Spectral Methods for Multi-Scale Feature Extraction and Data Clustering

Chakra Chennubhotla

University of Pittsburgh

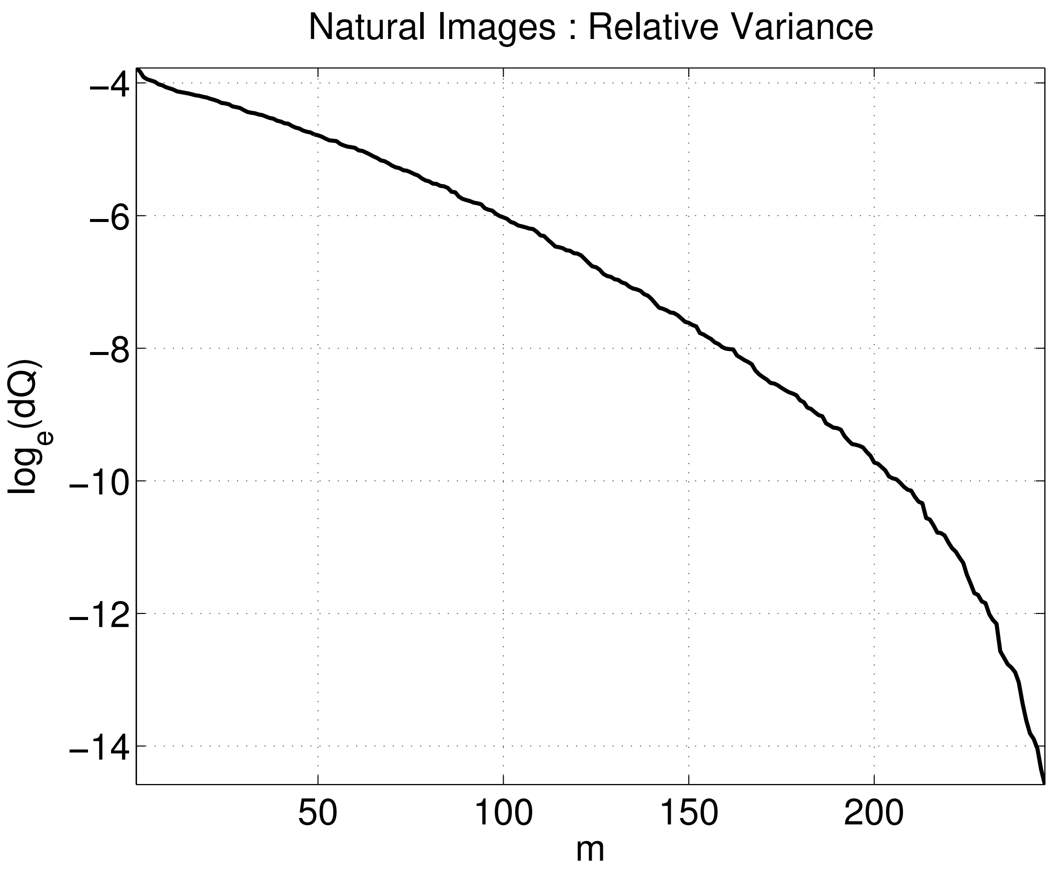
Image Understanding

http://aris.ss.uci.edu/cogsci/personnel/hoffman/cb.html

Representation

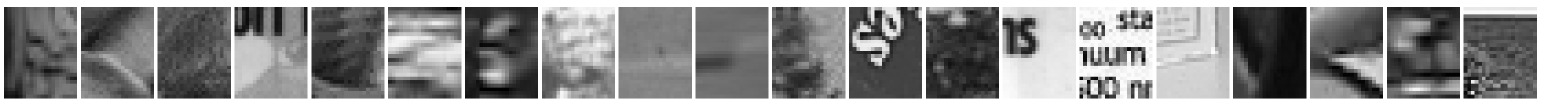
Power-Law Statistics

$P(f) \propto f^{-k}$

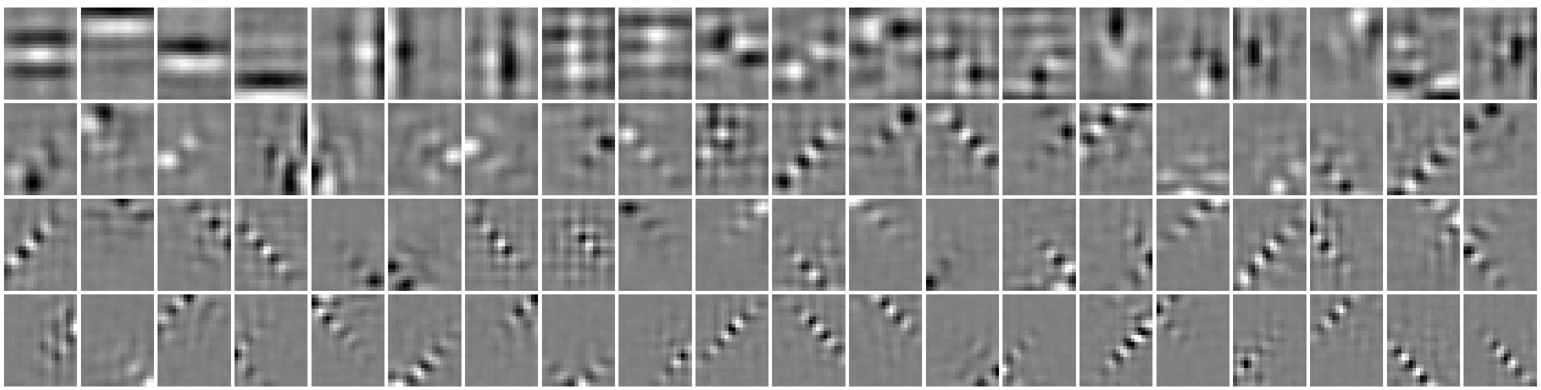


Building Multi-Scale Representation

Generic Image Patches



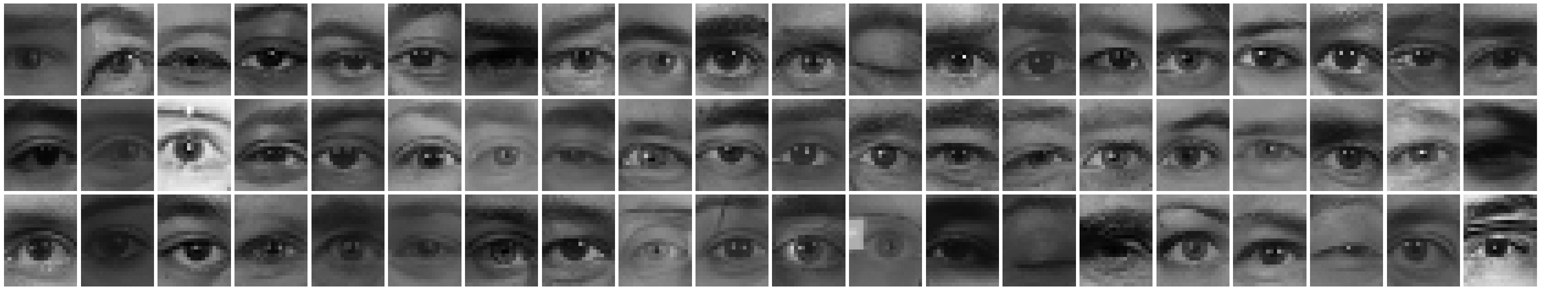
Sparse Coding/ICA/Wavelets



