High-Dimensional Multi-Model Estimation
– Its Algebra, Statistics, and Sparse Representation

Allen Y. Yang
yang@eecs.berkeley.edu

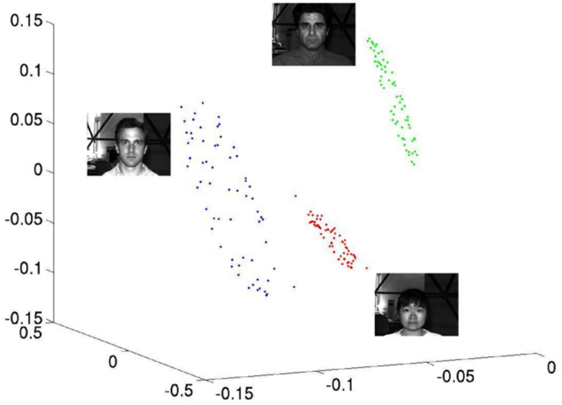
Feb 3, 2009, University of California, Berkeley

High-Dimensional Data: Images, Videos, etc...

Figure: Dimension of an image: $1000 \times 700 \times 3 > 2 \mathit{million}$!

HD data are often **multi-model**

Mixture-Subspace Model



Face Recognition: *"Where amazing happens!"*

Face Recognition: *"Where amazing happens!"*

Face Recognition: *"Where amazing happens!"*

Face Recognition: *"Where amazing happens!"*

Face Recognition: *"Where amazing happens!"*

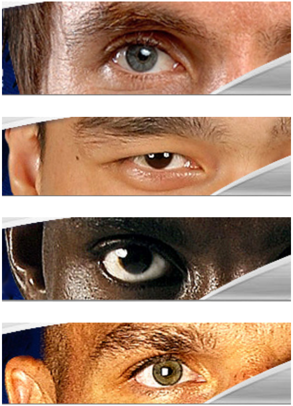
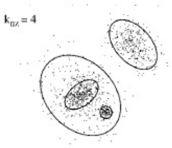


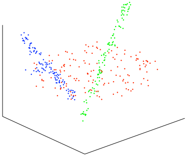
Figure: Steve Nash, Yao Ming, Kevin Garnett, Jason Kidd.

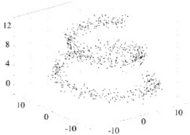
How to let computer compete with human perception?

How to let computer compete with human perception?

- How to determine a class of models and the number of models?

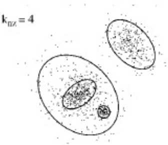
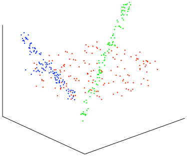
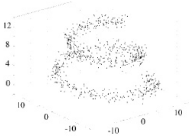




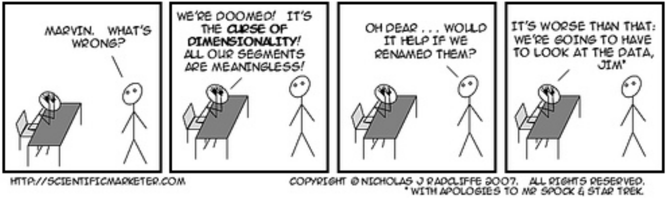


How to let computer compete with human perception?

- How to determine a class of models and the number of models?

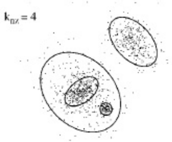




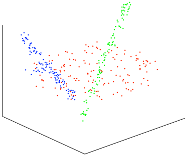
- Curse of dimensionality! [Richard Bellman 1957]

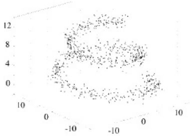


How to let computer compete with human perception?

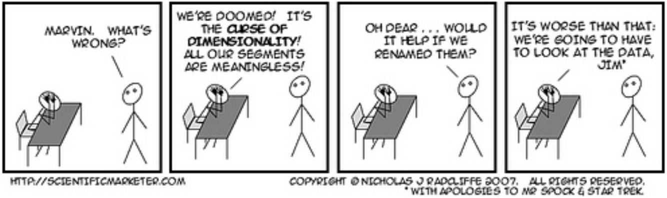
- How to determine a class of models and the number of models?







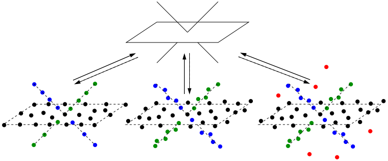
- Curse of dimensionality! [Richard Bellman 1957]



- To make things worse: Robust to high noise and outliers?

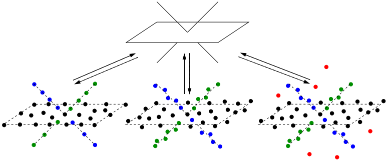
Outline

1. **Unsupervised** segmentation
   Segment samples drawn from $\mathcal{A} = S_1 \cup S_2 \cup \ldots \cup S_K$ in $\mathbb{R}^D$, and estimate model parameters.

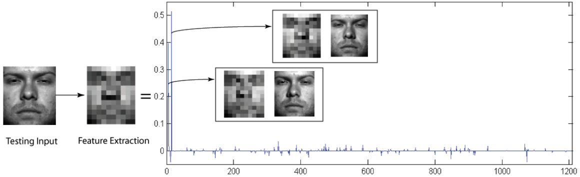


Outline

1. **Unsupervised** segmentation
   Segment samples drawn from $\mathcal{A} = S_1 \cup S_2 \cup \ldots \cup S_K$ in $\mathbb{R}^D$, and estimate model parameters.



2. **Supervised** recognition
   Assume training examples $\{A_1, \cdots, A_K\}$ for K models. Given a test sample $\mathbf{y}$, determine its membership label$(\mathbf{y}) \in [1, 2, \cdots, K]$.



Affine Motion Segmentation

Assume multiple 3-D objects far away from the camera in a dynamic scene

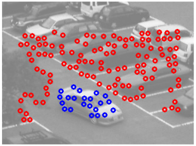
- 3-D features $\mathbf{p}_1, \ldots, \mathbf{p}_N \in \mathbb{R}^3$ are tracked in F image frames.
- Image of $\mathbf{p}_i$ in jth frame:

$$\mathbf{m}_{ij} \doteq \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix}^T = A_j \mathbf{p}_i + \mathbf{b}_j \in \mathbb{R}^2, \quad j = 1, \ldots, F.$$

- Stack images of $\mathbf{p}_i$ in all F frames

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{m}_{i1} \\ \vdots \\ \mathbf{m}_{iF} \end{bmatrix} = \begin{bmatrix} A_1 & \mathbf{b}_1 \\ \vdots & \\ A_F & \mathbf{b}_F \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} \in \mathbb{R}^{2F}.$$

parking-lot movie



Affine Motion Segmentation

Assume multiple 3-D objects far away from the camera in a dynamic scene

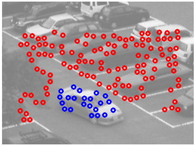
- 3-D features $\mathbf{p}_1, \ldots, \mathbf{p}_N \in \mathbb{R}^3$ are tracked in F image frames.
- Image of $\mathbf{p}_i$ in jth frame:

$$\mathbf{m}_{ij} \doteq \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix}^T = A_j \mathbf{p}_i + \mathbf{b}_j \in \mathbb{R}^2, \quad j = 1, \ldots, F.$$

- Stack images of $\mathbf{p}_i$ in all F frames

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{m}_{i1} \\ \vdots \\ \mathbf{m}_{iF} \end{bmatrix} = \begin{bmatrix} A_1 & \mathbf{b}_1 \\ \vdots & \\ A_F & \mathbf{b}_F \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} \in \mathbb{R}^{2F}.$$

parking-lot movie

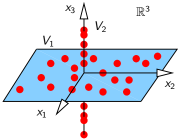


Challenge: Affine Motion Segmentation $\Rightarrow$ Subspace Segmentation

Each motion satisfies a 4-D subspace model.

