delta(\phi)\left(\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|}\right) + \lambda_1 \int_{int(C)} \sum_{f=p+1}^{F} (I(x,y,f) - c_1(x,y,f))^2 dxdy \right.$$
$$\left. - \lambda_2 \int_{out(C)} \sum_{f=p+1}^{F} (I(x,y,f) - c_2(x,y,f))^2 dxdy \right)$$

- Given the implicit function φ, we can solve for the ARX parameters of the jth region by solving the linear system

$$\begin{bmatrix} 1 & I(x_1^j, y_1^j, f-1) & \cdots & I(x_1^j, y_1^j, f-p) \\ \vdots & \vdots & & \vdots \\ 1 & I(x_{k_j}^j, y_{k_j}^j, f-1) & \cdots & I(x_{k_j}^j, y_{k_j}^j, f-p) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} I(x_1^j, y_1^j, f) \\ \vdots \\ I(x_{k_j}^j, y_{k_j}^j, f) \end{bmatrix}$$

Variational segmentation of dynamic textures

- Fixed boundary segmentation results and comparison



Ocean-smoke | Ocean-dynamics | Ocean-appearance

Variational segmentation of dynamic textures

- Moving boundary segmentation results and comparison

Ocean-fire

Variational segmentation of dynamic textures

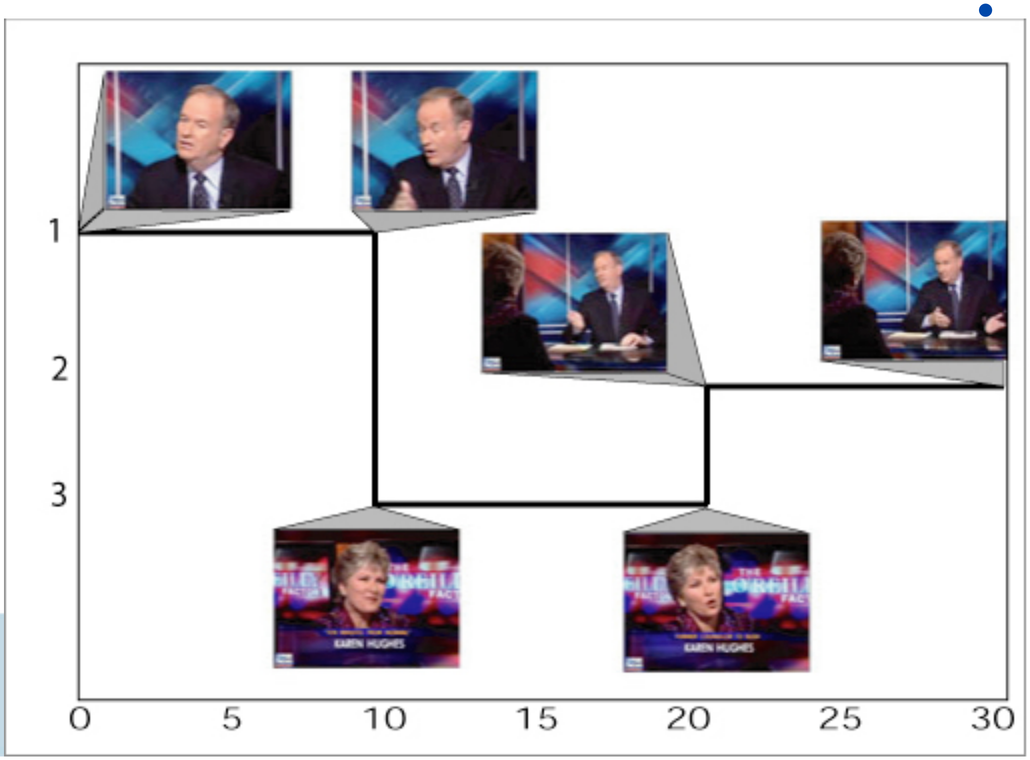
- Results on a real sequence

Raccoon on River

Temporal video segmentation

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

linear system



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

dynamics

images

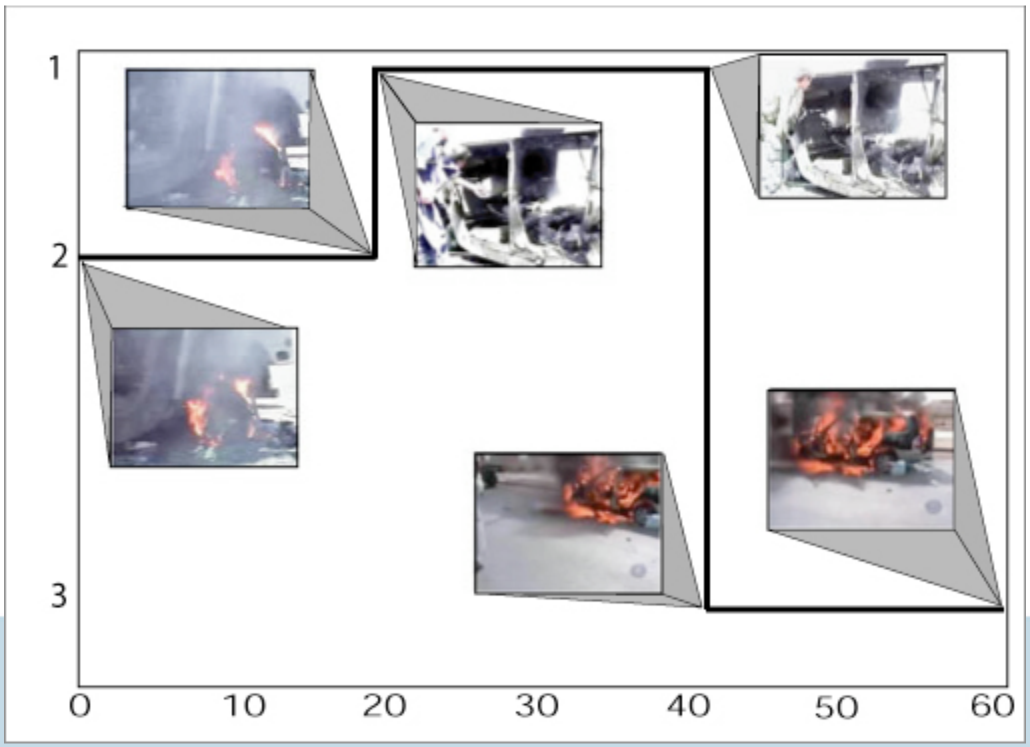
appearance

Apply GPCA to fit n=3 observability subspaces

Temporal video segmentation

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

linear system



Apply GPCA to fit n=3 observability subspaces

$$x_{t+1} = A x_t + v_t$$
$$y_t = C x_t + w_t$$

dynamics

images

appearance

Conclusions

- Many problems in computer vision can be posed as subspace clustering problems
  - Temporal video segmentation
  - 2-D and 3-D motion segmentation
  - Dynamic texture segmentation
  - Nonrigid motion segmentation

- These problems can be solved using GPCA: an algorithm for clustering subspaces
  - Deals with unknown and possibly different dimensions
  - Deals with arbitrary intersections among the subspaces

- GPCA is based on
  - Projecting data onto a low-dimensional subspace
  - Recursively fitting polynomials to projected subspaces
  - Differentiating polynomials to obtain a basis

For more information,

Vision, Dynamics and Learning Lab

@

Johns Hopkins University

http://www.vision.jhu.edu

Thank You!

Center for IMAGING SCIENCE