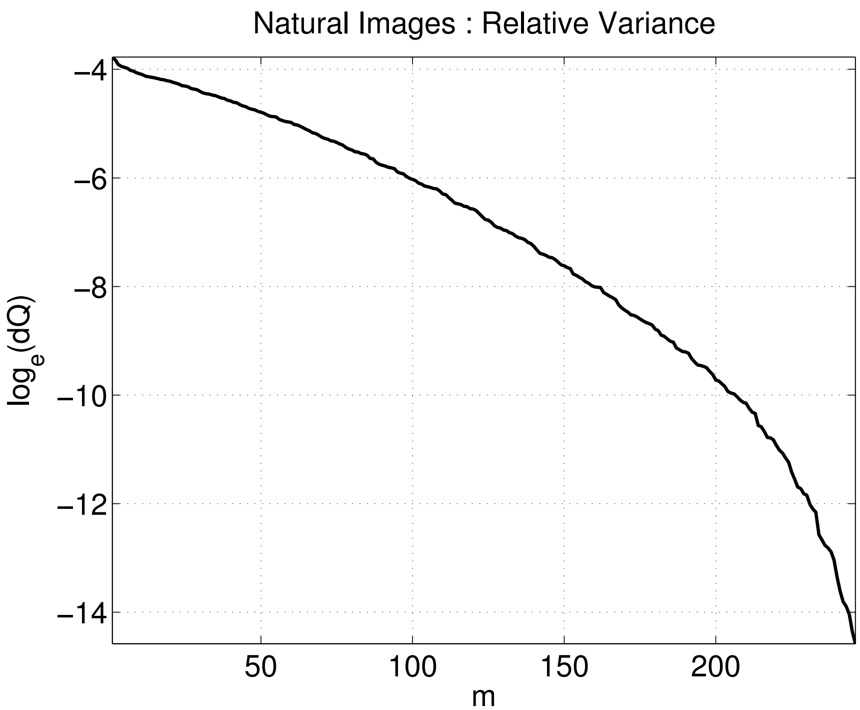
Object-Specific Datasets

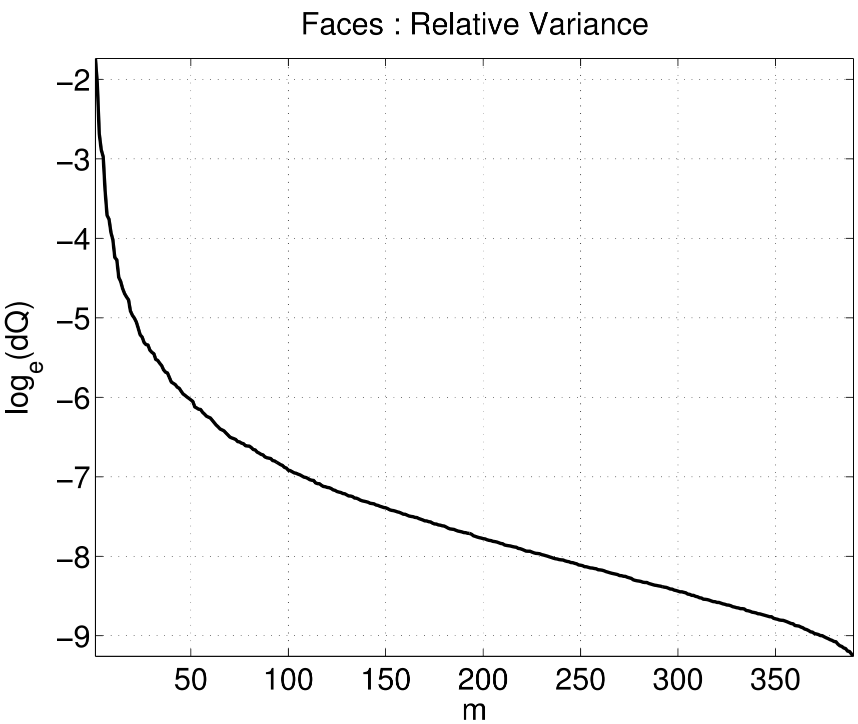
Faces

Eyes



Object-Specific Datasets





Generic Images

Faces

Multi-Scale Representation for Object-Specific Datasets

No Power-Law Statistics but..

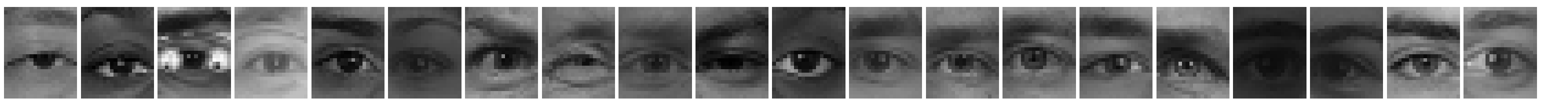
- Coarse-scale structure $\Rightarrow$ spatially distributed
- Fine-scale structure $\Rightarrow$ spatially localized

1. Sparse-PCA: Intuition

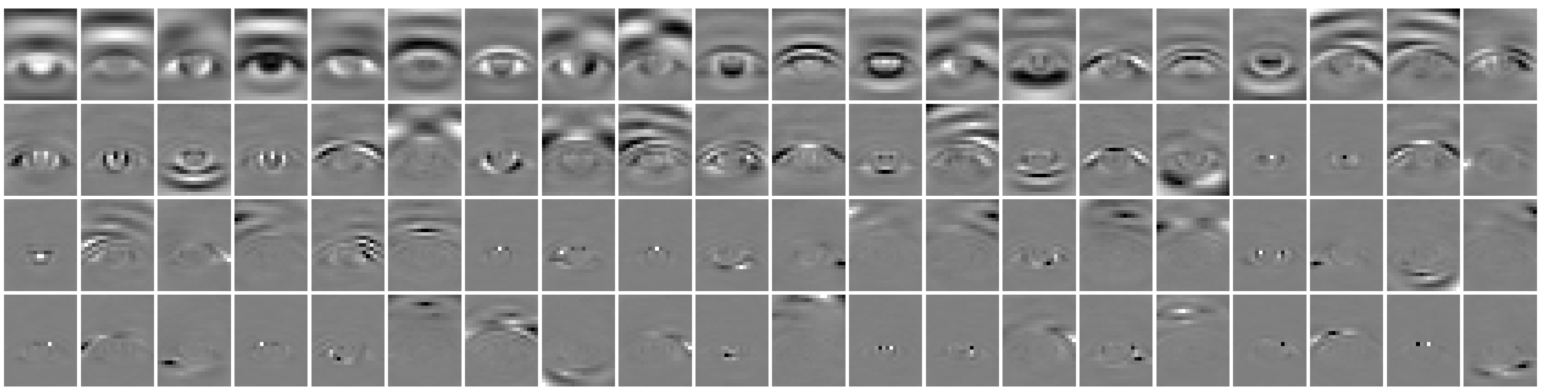
Promote Sparsity $\Rightarrow$ rotate basis directions