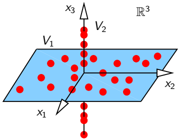
Generalized Principal Component Analysis (GPCA)

1. For a single subspace
   - $V_1^\perp : (x_3 = 0)$
   - $V_2^\perp : (x_1 = 0) \& (x_2 = 0)$



Generalized Principal Component Analysis (GPCA)

1. For a single subspace
   - $V_1^{\perp} : (x_3 = 0)$
   - $V_2^{\perp} : (x_1 = 0)\&(x_2 = 0)$

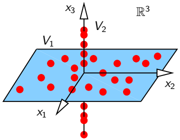


2. For $\mathcal{A} = V_1 \cup V_2$

$$\forall \mathbf{z} = (x_1, x_2, x_3)^T, \quad \mathbf{z} \in V_1 \cup V_2 \Leftrightarrow \{x_3 = 0\} | \{(x_1 = 0)\&(x_2 = 0)\}$$

Generalized Principal Component Analysis (GPCA)

1. For a single subspace
   - $V_1^\perp : (x_3 = 0)$
   - $V_2^\perp : (x_1 = 0) \& (x_2 = 0)$



2. For $\mathcal{A} = V_1 \cup V_2$

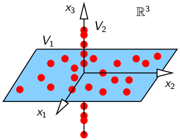
$$\forall \mathbf{z} = (x_1, x_2, x_3)^T, \quad \mathbf{z} \in V_1 \cup V_2 \Leftrightarrow \{x_3 = 0\} | \{(x_1 = 0) \& (x_2 = 0)\}$$

3. By De Morgan's law

$$\{x_3 = 0\} | \{(x_1 = 0) \& (x_2 = 0)\} \Leftrightarrow (x_1 x_3 = 0) \& (x_2 x_3 = 0) \Leftrightarrow \begin{cases} x_1 x_3 = 0 \\ x_2 x_3 = 0 \end{cases}$$

Generalized Principal Component Analysis (GPCA)

1. For a single subspace
   - $V_1^{\perp} : (x_3 = 0)$
   - $V_2^{\perp} : (x_1 = 0)\&(x_2 = 0)$



2. For $\mathcal{A} = V_1 \cup V_2$

$$\forall \mathbf{z} = (x_1, x_2, x_3)^T, \quad \mathbf{z} \in V_1 \cup V_2 \Leftrightarrow \{x_3 = 0\} | \{(x_1 = 0)\&(x_2 = 0)\}$$

3. By De Morgan's law

$$\{x_3 = 0\} | \{(x_1 = 0)\&(x_2 = 0)\} \Leftrightarrow (x_1x_3 = 0)\&(x_2x_3 = 0) \Leftrightarrow \begin{cases} x_1x_3 = 0 \\ x_2x_3 = 0 \end{cases}$$

4. **Vanishing polynomials**: $p_1 = x_1x_3, p_2 = x_2x_3$

Equivalence Relation

- The **equivalence** between K subspaces and Kth-degree vanishing polynomials
  1. Given $p_1 = x_1x_3, p_2 = x_2x_3$, $V_1 \cup V_2$ uniquely determined.
  2. All vanishing polynomials of arbitrary degree for $V_1 \cup V_2$ generated by $p_1 = x_1x_3, p_2 = x_2x_3$.

Equivalence Relation

- The **equivalence** between K subspaces and Kth-degree vanishing polynomials
  1. Given $p_1 = x_1 x_3, p_2 = x_2 x_3$, $V_1 \cup V_2$ uniquely determined.
  2. All vanishing polynomials of arbitrary degree for $V_1 \cup V_2$ generated by $p_1 = x_1 x_3, p_2 = x_2 x_3$.
- Kth-degree vanishing polynomials $I_K(\mathcal{A})$ as a **global signature**

Equivalence Relation

- The **equivalence** between K subspaces and Kth-degree vanishing polynomials
  1. Given $p_1 = x_1x_3, p_2 = x_2x_3$, $V_1 \cup V_2$ uniquely determined.
  2. All vanishing polynomials of arbitrary degree for $V_1 \cup V_2$ generated by $p_1 = x_1x_3, p_2 = x_2x_3$.
- Kth-degree vanishing polynomials $I_K(\mathcal{A})$ as a **global signature**
- $I_K(\mathcal{A})$ is a **polynomial subspace**.

Subspace Properties

If $p_1, p_2 \in I_K(\mathcal{A})$, $p_1(\mathbf{x}) = 0$ and $p_2(\mathbf{x}) = 0$

1. Closed under **addition**: $(p_1 + p_2)(\mathbf{x}) = 0 \Rightarrow (p_1 + p_2) \in I_K(\mathcal{A})$.
2. Closed under **scalar multiplication**: $\forall a \in \mathbb{R}$, $ap_1(\mathbf{x}) = 0 \Rightarrow ap_1 \in I_K(\mathcal{A})$.

Equivalence Relation

- The **equivalence** between K subspaces and Kth-degree vanishing polynomials
  1. Given $p_1 = x_1x_3$, $p_2 = x_2x_3$, $V_1 \cup V_2$ uniquely determined.
  2. All vanishing polynomials of arbitrary degree for $V_1 \cup V_2$ generated by $p_1 = x_1x_3$, $p_2 = x_2x_3$.
- Kth-degree vanishing polynomials $I_K(\mathcal{A})$ as a **global signature**
- $I_K(\mathcal{A})$ is a **polynomial subspace**.

Subspace Properties

If $p_1, p_2 \in I_K(\mathcal{A})$, $p_1(\mathbf{x}) = 0$ and $p_2(\mathbf{x}) = 0$

1. Closed under **addition**: $(p_1 + p_2)(\mathbf{x}) = 0 \Rightarrow (p_1 + p_2) \in I_K(\mathcal{A})$.
2. Closed under **scalar multiplication**: $\forall a \in \mathbb{R}$, $ap_1(\mathbf{x}) = 0 \Rightarrow ap_1 \in I_K(\mathcal{A})$.

- $I_K(\mathcal{A})$ is determined by a linearly-independent polynomial basis.

Estimation of Vanishing Polynomials

1. **Veronese embedding**: Given N samples $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^3$,

$$\begin{aligned} L_2 &\doteq [\nu_2(\mathbf{x}_1), \dots, \nu_2(\mathbf{x}_N)] \in \mathbb{R}^{M_2^{[3]} \times N} \\ &= \begin{bmatrix} \cdots & (x_1)^2 & \cdots \\ \cdots & (x_1 x_2) & \cdots \\ \cdots & (x_1 x_3) & \cdots \\ \cdots & (x_2)^2 & \cdots \\ \cdots & (x_2 x_3) & \cdots \\ \cdots & (x_3)^2 & \cdots \end{bmatrix} \end{aligned}$$

Estimation of Vanishing Polynomials

1. **Veronese embedding**: Given N samples $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^3$,

$$
\begin{aligned}
L_2 & \doteq [\nu_2(\mathbf{x}_1), \ldots, \nu_2(\mathbf{x}_N)] \in \mathbb{R}^{M_2^{[3]} \times N} \\
& = \begin{bmatrix} \cdots & (x_1)^2 & \cdots \\ \cdots & (x_1 x_2) & \cdots \\ \cdots & (x_1 x_3) & \cdots \\ \cdots & (x_2)^2 & \cdots \\ \cdots & (x_2 x_3) & \cdots \\ \cdots & (x_3)^2 & \cdots \end{bmatrix}
\end{aligned}
$$

2. The null space of L_2 is $\begin{matrix} \mathbf{c}_1 = [0,0,1,0,0,0] \\ \mathbf{c}_2 = [0,0,0,0,1,0] \end{matrix} \Rightarrow \begin{matrix} p_1 = \mathbf{c}_1 \nu_2(\mathbf{x}) = x_1 x_3 \\ p_2 = \mathbf{c}_2 \nu_2(\mathbf{x}) = x_2 x_3 \end{matrix}$

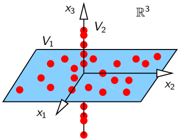


Figure: $2nd$-degree vanishing polynomials: $p_1 = x_1 x_3$, $p_2 = x_2 x_3$.