1. Sparse Principal Component Analysis

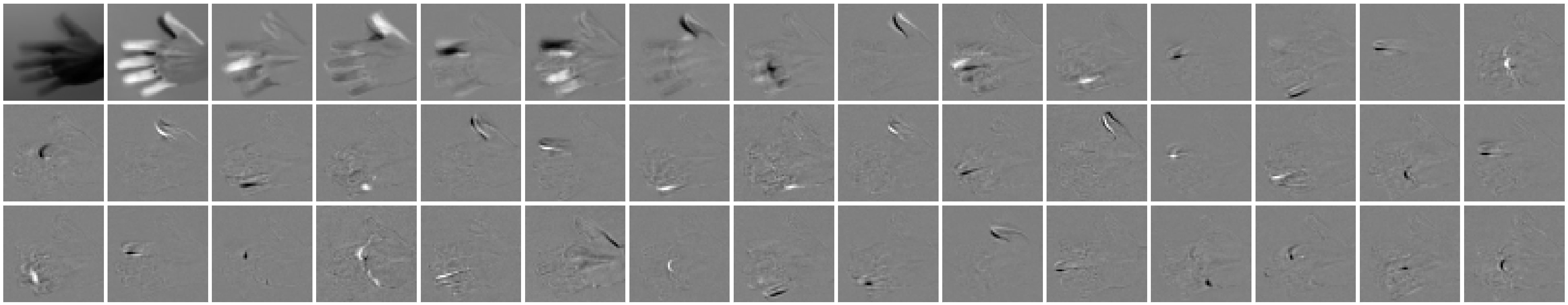
Eyes



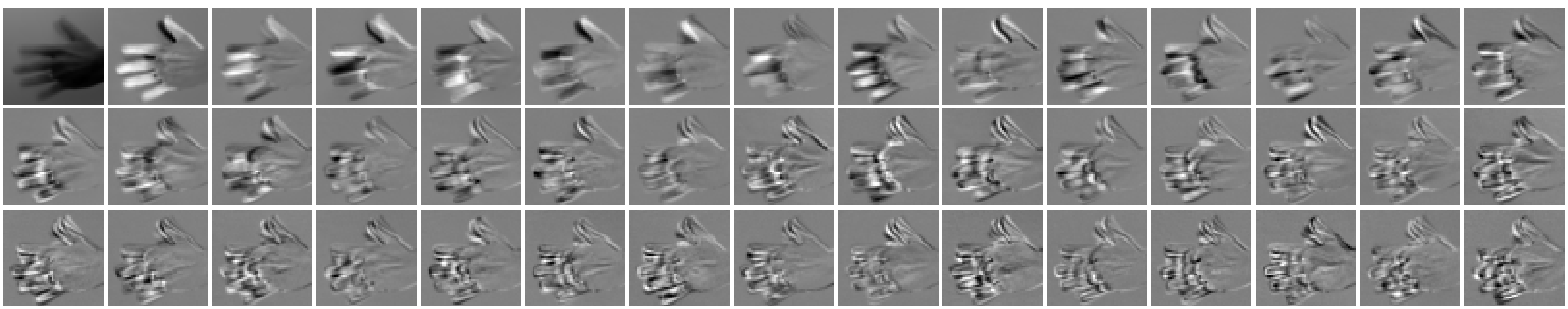
Sparse-PCA Basis



S-PCA

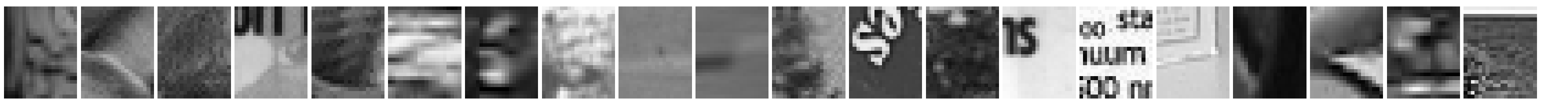


PCA

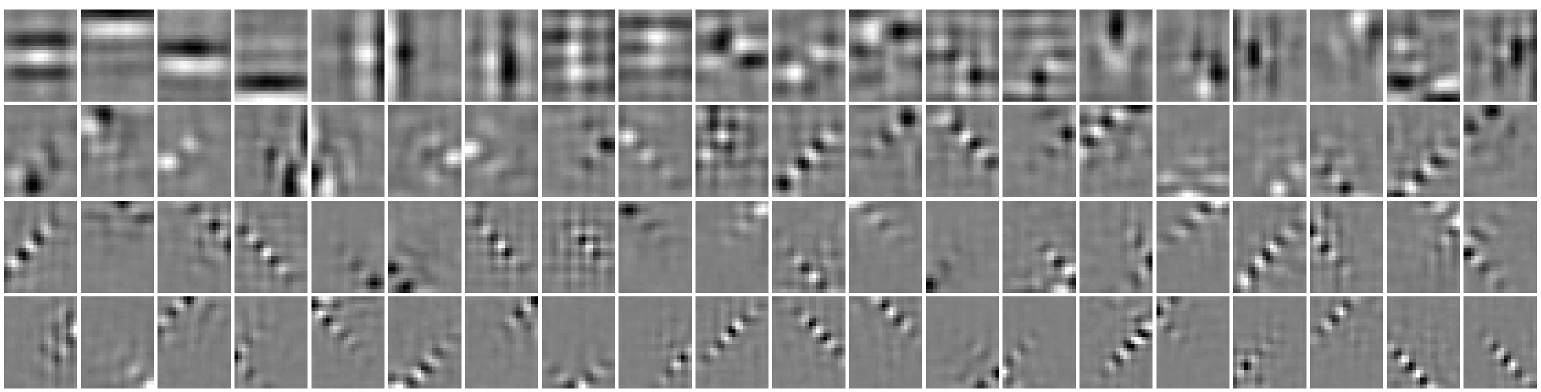


1. Sparse Principal Component Analysis

Generic Image Patches

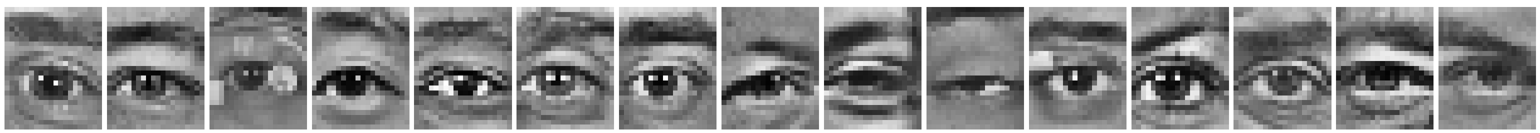


Sparse-PCA Basis

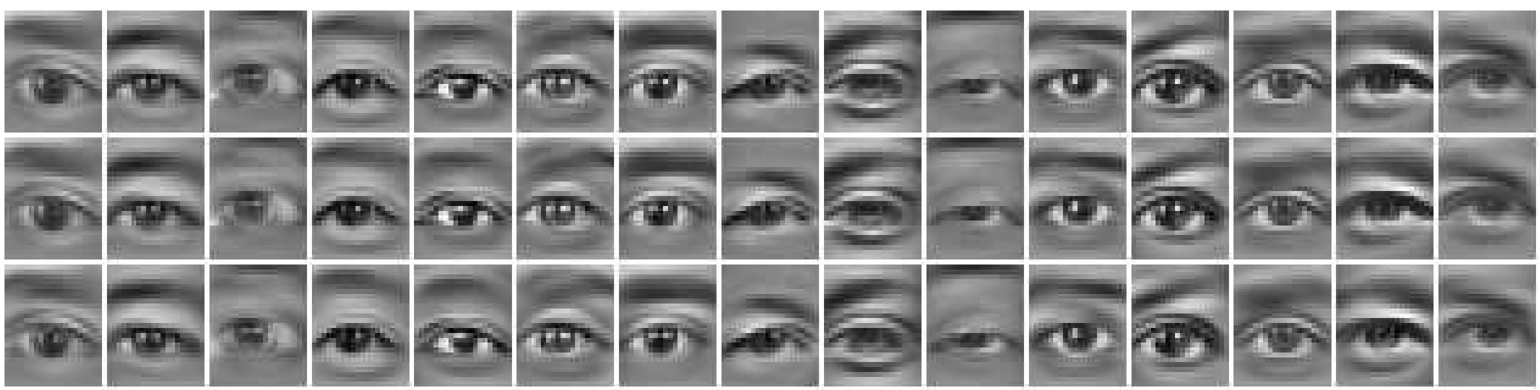


2. Iterative Image Reconstruction (S-PCA)

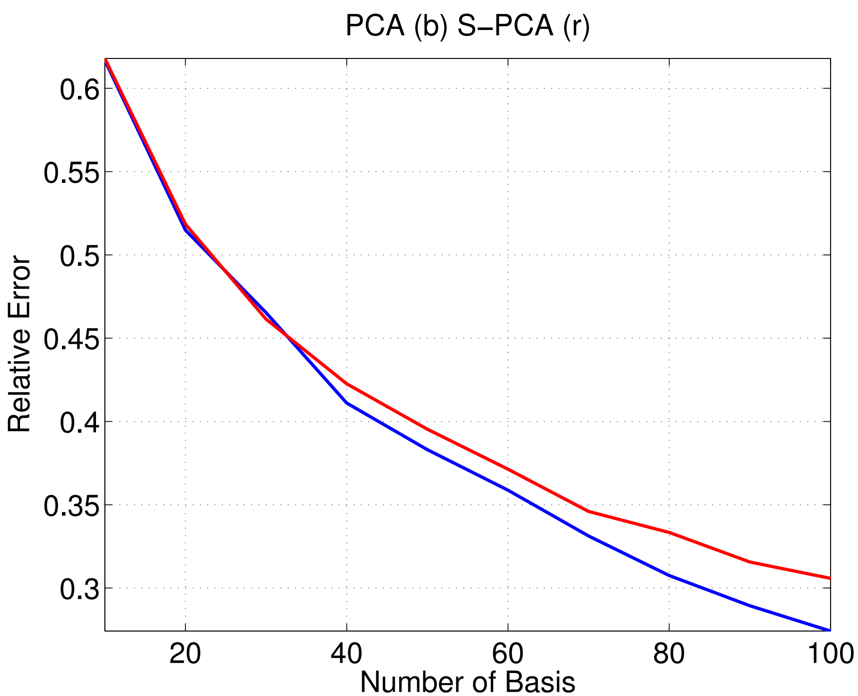
Input

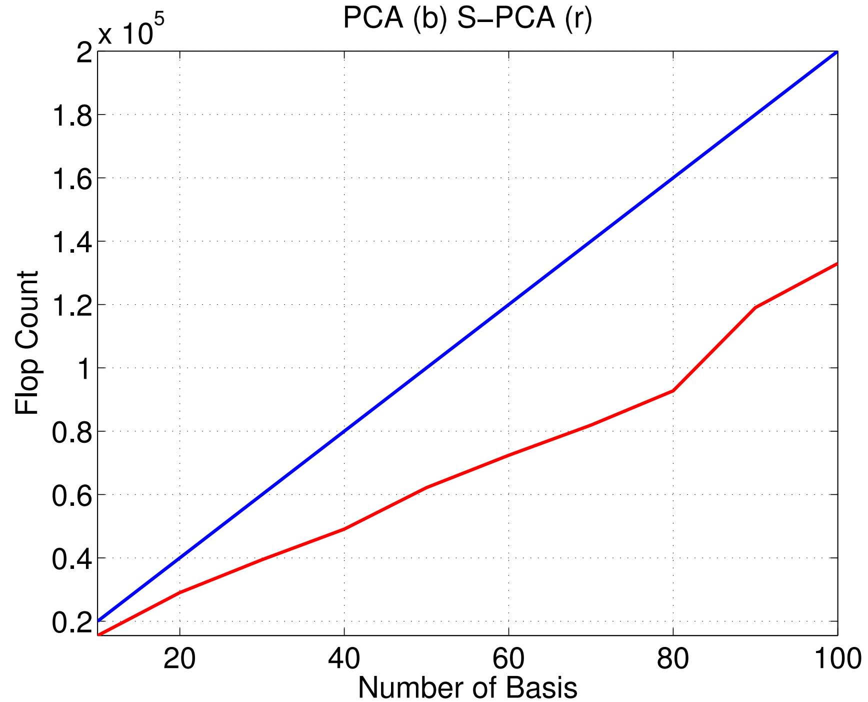


Reconstruction



2. Iterative Image Reconstruction



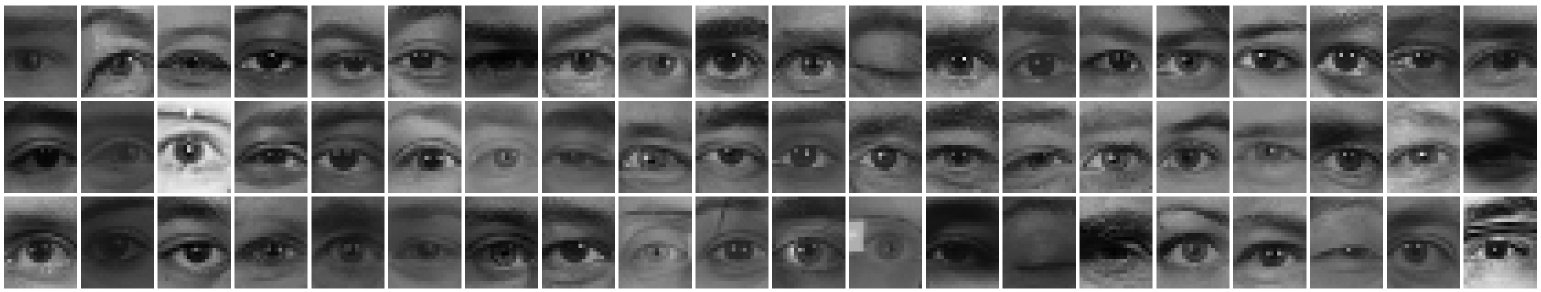


Reconstruction Error

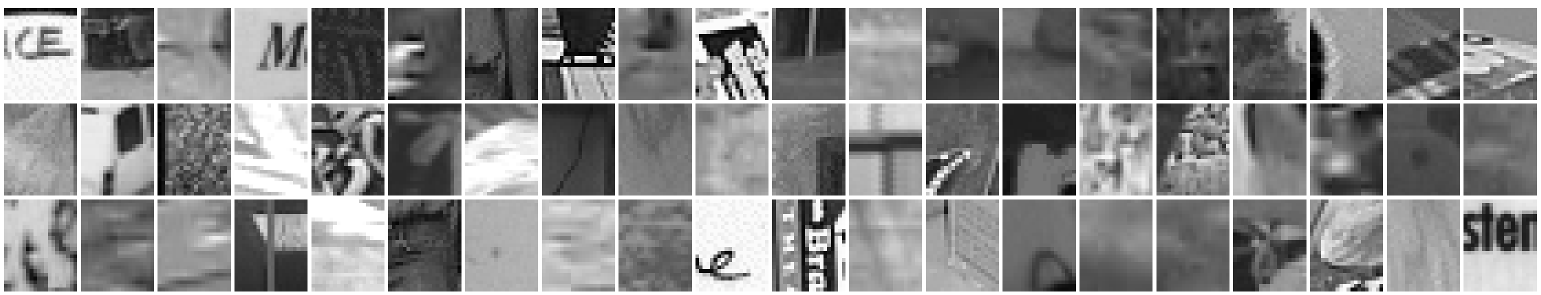
Flops

3. Contrast-Invariant Detection

Eyes



Generic Image Patches



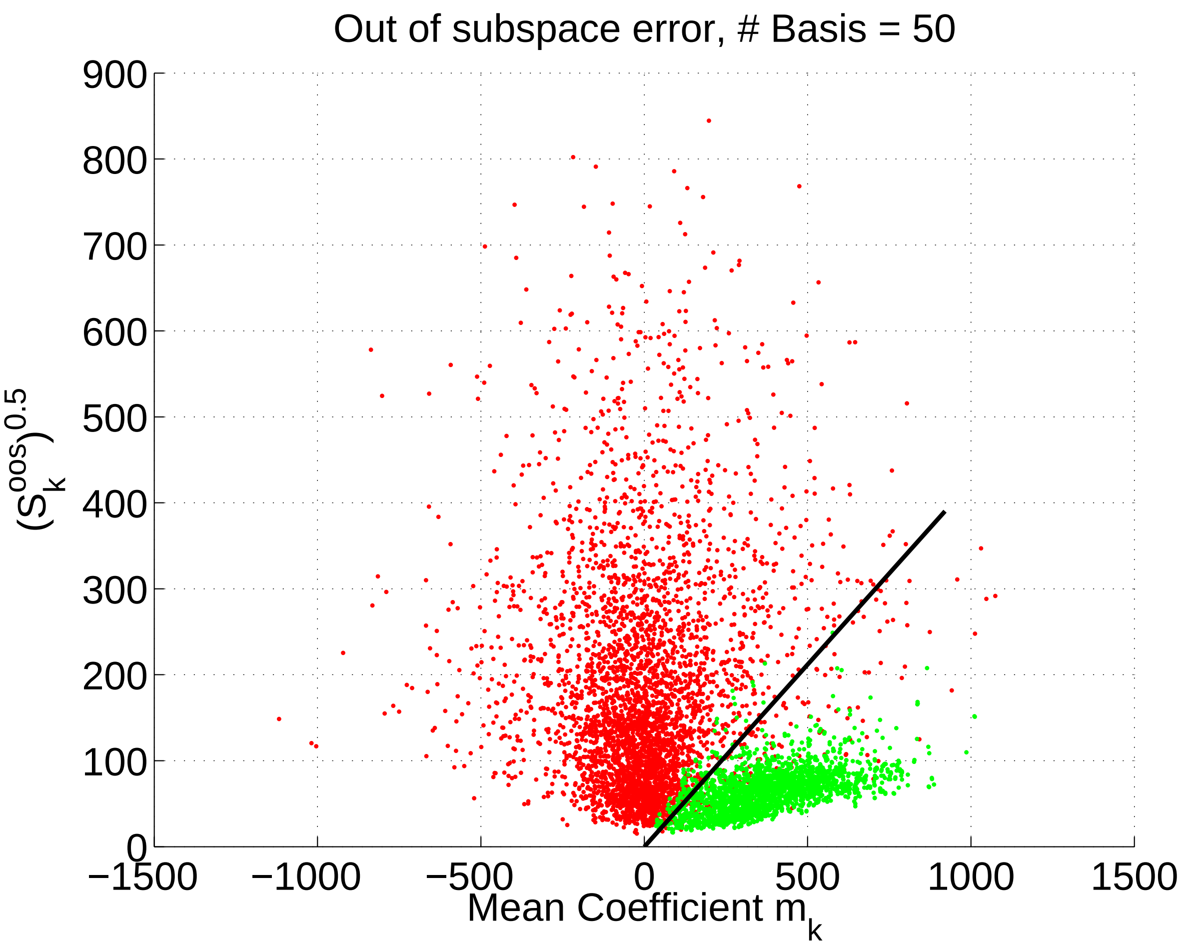
3. Detector Model–I [ChakraJepson, ICPR'02]

Residual = Original − Low-Dim Reconstruction

$$\underset{\text{Residual}}{\vec{e}} = \underset{\text{Image}}{\vec{t}} - \left[\underset{\text{DC}}{d} \times \vec{\xi} + \underset{\text{Mean}}{m} \times \vec{\mu} + \underbrace{\sum_{j=1}^{M} c_j \times \vec{b}_j}_{\text{Eye Basis}} \right]$$