Calculate Subspace Basis Vectors using Polynomial Derivatives

1. $V_1^\perp, \cdots, V_K^\perp$ recovered by the *derivatives*

$$\nabla_{\mathbf{x}} P = [\,\nabla_{\mathbf{x}} p_1 \;\; \nabla_{\mathbf{x}} p_2\,] = \begin{bmatrix} x_3 & 0 \\ 0 & x_3 \\ x_1 & x_2 \end{bmatrix}.$$

2. Pick $\mathbf{z} = [1, 1, 0]^T \in V_1$, then $\nabla_{\mathbf{x}} P(\mathbf{z}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$.

   Pick $\mathbf{z} = [0, 0, 1]^T \in V_2$, then $\nabla_{\mathbf{x}} P(\mathbf{z}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$.

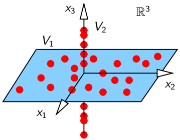


Figure: $P(\mathbf{x}) \doteq [p_1(\mathbf{x}) \;\; p_2(\mathbf{x})] = [x_1 x_3, x_2 x_3]$.

Calculate Subspace Basis Vectors using Polynomial Derivatives

1. $V_1^\perp, \cdots, V_K^\perp$ recovered by the *derivatives*

$$\nabla_{\mathbf{x}} P = [\, \nabla_{\mathbf{x}} p_1 \;\; \nabla_{\mathbf{x}} p_2 \,] = \begin{bmatrix} x_3 & 0 \\ 0 & x_3 \\ x_1 & x_2 \end{bmatrix}.$$

2. Pick $\mathbf{z} = [1,1,0]^T \in V_1$, then $\nabla_{\mathbf{x}} P(\mathbf{z}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$.

   Pick $\mathbf{z} = [0,0,1]^T \in V_2$, then $\nabla_{\mathbf{x}} P(\mathbf{z}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$.

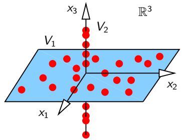
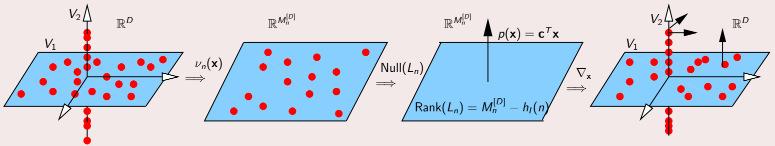
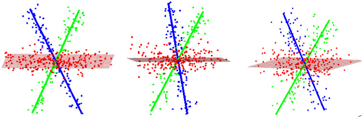


Figure: $P(\mathbf{x}) \doteq [p_1(\mathbf{x}) \;\; p_2(\mathbf{x})] = [x_1 x_3, x_2 x_3]$.

Diagram of GPCA

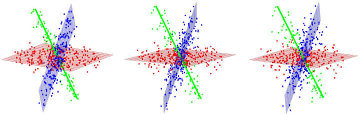


Robust GPCA



(a) .08 (b) .12 (c) .16

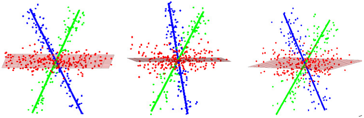
Figure: (2, 1, 1) with various noise-to-signal ratios



(a) .08 (b) .12 (c) .16

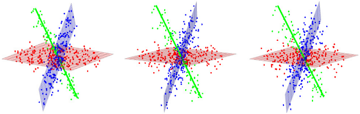
Figure: (2, 2, 1) with various noise-to-signal ratios

Robust GPCA



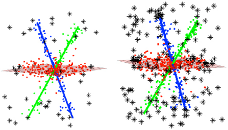
(a) .08 (b) .12 (c) .16

Figure: (2, 1, 1) with various noise-to-signal ratios



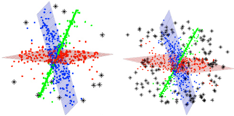
(a) .08 (b) .12 (c) .16

Figure: (2, 2, 1) with various noise-to-signal ratios



(a) 12% (b) 32%

Figure: One plane and two lines with various outlier percentages



(a) 12% (b) 32%

Figure: Two planes and one line with various outlier percentages

Outlier Elimination

Figure: Elimination of outliers.

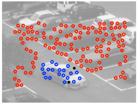
Reference:

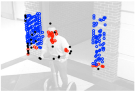
SIAM Review: *Estimation of subspace arrangements with applications in modeling and segmenting mixed data*, 2008.

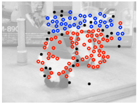
Experiment: Affine Motion Segmentation

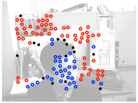
Sequences:

RGPCA:





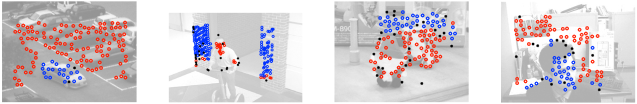




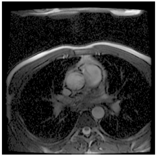
Experiment: Affine Motion Segmentation

Sequences:

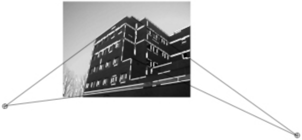
RGPCA:



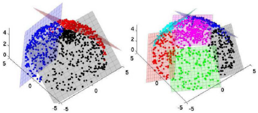
- Other applications



(d) Image/Video Segmentation



(e) Vanishing Point Detection



(f) Manifold Fitting

Summary: GPCA

Advantages:

- Closed-form algebraic solution, not iterative.
- Segmentation of subspaces with mixed dimensions.
- Robust to noise and outliers.

Summary: GPCA

Advantages:

- Closed-form algebraic solution, not iterative.
- Segmentation of subspaces with mixed dimensions.
- Robust to noise and outliers.

Limitations:

- **Only apply to mixture linear subspaces. (How about mixture nonlinear manifolds?)**

Summary: GPCA

Advantages:

- Closed-form algebraic solution, not iterative.
- Segmentation of subspaces with mixed dimensions.
- Robust to noise and outliers.

Limitations:

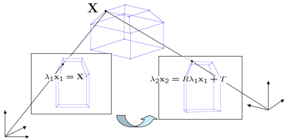
- **Only apply to mixture linear subspaces. (How about mixture nonlinear manifolds?)**
- **User provides correct subspace number and dimensions. (How to select a good mixture model?)**

Mixture Perspective Motions

Mixture Perspective Motions

Given two image correspondences $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$

- Epipolar

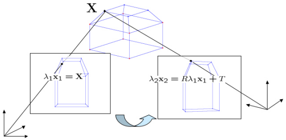


$$\mathbf{x}_2^T \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \mathbf{x}_1 = 0$$

Mixture Perspective Motions

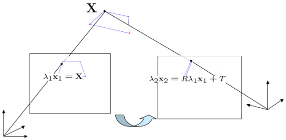
Given two image correspondences $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$

- Epipolar



$$\mathbf{x}_2^T \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \mathbf{x}_1 = 0$$

- Homography

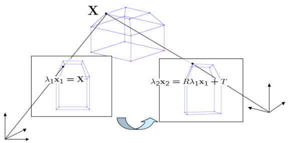


$$\mathbf{x}_2 \times \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \mathbf{x}_1 = 0$$

Mixture Perspective Motions

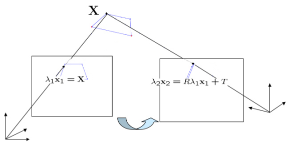
Given two image correspondences $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$

- Epipolar



$$\mathbf{x}_2^T \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \mathbf{x}_1 = 0$$

- Homography



$$\mathbf{x}_2 \times \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \mathbf{x}_1 = 0$$

Segmentation of mixture perspective motions

Each perspective constraint is linear w.r.t. $(\mathbf{x}_1, \mathbf{x}_2)$, but in different form!

Quadratic Manifolds in Joint Image Space

Joint image space: Stack $\mathbf{x}_1 = (x_1, y_1, 1)^T$ and $\mathbf{x}_2 = (x_2, y_2, 1)^T$

$$\mathbf{y} = (x_1, y_1, x_2, y_2, 1)^T \in \mathbb{R}^5$$

Quadratic Manifolds in Joint Image Space

Joint image space: Stack $\mathbf{x}_1 = (x_1, y_1, 1)^T$ and $\mathbf{x}_2 = (x_2, y_2, 1)^T$

$$\mathbf{y} = (x_1, y_1, x_2, y_2, 1)^T \in \mathbb{R}^5$$

- Quadratic fundamental manifold (QFM)

$$\mathbf{y}^T A \mathbf{y} \doteq \mathbf{y}^T \begin{pmatrix} 0 & 0 & f_{11} & f_{21} & f_{31} \\ 0 & 0 & f_{12} & f_{22} & f_{32} \\ f_{11} & f_{12} & 0 & 0 & f_{13} \\ f_{21} & f_{22} & 0 & 0 & f_{23} \\ f_{31} & f_{32} & f_{13} & f_{23} & 2f_{33} \end{pmatrix} \mathbf{y} = 0. \tag{1}$$

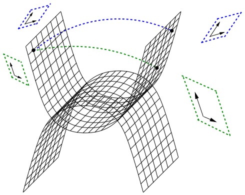
- Quadratic homograpy manifold (QHM)

$$\begin{aligned} \mathbf{y}^T B_1 \mathbf{y} &\doteq \mathbf{y}^T \begin{pmatrix} 0 & 0 & 0 & h_{31} & -h_{21} \\ 0 & 0 & 0 & h_{32} & -h_{22} \\ 0 & 0 & 0 & 0 & 0 \\ h_{31} & h_{32} & 0 & 0 & h_{33} \\ -h_{21} & -h_{22} & 0 & h_{33} & -2h_{23} \end{pmatrix} \mathbf{y} = 0, \\ \mathbf{y}^T B_2 \mathbf{y} &\doteq \mathbf{y}^T \begin{pmatrix} 0 & 0 & -h_{31} & 0 & h_{11} \\ 0 & 0 & -h_{32} & 0 & h_{12} \\ -h_{31} & -h_{32} & 0 & 0 & -h_{33} \\ 0 & 0 & 0 & 0 & 0 \\ h_{11} & h_{12} & -h_{33} & 0 & 2h_{13} \end{pmatrix} \mathbf{y} = 0, \\ \mathbf{y}^T B_3 \mathbf{y} &\doteq \mathbf{y}^T \begin{pmatrix} 0 & 0 & h_{21} & -h_{11} & 0 \\ 0 & 0 & h_{22} & -h_{12} & 0 \\ h_{21} & h_{22} & 0 & 0 & h_{23} \\ -h_{11} & -h_{12} & 0 & 0 & -h_{13} \\ 0 & 0 & h_{23} & -h_{13} & 0 \end{pmatrix} \mathbf{y} = 0. \end{aligned} \tag{2}$$

Segmentation of Quadratic Manifolds

- Convert mixture perspective motion as **segmentation of mixture quadratic manifolds** defined by

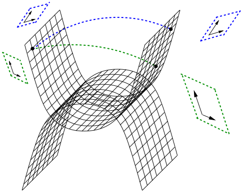
$$p_j(\mathbf{y}) \doteq \mathbf{y}^T Q_j \mathbf{y} = 0. \tag{3}$$



Segmentation of Quadratic Manifolds

- Convert mixture perspective motion as **segmentation of mixture quadratic manifolds** defined by

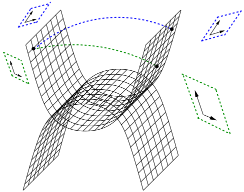
$$p_j(\mathbf{y}) \doteq \mathbf{y}^T Q_j \mathbf{y} = 0. \tag{3}$$



- **Vanishing polynomials** (as global signature): The set of $2K$th degree polynomials $I_{2K}(\mathcal{A})$ **uniquely determines** $\mathcal{A} = S_1 \cup \cdots \cup S_K$.

Segmentation of Quadratic Manifolds

- Convert mixture perspective motion as **segmentation of mixture quadratic manifolds** defined by
$$p_j(\mathbf{y}) \doteq \mathbf{y}^T Q_j \mathbf{y} = 0. \tag{3}$$



- **Vanishing polynomials** (as global signature): The set of $2K$th degree polynomials $I_{2K}(\mathcal{A})$ **uniquely determines** $\mathcal{A} = S_1 \cup \cdots \cup S_K$.

Robust Algebraic Segmentation

$$Y = \{\mathbf{y}_1, \cdots, \mathbf{y}_n\} \Rightarrow I_{2K}(\mathcal{A}) \Rightarrow \mathcal{A} \Rightarrow \{S_1, \cdots, S_K\}$$

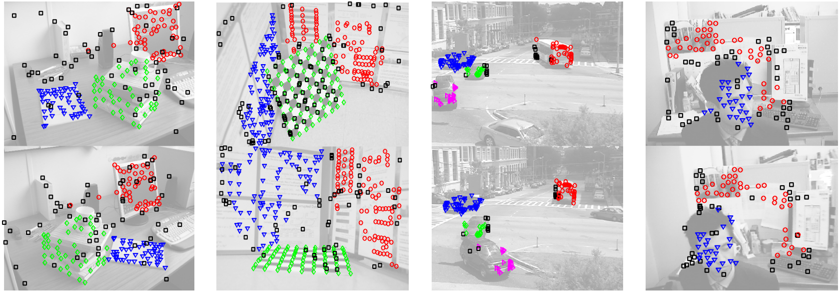
Reference:

IJCV (draft): *Robust Algebraic Segmentation of Mixed Rigid-Body and Planar Motions*, 2008.

boxes	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	9.24%	**0.84%**	1.68%	**0.84%**
VR	36.97%	84.87%	100%	87.39%
carsnbus3	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	45.75%	12.55%	2.83%	**1.62%**
VR	83.81%	90.28%	97.17%	85.83%
deliveryvan	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	23.23%	10.63%	5.91%	**0.39%**
VR	97.64%	96.85%	100%	94.09%
desk	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	9.00%	2.50%	3.00%	**0.50%**
VR	55.50%	93.50%	91.50%	93.50%
lightbulb	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	39.52%	**0.00%**	**0.00%**	**0.00%**
VR	76.19%	82.86%	100%	99.52 %
manycars	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	30.56%	22.22%	**0.00%**	**0.00%**
VR	90.28%	95.83%	100%	88.89%
man-in-office	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	20.56%	34.58%	20.56%	**11.21%**
VR	89.72%	95.33%	84.11%	82.24%
nrbooks3	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	12.38%	9.05%	5.48%	**0.95%**
VR	41.19%	65.48%	94.29%	88.33%
office	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	2.28%	0.33%	10.42%	**0.00%**
VR	89.59%	90.55%	86.97%	93.49%
parking-lot	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	7.86%	5.00%	3.57%	**2.86%**
VR	98.57%	96.43%	100%	97.86%
posters-checkerboard	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	20.58%	1.06%	9.23%	**0.00%**
VR	49.87%	97.36%	70.71%	95.25%
posters-keyboard	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	8.59%	**0.25%**	10.61%	0.51%
VR	56.06%	83.33%	78.03%	88.13%
toys-on-table	MLESAC	MC-RANSAC	RAS	RAS+RANSAC
FPR	38.10%	38.10%	15.08%	**7.94%**
VR	91.27%	92.86%	81.75%	77.78%

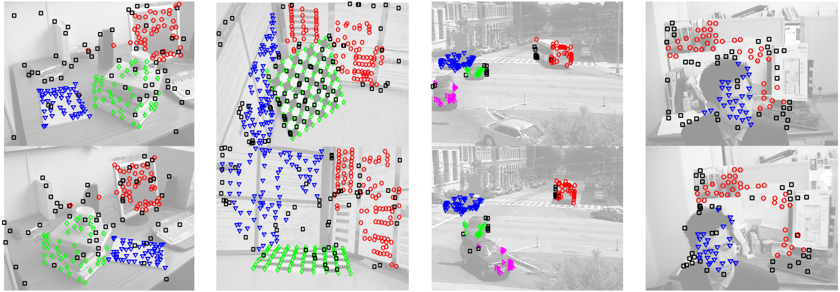
Experiment

1. Visualization

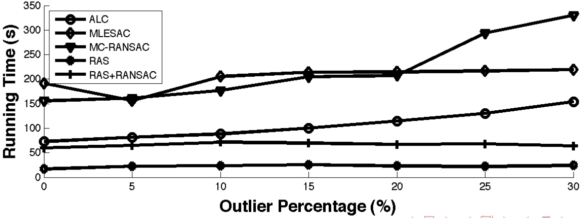


Experiment

1. Visualization



2. Faster than RANSAC!



Summary: Robust Algebraic Segmentation

Advantages:

- **Segmentation of quadratic manifolds with mixed dimensions.**
- Closed-form algebraic solution, not iterative.
- Robust to noise and outliers.