$$S^{\text{oos}} = \frac{1}{N} \sum_{i=1}^{N} \frac{e_k^2(i)}{v_M(i)}$$

3. Detector Model–I: Eye Results

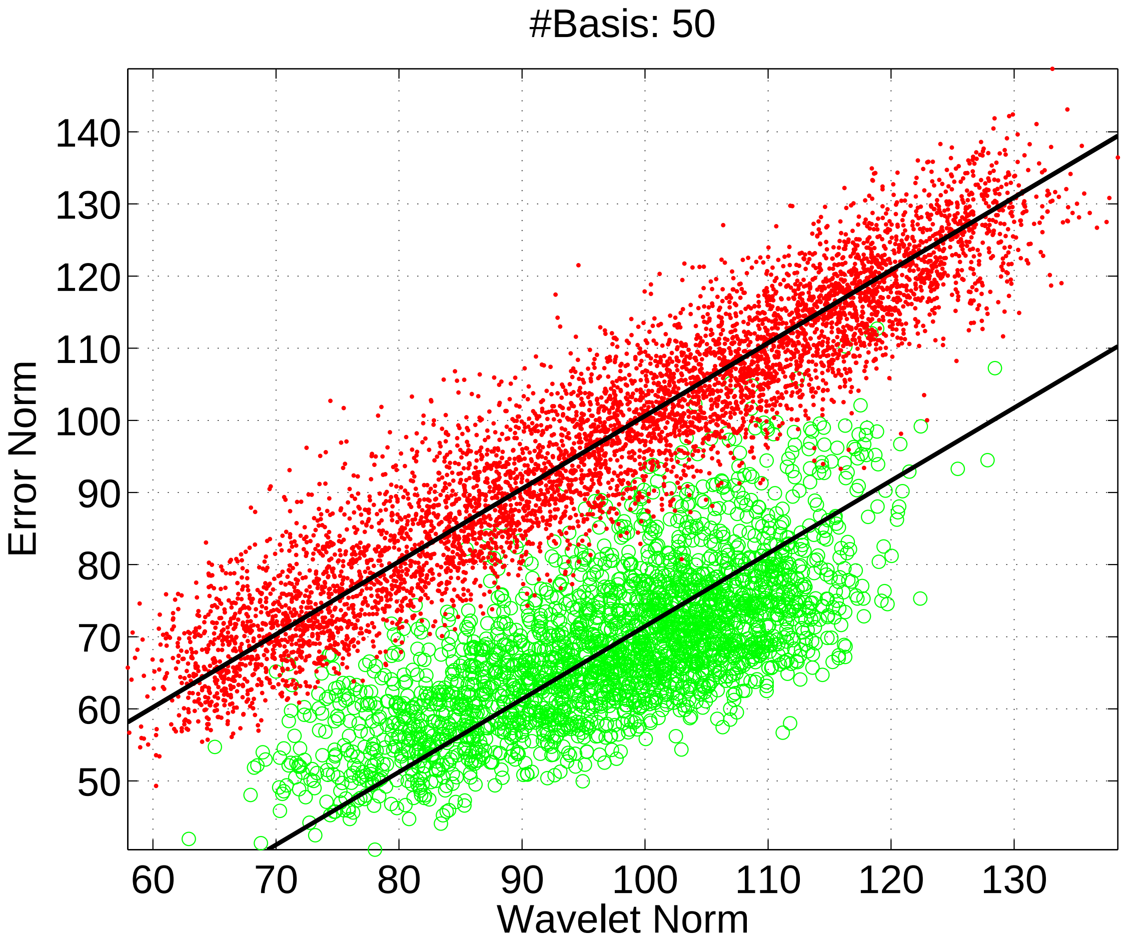


3. Perceptual-Distance Based Detector

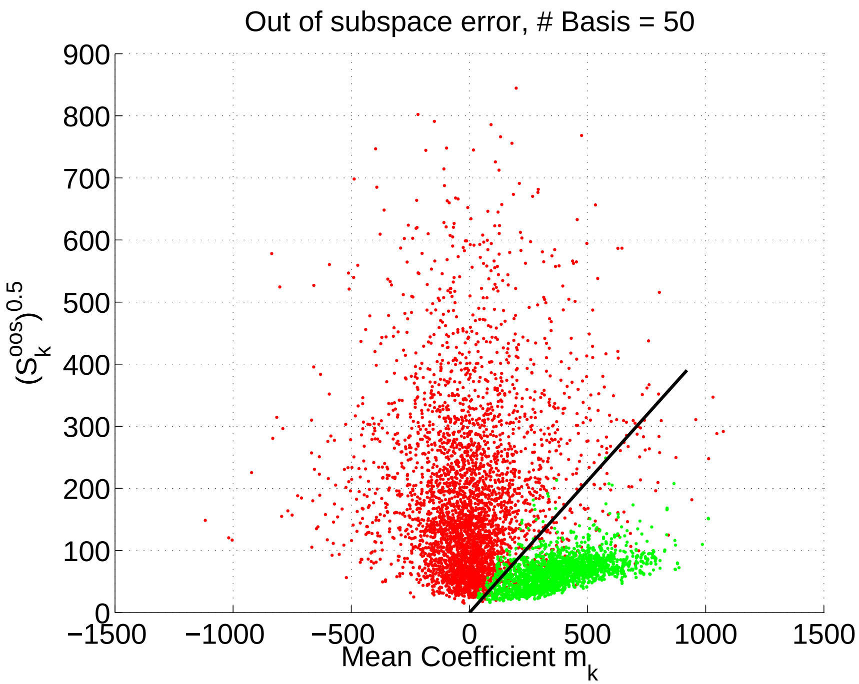
1. $\mathcal{WCN}$: Weber-Contrast Normalize
2. W: Representation for generic images
3. B: Representation for object-specific images
4. $\mathcal{PDN}$: Perceptual Distance Normalize

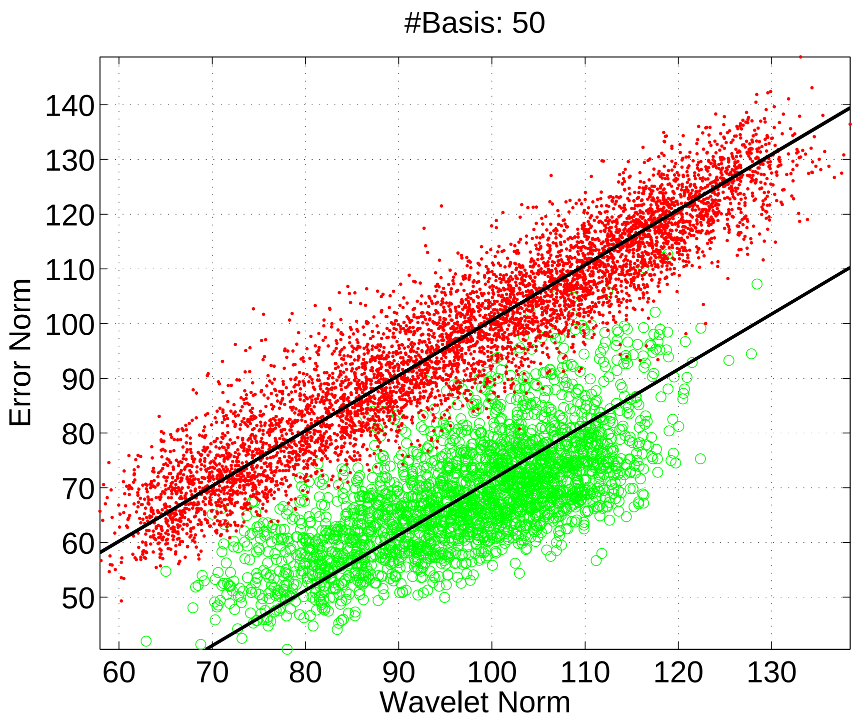
$$\begin{tikzcd}
\vec{t} \arrow[d, "W^T"] & \vec{x} \arrow[l, "\mathcal{WCN}"'] & \\
\vec{d} \arrow[r, "B^T"] \arrow[d, "\mathcal{PDN}"] & \vec{b} \arrow[r, "B"] & \vec{\hat{d}} \arrow[d, "\mathcal{PDN}"'] \\
\vec{z} & & \vec{\hat{z}}
\end{tikzcd}$$