Limitations:

- **User provides correct subspace number and dimensions. (How to select a good mixture model?)**

Lossy Minimum Description Length (LMDL)

1. **Lossy coding length** $L_\epsilon(V, \mathcal{A})$:
   Quantize $V = (v_1, \cdots, v_N) \in \mathbb{R}^{D\times N}$ as **a sequence of binary bits** up to a distortion $\mathbb{E}[\|v_i - \hat{v}_i\|^2] \le \epsilon^2$.

Lossy Minimum Description Length (LMDL)

1. **Lossy coding length** $L_\epsilon(V, \mathcal{A})$:
   Quantize $V = (v_1, \cdots, v_N) \in \mathbb{R}^{D\times N}$ as **a sequence of binary bits** up to a distortion $\mathbb{E}[\|v_i - \hat{v}_i\|^2] \leq \epsilon^2$.
2. **Lossy MDL**

$$\mathcal{A}^*(\epsilon) = \arg\min\{L_\epsilon(V, \mathcal{A}) + \text{Overhead}(\mathcal{A})\}.$$

Lossy Minimum Description Length (LMDL)

1. **Lossy coding length** $L_\epsilon(V, \mathcal{A})$:
   Quantize $V = (v_1, \cdots, v_N) \in \mathbb{R}^{D \times N}$ as **a sequence of binary bits** up to a distortion $\mathbb{E}[\|v_i - \hat{v}_i\|^2] \leq \epsilon^2$.
2. **Lossy MDL**

$$\mathcal{A}^*(\epsilon) = \arg\min\{L_\epsilon(V, \mathcal{A}) + \text{Overhead}(\mathcal{A})\}.$$

3. For mixture subspace model
   - Model V_i as a (degenerate) Gaussian model

$$\textbf{Bit rate: } R(V_i) = \frac{1}{2}\log_2 \det(I + \frac{D}{\epsilon^2 N_i} V_i V_i^T).$$

Lossy Minimum Description Length (LMDL)

1. **Lossy coding length** $L_\epsilon(V, \mathcal{A})$:
   Quantize $V = (v_1, \cdots, v_N) \in \mathbb{R}^{D\times N}$ as **a sequence of binary bits** up to a distortion $\mathbb{E}[\|v_i - \hat{v}_i\|^2] \leq \epsilon^2$.
2. **Lossy MDL**
$$\mathcal{A}^*(\epsilon) = \arg\min\{L_\epsilon(V, \mathcal{A}) + \text{Overhead}(\mathcal{A})\}.$$
3. For mixture subspace model
   - Model V_i as a (degenerate) Gaussian model
$$\textbf{Bit rate: } R(V_i) = \frac{1}{2}\log_2\det(I + \frac{D}{\epsilon^2 N_i} V_i V_i^T).$$
   - **Coding length** for V_i of N_i samples
$$L(V_i) = (N_i + D)R(V_i) + \frac{D}{2}\log_2\det(1 + \frac{1}{\epsilon^2}\mu_i\mu_i^T) + N_i(-\log_2(N_i/N)).$$

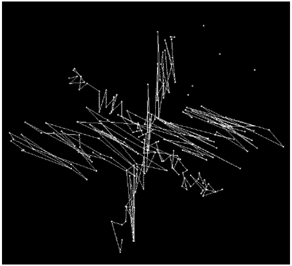
Lossy Minimum Description Length (LMDL)

1. **Lossy coding length** $L_\epsilon(V, \mathcal{A})$:
   Quantize $V = (v_1, \cdots, v_N) \in \mathbb{R}^{D \times N}$ as **a sequence of binary bits** up to a distortion $\mathbb{E}[\|v_i - \hat{v}_i\|^2] \leq \epsilon^2$.
2. **Lossy MDL**
$$\mathcal{A}^*(\epsilon) = \arg\min\{L_\epsilon(V, \mathcal{A}) + \text{Overhead}(\mathcal{A})\}.$$
3. For mixture subspace model
   - Model V_i as a (degenerate) Gaussian model
$$\textbf{Bit rate: } R(V_i) = \frac{1}{2}\log_2\det(I + \frac{D}{\epsilon^2 N_i} V_i V_i^T).$$
   - **Coding length** for V_i of N_i samples
$$L(V_i) = (N_i + D)R(V_i) + \frac{D}{2}\log_2\det(1 + \frac{1}{\epsilon^2}\mu_i\mu_i^T) + N_i(-\log_2(N_i/N)).$$
   - **Total coding length**: $L^s(V_1, \cdots, V_K) = \sum_i L(V_i)$.

A Greedy Optimization

1. **Initialize**: Assume N samples as individual groups.
2. Each iteration: Merge two groups that reduces largest coding length.
3. To stop: If any further merging cannot reduces L^s.
4. **Output**: Estimation of K and the grouping.

animation



Simulation

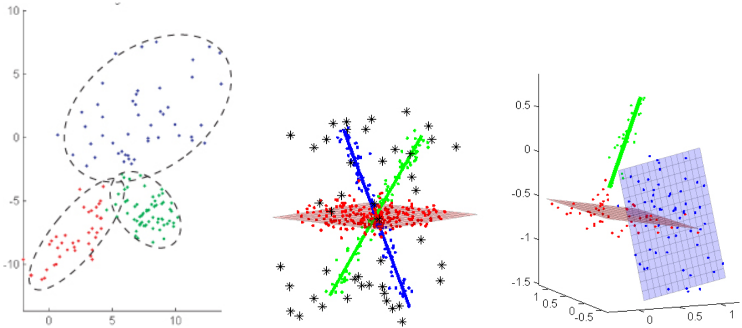
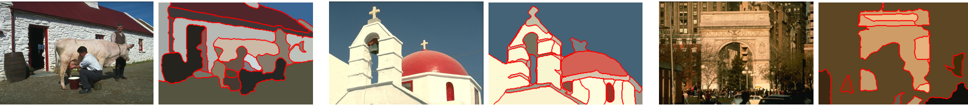


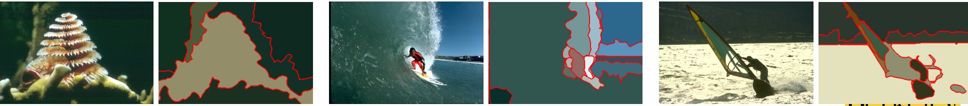
Image Segmentation via Mixture Subspace Models

(g) Nature



(h) Urban

(i) Portraits



(j) Water

Quantitative Comparison

Table: Average performance on the Berkeley image segmentation database.

	PRI	VoI	GCE	BDE
Humans	0.8754	1.1040	0.0797	4.994
CTM$_{\gamma=0.1}$	0.7561	**2.4640**	**0.1767**	**9.4211**
Mean-Shift [Comaniciu 2002]	0.7550	2.477	0.2598	9.7001
N-Cuts [Shi 2000]	0.7229	2.9329	0.2182	9.6038
F-H [Felzenszwalb 2004]	**0.7841**	2.6647	0.1895	9.9497

PRI: Probabilistic Rand Index [Pantofaru 2005].
VoI: Variation of Information [Meila 2005].
GCE: Global Consistency Error [Martin 2001].
BDE: Boundary Displacement Error [Freixenet 2002].

Reference:

Unsupervised Segmentation of Natural Images via Lossy Data Compression, *CVIU*, 2008.