3. Perceptual Detector: Eyes

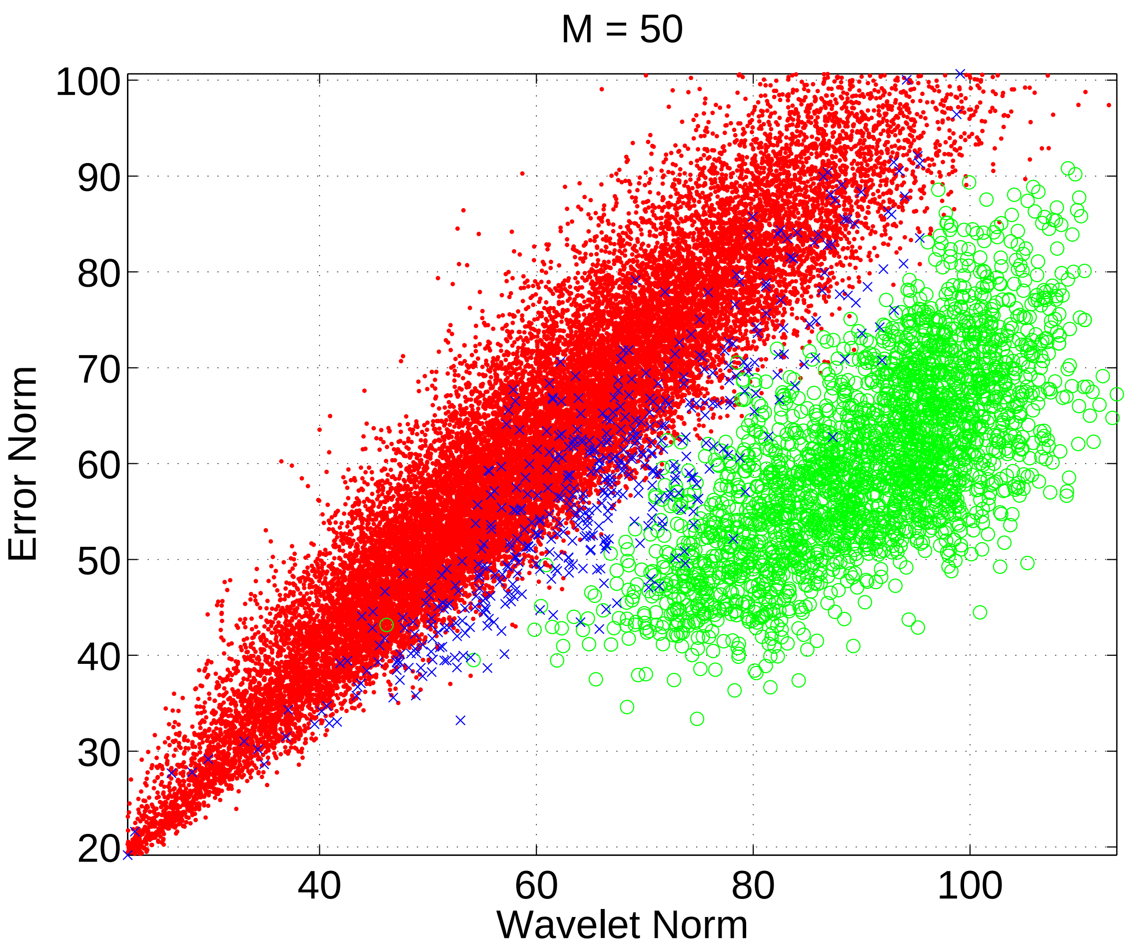


3. Perceptual Detector: Eyes

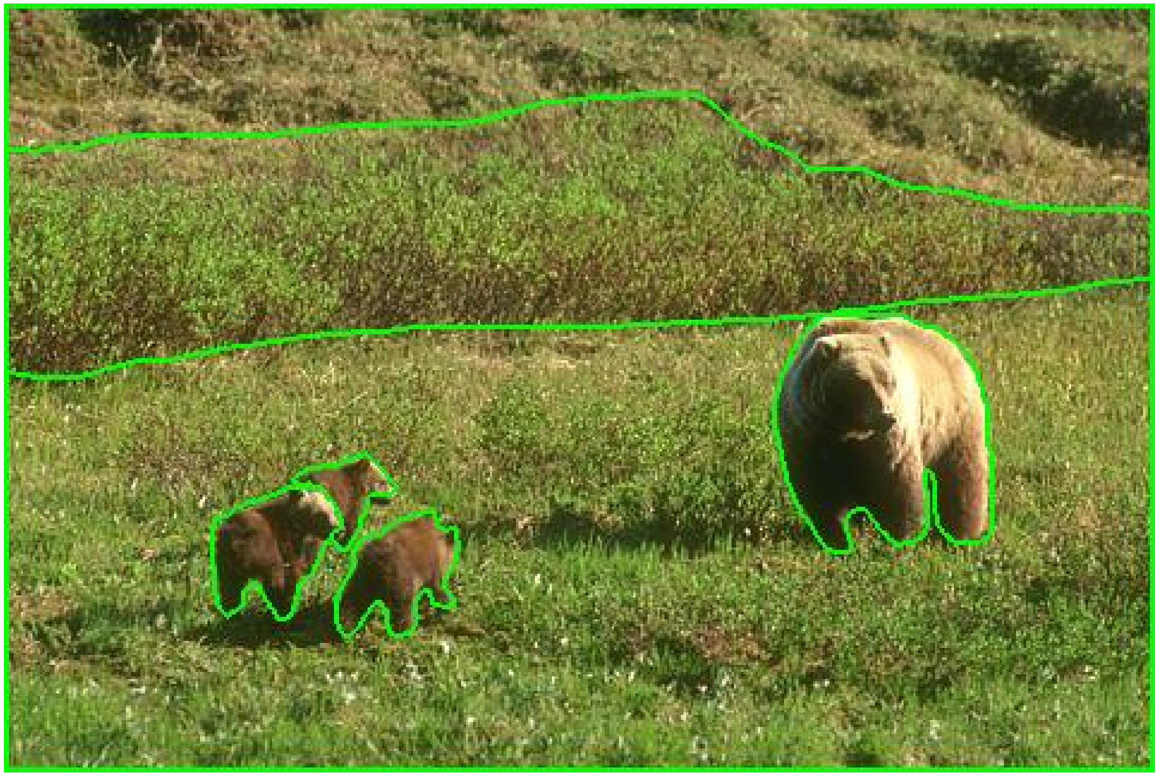




3. Perceptual Detector: MIT Face Data

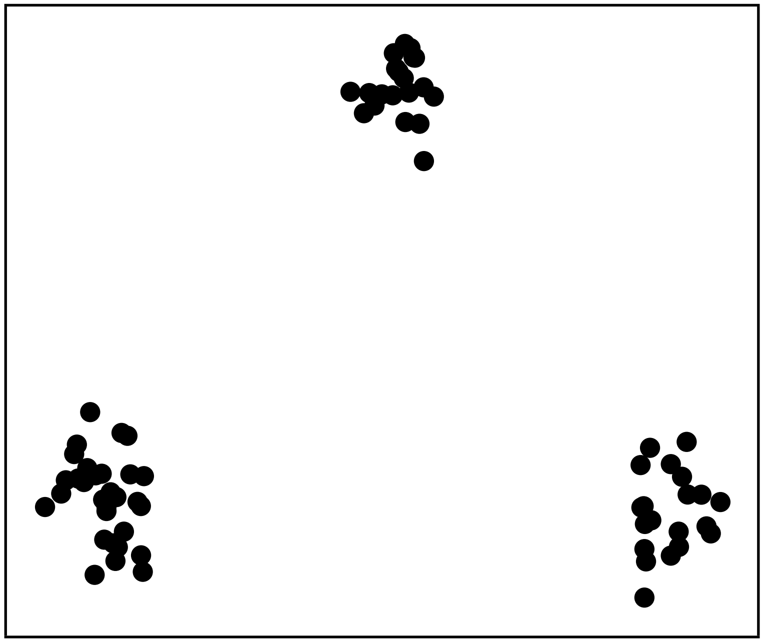
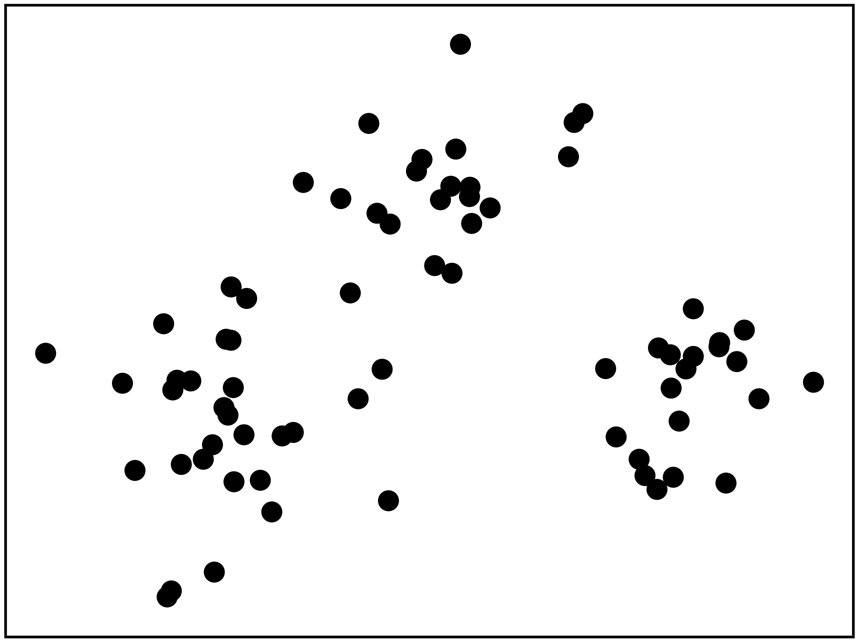


Segmentation



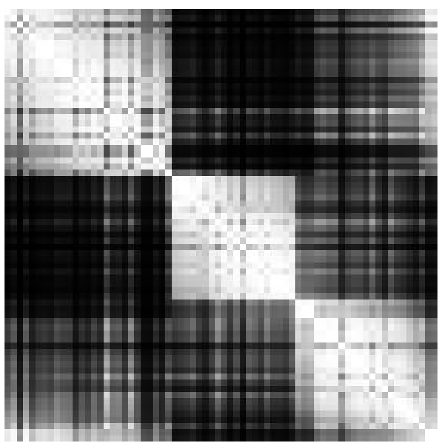
http://www.cs.berkeley.edu/projects/vision/grouping/segbench/BSDS300/html/dataset/images.html

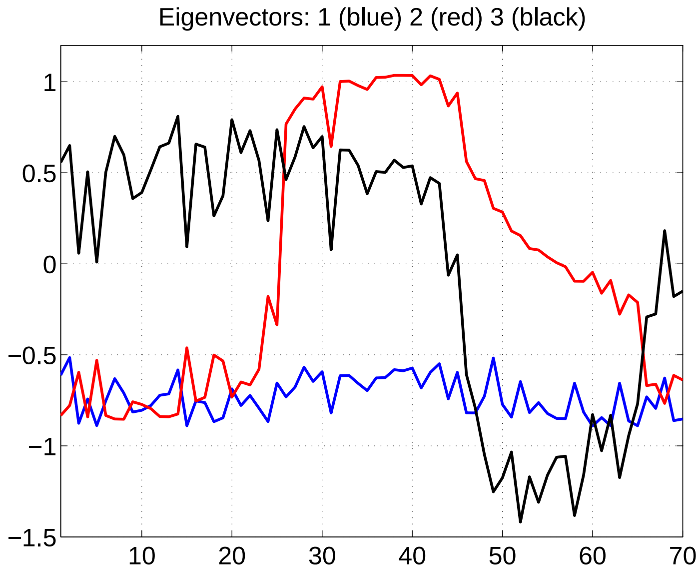
Segmentation

Well Separated	Weakly Coupled
	

Segmentation

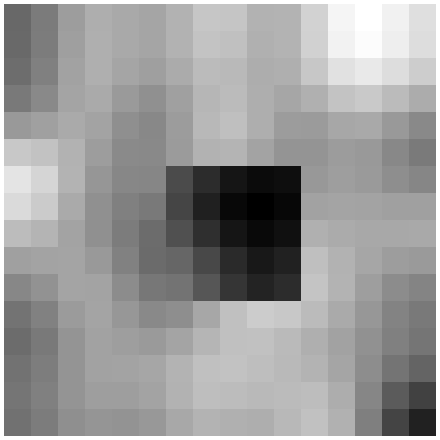
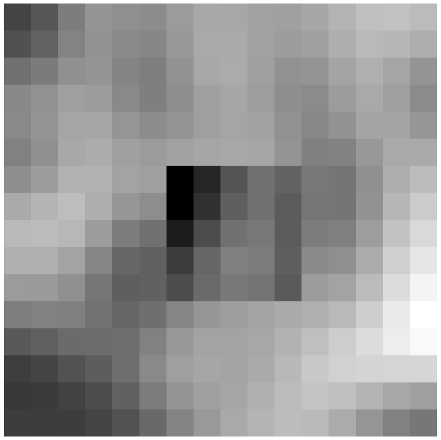
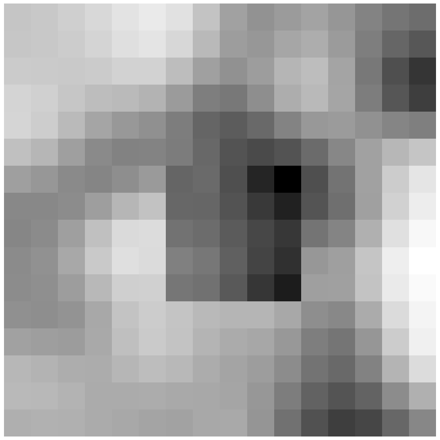
Weakly Separated	Affinities	Eigenvectors