GPCA Website: `http://perception.csl.uiuc.edu/gpca/`

Generalized Principal Component Analysis

Welcome

Introduction

Sample Code

Applications

Publications

Sponsors

JHU

Berkeley

NSF

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.

Website Credits

This site is jointly developed and maintained by the research groups of

- Professor Yi Ma of the Electrical & Computer Engineering Department at the University of Illinois at Urbana-Champaign
- Professor Rene Vidal of the Biomedical Engineering Department at the Johns Hopkins University
- Professor Kun Huang of the Biomedical Informatics Department at the Ohio State University

www.eecs.berkeley.edu/~yang/software/lossy_segmentation/

Unsupervised Segmentation of Natural Images via Lossy Data Compression

Allen Y. Yang, John Wright, Yi Ma, and Shankar Sastry

ABSTRACT:

We cast natural-image segmentation as a problem of clustering texure features as multivariate mixed data. We model the distribution of the texture features using a mixture of Gaussian distributions. Unlike most existing clustering methods, we allow the mixture components to be degenerate or nearly-degenerate. We contend that this assumption is particularly important for mid-level image segmentation, where degeneracy is typically introduced by using a common feature representation for different textures in an image. We show that such a mixture distribution can be effectively segmented by a simple agglomerative clustering algorithm derived from a lossy data compression approach. Using either 2D texture filter banks or simple fixed-size windows as texture features, the algorithm effectively segments an image by minimizing the overall coding length of the feature vectors. We conduct comprehensive experiments to measure the performance of the algorithm in terms of visual evaluation and a variety of quantitative indices for image segmentation. The algorithm compares favorably against other well-known image segmentation methods on the Berkeley image database.

- **Publications:**

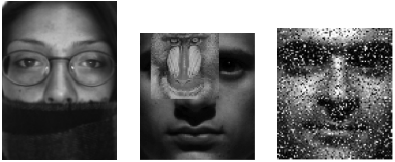
  Allen Y. Yang, John Wright, Yi Ma, and Shankar Sastry. *Unsupervised segmentation of natural images via lossy data compression*. To appear in CVIU 2007. [PDF]

Classification of Mixture Subspaces

- **Notation**
  - Training: For K classes, collect training samples $\{\mathbf{v}_{1,1}, \cdots, \mathbf{v}_{1,n_1}\}, \cdots, \{\mathbf{v}_{K,1}, \cdots, \mathbf{v}_{K,n_K}\} \in \mathbb{R}^D$.
  - Test: Present a new $\mathbf{y} \in \mathbb{R}^D$, solve for label$(\mathbf{y}) \in [1, 2, \cdots, K]$.

Classification of Mixture Subspaces

- **Notation**
  - Training: For K classes, collect training samples $\{\mathbf{v}_{1,1}, \cdots, \mathbf{v}_{1,n_1}\}, \cdots, \{\mathbf{v}_{K,1}, \cdots, \mathbf{v}_{K,n_K}\} \in \mathbb{R}^D$.
  - Test: Present a new $\mathbf{y} \in \mathbb{R}^D$, solve for label$(\mathbf{y}) \in [1, 2, \cdots, K]$.
- **Facial disguise & occlusion**



Sparse Representation

Sparsity

A signal is sparse if most of its coefficients are (approximately) zero.

Sparse Representation

Sparsity

A signal is sparse if most of its coefficients are (approximately) zero.

1. Sparsity in frequency domain

Figure: 2-D DCT transform.

2. Sparsity in spatial domain

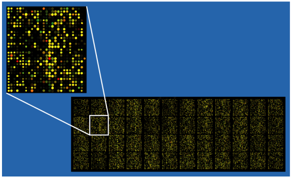
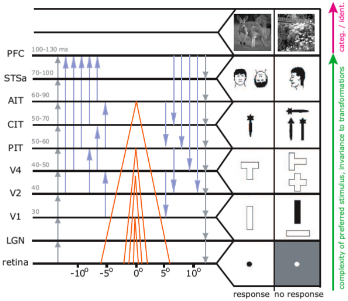
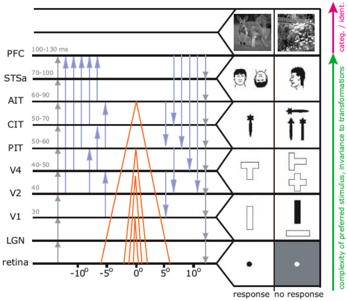


Figure: Gene microarray data.

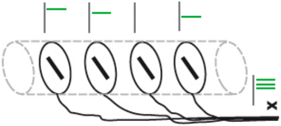
- **Sparsity in human visual cortex** [Perrett & Oram 1993, Olshausen & Field 1997, Riesenhuber & Poggio 2000]



- **Sparsity in human visual cortex** [Perrett & Oram 1993, Olshausen & Field 1997, Riesenhuber & Poggio 2000]

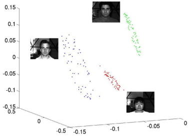


1. **Feed-forward**: No iterative feedback loop.
2. **Redundancy**: Average 80-200 neurons for each feature representation.
3. **Recognition**: Information exchange between stages is not about individual neurons, but rather how many neurons as a group fire together.



Classification of Mixture Subspace Model

1. Face-subspace model: Assume $\mathbf{y}$ belongs to Class i

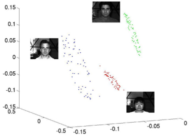


$$\begin{aligned} \mathbf{y} &= \alpha_{i,1}\mathbf{v}_{i,1} + \alpha_{i,2}\mathbf{v}_{i,2} + \cdots + \alpha_{i,n_1}\mathbf{v}_{i,n_i}, \\ &= A_i \alpha_i, \end{aligned}$$

where $A_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \cdots, \mathbf{v}_{i,n_i}]$.

Classification of Mixture Subspace Model

1. Face-subspace model: Assume **y** belongs to Class i



$$\begin{aligned}\mathbf{y} &= \alpha_{i,1}\mathbf{v}_{i,1} + \alpha_{i,2}\mathbf{v}_{i,2} + \cdots + \alpha_{i,n_1}\mathbf{v}_{i,n_i},\\ &= A_i\alpha_i,\end{aligned}$$

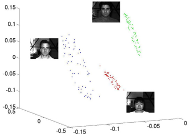
where $A_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \cdots, \mathbf{v}_{i,n_i}]$.

2. Nevertheless, Class i is the **unknown** variable we need to solve:

Sparse representation $\quad \mathbf{y} = [A_1, A_2, \cdots, A_K]\begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_K\end{bmatrix} = A\mathbf{x}.$

Classification of Mixture Subspace Model

1. Face-subspace model: Assume $\mathbf{y}$ belongs to Class i



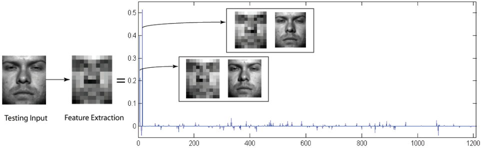
$$\begin{aligned} \mathbf{y} &= \alpha_{i,1}\mathbf{v}_{i,1} + \alpha_{i,2}\mathbf{v}_{i,2} + \cdots + \alpha_{i,n_1}\mathbf{v}_{i,n_i}, \\ &= A_i \alpha_i, \end{aligned}$$

where $A_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \cdots, \mathbf{v}_{i,n_i}]$.

2. Nevertheless, Class i is the **unknown** variable we need to solve:

Sparse representation $\mathbf{y} = [A_1, A_2, \cdots, A_K] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_K \end{bmatrix} = A\mathbf{x}.$

3. $\mathbf{x}_0 = [\, 0 \cdots 0 \ \alpha_i^T \ 0 \cdots 0 \,]^T \in \mathbb{R}^n.$



Sparse representation $\mathbf{x}_0$ encodes membership!

ℓ^1-Minimization

1. **Ideal solution:** ℓ^0-Minimization

$$(P_0) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \mathbf{y} = A\mathbf{x}.$$

$\|\cdot\|_0$ simply counts the number of nonzero terms.
However, generally ℓ^0-minimization is *NP-hard*.

ℓ^1-Minimization

1. **Ideal solution:** ℓ^0-Minimization

$$(P_0) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \mathbf{y} = A\mathbf{x}.$$

$\|\cdot\|_0$ simply counts the number of nonzero terms.
However, generally ℓ^0-minimization is *NP-hard*.

2. **Compressive Sensing**: Under mild condition, ℓ^0-minimization is equivalent to

$$(P_1) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = A\mathbf{x},$$

where $\|\mathbf{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$.

ℓ^1-Minimization

1. **Ideal solution:** ℓ^0-Minimization

$$(P_0) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \mathbf{y} = A\mathbf{x}.$$

   $\|\cdot\|_0$ simply counts the number of nonzero terms.
   However, generally ℓ^0-minimization is *NP-hard*.

2. **Compressive Sensing**: Under mild condition, ℓ^0-minimization is equivalent to

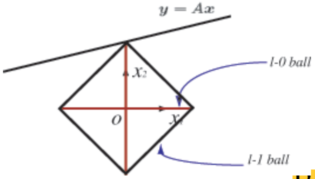
$$(P_1) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = A\mathbf{x},$$

   where $\|\mathbf{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$.

3. ℓ^1-Ball

ℓ^0/ℓ^1 Equivalence

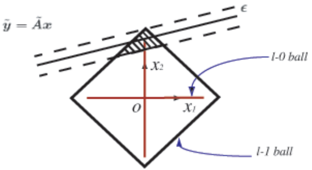
- ℓ^1-Minimization is convex.
- Solution equal to ℓ^0-minimization.



Stability of ℓ^1-Minimization

- ℓ^1 near solution

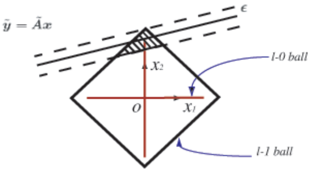
$$\mathbf{y} = A\mathbf{x} + \mathbf{e} \quad \text{s.t. } \|e\|_2 < \epsilon.$$



Stability of ℓ^1-Minimization

- ℓ^1 near solution

$$\mathbf{y} = A\mathbf{x} + \mathbf{e} \quad \text{s.t. } \|e\|_2 < \epsilon.$$



- **Bounded noise produces bounded ℓ^1 solution**

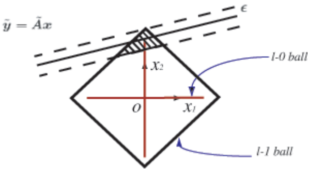
$$(P_1') \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \|\mathbf{y} - A\mathbf{x}\|_2 < \epsilon.$$

Restricted Isometry Property [Candès, Romberg, Tao 2004]: $\|\mathbf{x}^* - \mathbf{x}_0\|_2 < C\epsilon$.

Stability of ℓ^1-Minimization

- ℓ^1 near solution

$$\mathbf{y} = A\mathbf{x} + \mathbf{e} \quad \text{s.t. } \|e\|_2 < \epsilon.$$



- **Bounded noise produces bounded ℓ^1 solution**

$$(P_1') \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \|\mathbf{y} - A\mathbf{x}\|_2 < \epsilon.$$

Restricted Isometry Property [Candès, Romberg, Tao 2004]: $\|\mathbf{x}^* - \mathbf{x}_0\|_2 < C\epsilon$.

- ℓ^1-minimization routines
  1. **Matching pursuit** [Mallat 1993]
  2. **Basis pursuit** [Chen 1998]
  3. **Lasso** [Tibshirani 1996]

Partial Features on Extended Yale B Database

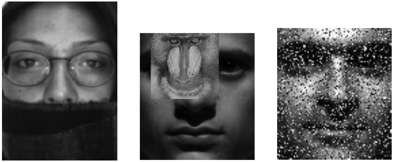
Features	Nose	Right Eye	Mouth & Chin
Dimension	4,270	5,040	12,936
SRC [%]	87.3	93.7	98.3
nearest-neighbor [%]	49.2	68.8	72.7
nearest-subspace [%]	83.7	78.6	94.4
Linear SVM [%]	70.8	85.8	95.3

SRC: sparse-representation classifier

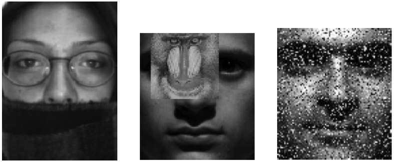
Reference:

Robust face recognition via sparse representation, **(in press) PAMI**, 2008.

Occlusion Compensation

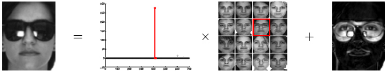


Occlusion Compensation



1. Sparse representation + sparse error

$$\mathbf{y} = A\mathbf{x} + \mathbf{e}$$



2. Occlusion compensation

$$\mathbf{y} = \begin{bmatrix} A & \mid & I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} = B\mathbf{w}$$

AR Database: 100 subjects, illumination, expression, occlusion

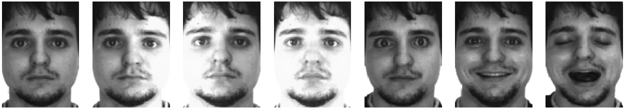
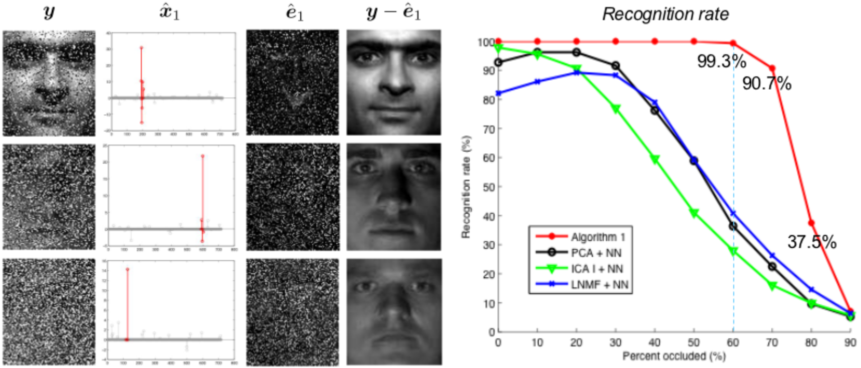


Figure: Training samples for Subject 1.

illumination & expression	sunglasses	scarves
		
95%	97.5%	93.5%

Random Pixel Corruption



Summary: Sparse Representation

1. Curse of dimensionality becomes **blessing of dimensionality**.

Summary: Sparse Representation

1. Curse of dimensionality becomes **blessing of dimensionality**.
2. Retains characteristics of human perception
    - Feed-forward: noniterative, no model parameters to train.
    - Representation is redundant: simultaneously solve for **classification** and **occlusion compensation**.
    - Membership is encoded in the sparse representation.

Summary: Sparse Representation

1. Curse of dimensionality becomes **blessing of dimensionality**.
2. Retains characteristics of human perception
   - Feed-forward: noniterative, no model parameters to train.
   - Representation is redundant: simultaneously solve for **classification** and **occlusion compensation**.
   - Membership is encoded in the sparse representation.
3. Algorithm achieves state-of-the-art performance.

Acknowledgments

Collaborators

- **Berkeley**: Dr. Shankar Sastry, Dr. Ruzena Bajcsy
- **UIUC**: Dr. Yi Ma, Dr. Robert Fossum, John Wright, Shankar Rao
- **JHU**: Dr. René Vidal
- **UMich**: Dr. Harm Derksen

Acknowledgments

Collaborators

- **Berkeley**: Dr. Shankar Sastry, Dr. Ruzena Bajcsy
- **UIUC**: Dr. Yi Ma, Dr. Robert Fossum, John Wright, Shankar Rao
- **JHU**: Dr. René Vidal
- **UMich**: Dr. Harm Derksen

Funding Support

- ARO MURI: Heterogeneous Sensor Networks in Urban Terrains
- ARO MURI: Adaptive Coordinated Control of Intelligent Multi-Agent Teams
- NSF TRUST Center

Acknowledgments

Collaborators

- **Berkeley**: Dr. Shankar Sastry, Dr. Ruzena Bajcsy
- **UIUC**: Dr. Yi Ma, Dr. Robert Fossum, John Wright, Shankar Rao
- **JHU**: Dr. René Vidal
- **UMich**: Dr. Harm Derksen

Funding Support

- ARO MURI: Heterogeneous Sensor Networks in Urban Terrains
- ARO MURI: Adaptive Coordinated Control of Intelligent Multi-Agent Teams
- NSF TRUST Center

References

- **SIAM Review:** Estimation of Subspace Arrangements with Applications in Modeling and Segmenting Mixed Data, 2008.
- **IJCV (draft):** Robust Algebraic Segmentation of Mixed Rigid-Body and Planar Motions, 2008.
- **CVIU:** Unsupervised Segmentation of Natural Images via Lossy Data Compression, 2008.
- **PAMI:** Robust face recognition via sparse representation, 2009.