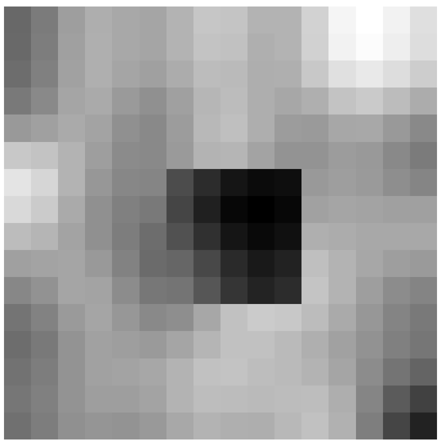
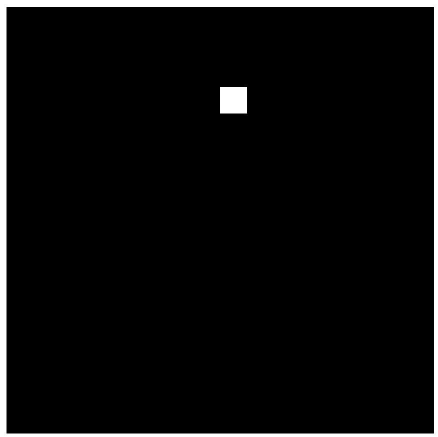
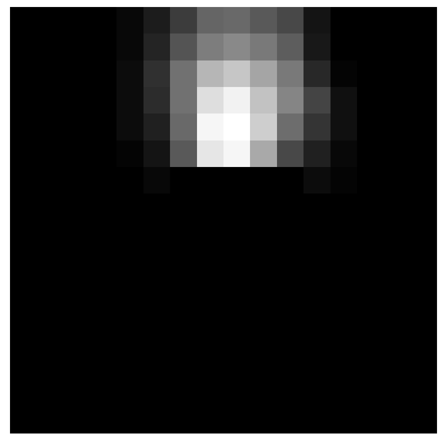
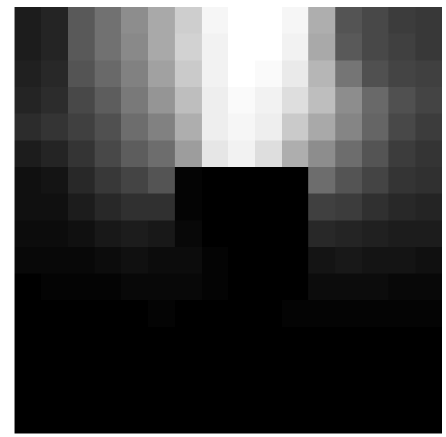


Segmentation

Segment images into a small number of "perceptually meaningful" groups

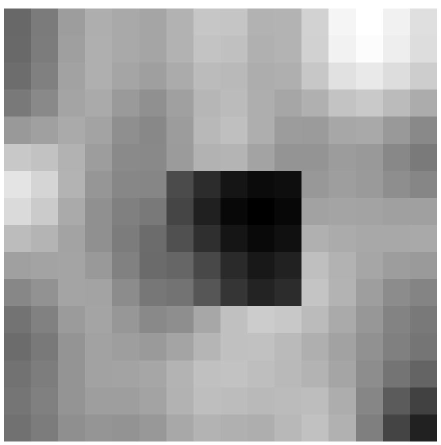
  

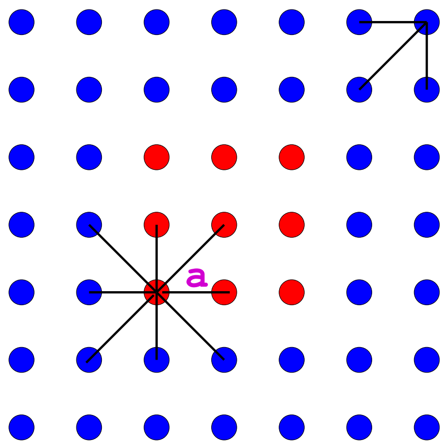
Random Walks

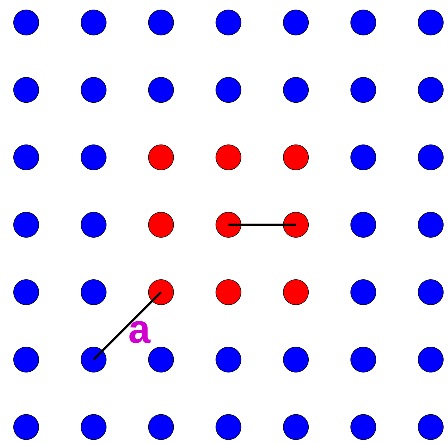
Occluder	Its 0	Its 5	Its 25
			

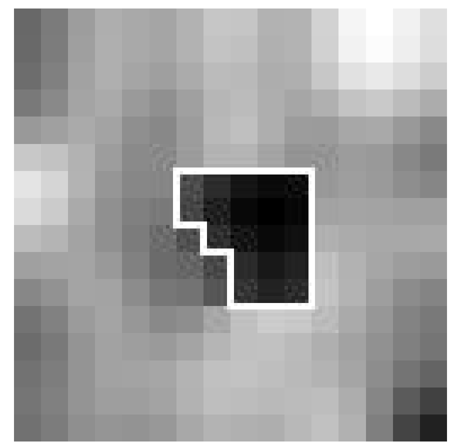
4. EigenCuts: Algorithm

Occluder	Graph	Sensitivities	Cuts



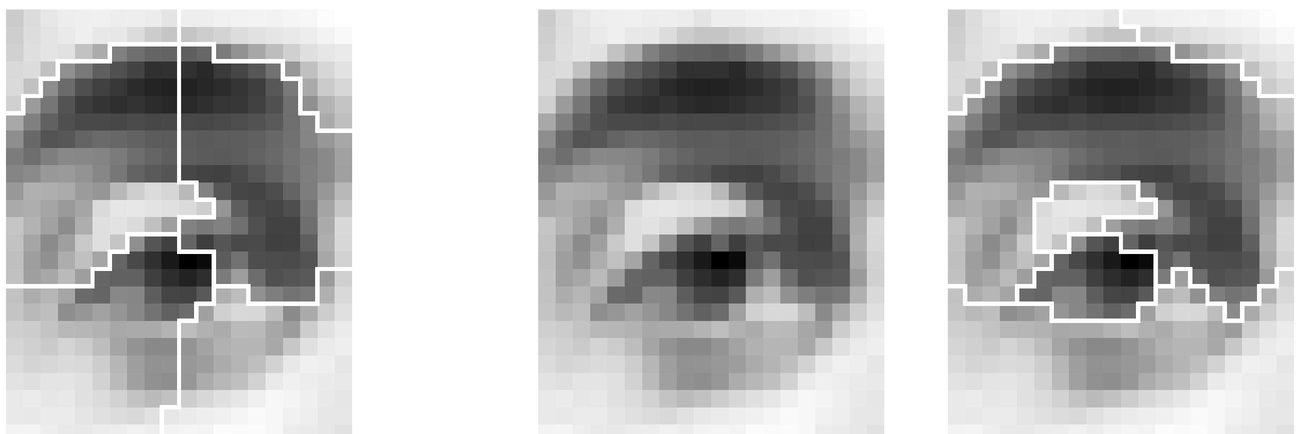






4. EigenCuts: Results and Comparison

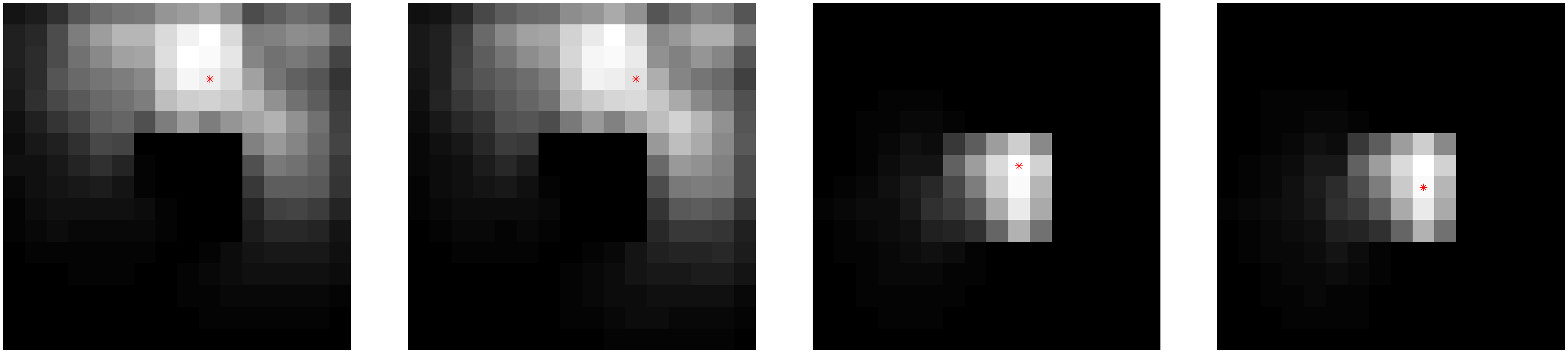
K-means-spectral	Eye	EigenCuts



K-means constrained to give same number of segments as EigenCuts.