ℓ^1-Minimization Routines

- **Matching pursuit** [Mallat 1993]
  1. Find most correlated vector $\mathbf{v}_i$ in A with $\mathbf{y}$: $i = \arg\max \langle \mathbf{y}, \mathbf{v}_j \rangle$.
  2. $A \leftarrow A^{(i)}$, $x_i \leftarrow \langle \mathbf{y}, \mathbf{v}_i \rangle$, $\mathbf{y} \leftarrow \mathbf{y} - x_i \mathbf{v}_i$.
  3. Repeat until $\|\mathbf{y}\| < \epsilon$.
- **Basis pursuit** [Chen 1998]
  1. Start with number of sparse coefficients $m = 1$.
  2. Select m linearly independent vectors B_m in A as a basis
  $$\mathbf{x}_m = B_m^{\dagger} \mathbf{y}.$$
  3. Repeat swapping one basis vector in B_m with another vector not in B_m if improve $\|\mathbf{y} - B_m \mathbf{x}_m\|$.
  4. If $\|\mathbf{y} - B_m \mathbf{x}_m\|_2 < \epsilon$, stop; Otherwise, $m \leftarrow m + 1$, repeat Step 2.
- **Quadratic solvers**: $\mathbf{y} = A\mathbf{x}_0 + \mathbf{z} \in \mathbb{R}^d$, where $\|\mathbf{z}\|_2 < \epsilon$
$$\mathbf{x}^* = \arg\min\{\|\mathbf{x}\|_1 + \lambda \|\mathbf{y} - A\mathbf{x}\|_2\}$$
  [LASSO, Second-order cone programming]: Much more expensive.

Matlab Toolboxes for ℓ^1-Minimization

- **ℓ^1-Magic** by Candes
- **SparseLab** by Donoho
- **cvx** by Boyd

Mild Conditions for ℓ^1/ℓ^0 Equivalence

$$(P_1) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = A\mathbf{x}$$

Solve ℓ^1-minimization achieves the optimal sparse solution under the following conditions

Mild Conditions for ℓ^1/ℓ^0 Equivalence

$$(P_1) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = A\mathbf{x}$$

Solve ℓ^1-minimization achieves the optimal sparse solution under the following conditions

- **Short answer**: For most underdetermined systems A, such as random matrices, the equivalence holds

$$\text{Asmyptotically with } \frac{k \uparrow}{d \uparrow} < 0.5$$

Mild Conditions for ℓ^1/ℓ^0 Equivalence

$$(P_1) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = A\mathbf{x}$$

Solve ℓ^1-minimization achieves the optimal sparse solution under the following conditions

- **Short answer**: For most underdetermined systems A, such as random matrices, the equivalence holds

$$\text{Asmyptotically with } \frac{k\uparrow}{d\uparrow} < 0.5$$

- Long answers
  1. **(In)-coherence** [Gribvonel & Nielsen 2003, Donoho & Elad 2003]:

$$\mu(A,B) \doteq \sup_{\mathbf{a}\in A, \mathbf{b}\in B} \frac{|\langle \mathbf{a}, \mathbf{b}\rangle|}{\|\mathbf{a}\|\|\mathbf{b}\|}$$

$\|\mathbf{x}\|_0 \leq \frac{1}{2}(1 + \frac{1}{\mu(A,B)})$ suffices. A **and** B **have to be incoherent.**

Mild Conditions for ℓ^1/ℓ^0 Equivalence

$$(P_1) \quad \mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = A\mathbf{x}$$

Solve ℓ^1-minimization achieves the optimal sparse solution under the following conditions

- **Short answer**: For most underdetermined systems A, such as random matrices, the equivalence holds

$$\text{Asmyptotically with } \frac{k\uparrow}{d\uparrow} < 0.5$$

- Long answers
  1. **(In)-coherence** [Gribvonel & Nielsen 2003, Donoho & Elad 2003]:

$$\mu(A,B) \doteq \sup_{\mathbf{a}\in A, \mathbf{b}\in B} \frac{|\langle \mathbf{a}, \mathbf{b}\rangle|}{\|\mathbf{a}\|\|\mathbf{b}\|}$$

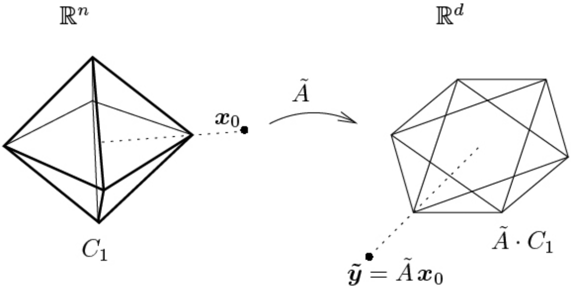
     $\|\mathbf{x}\|_0 \leq \frac{1}{2}(1 + \frac{1}{\mu(A,B)})$ suffices. **A and B have to be incoherent.**
  2. **Restricted Isometry** [Candes & Tao 2005]:
     Define $\delta_k(A) \doteq min\delta$ such that

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \|A\mathbf{x}\|_2^2 \leq (1+\delta)\|\mathbf{x}\|_2^2 \quad \forall k\text{-sparse } \mathbf{x}.$$

     $\delta_{2k}(A) \leq \sqrt{2} - 1$ suffices. **The columns of A should be uniformly well-spread.**

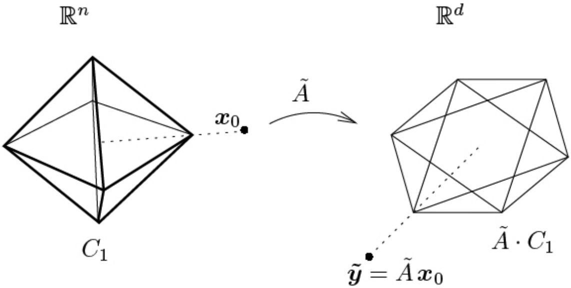
Berkeley
University of California

k-Neighborlyness [Donoho 2006]



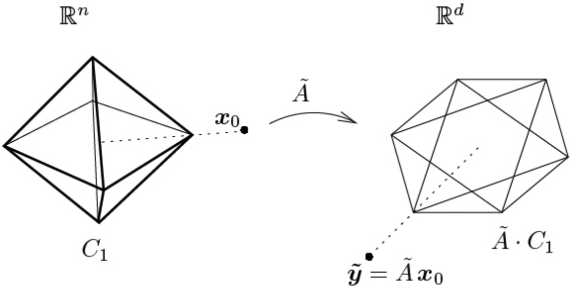
- Define **cross polytope** C and **quotient polytope** P such that $P = AC$.

k-Neighborlyness [Donoho 2006]



- Define **cross polytope** C and **quotient polytope** P such that $P = AC$.
- If $\mathbf{x}$ is k-sparse, $\mathbf{x}$ lie in a $(k-1)$-face of C in $\mathbb{R}^n$.

k-Neighborlyness [Donoho 2006]



- Define **cross polytope** C and **quotient polytope** P such that $P = AC$.
- If $\mathbf{x}$ is k-sparse, $\mathbf{x}$ lie in a $(k-1)$-face of C in $\mathbb{R}^n$.
- **Necessary and Sufficient:** If ℓ^1/ℓ^0 holds for all k-sparse $\mathbf{x}$, all $(k-1)$-faces of C must be the faces of P on the boundary.