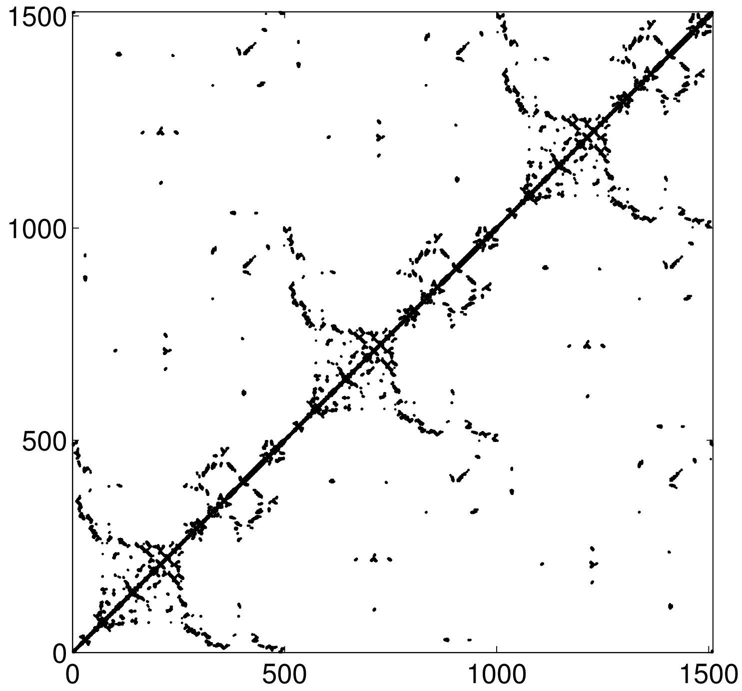
5. Fast EigenSolver

Intuition for transition matrix hierarchy

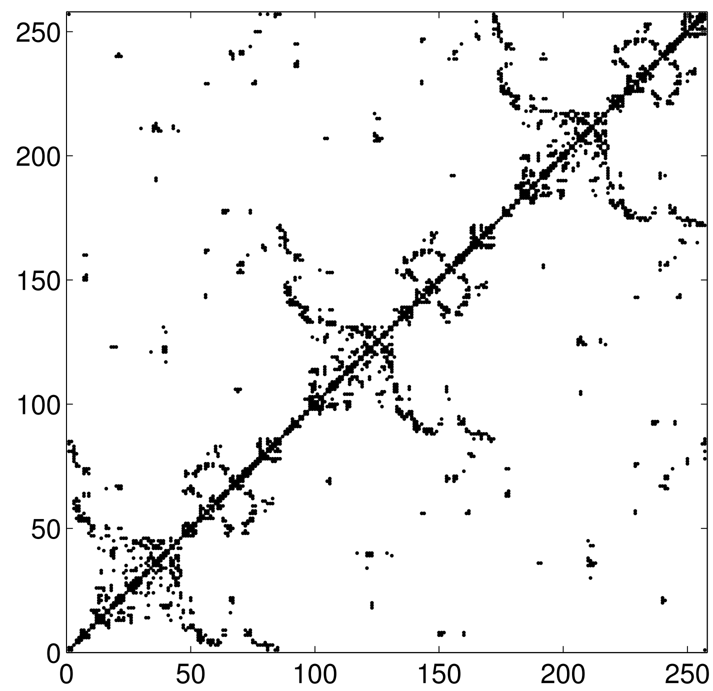


5. Fast EigenSolver

Hierarchical transition matrix



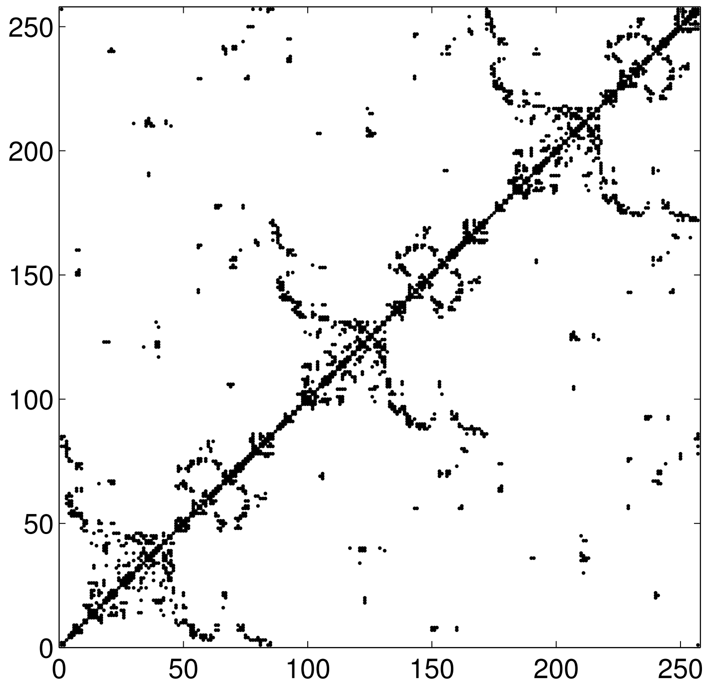
Fine-Scale



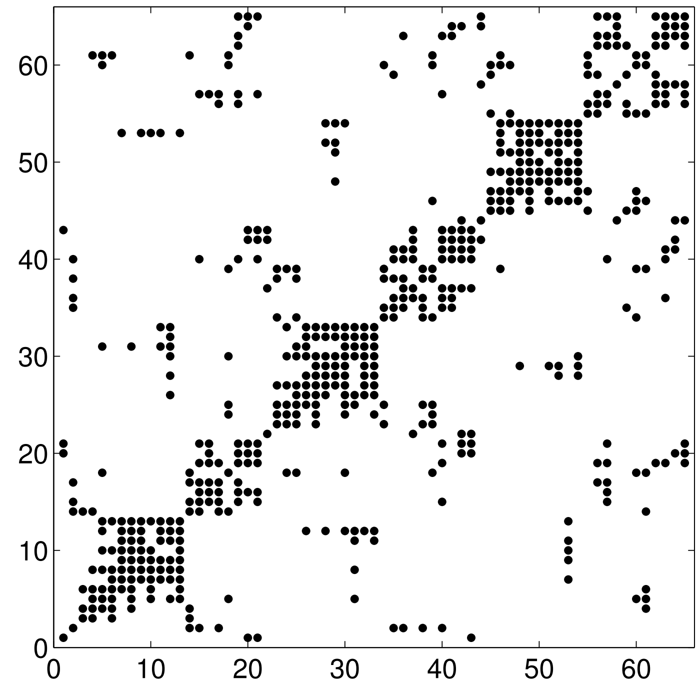
Coarse-Scale 1

5. Fast EigenSolver

Hierarchical transition matrix



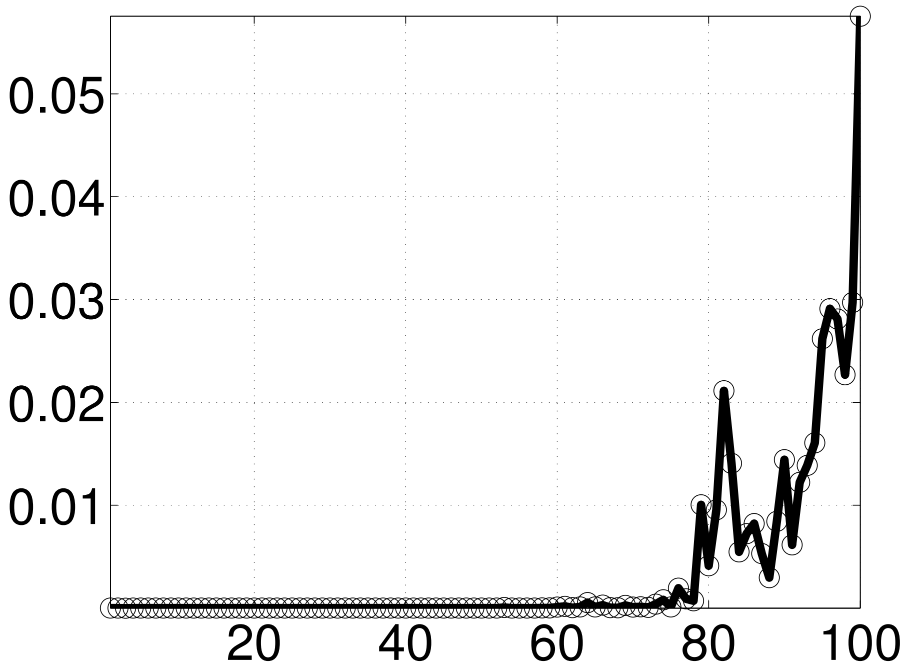
Coarse-Scale 1



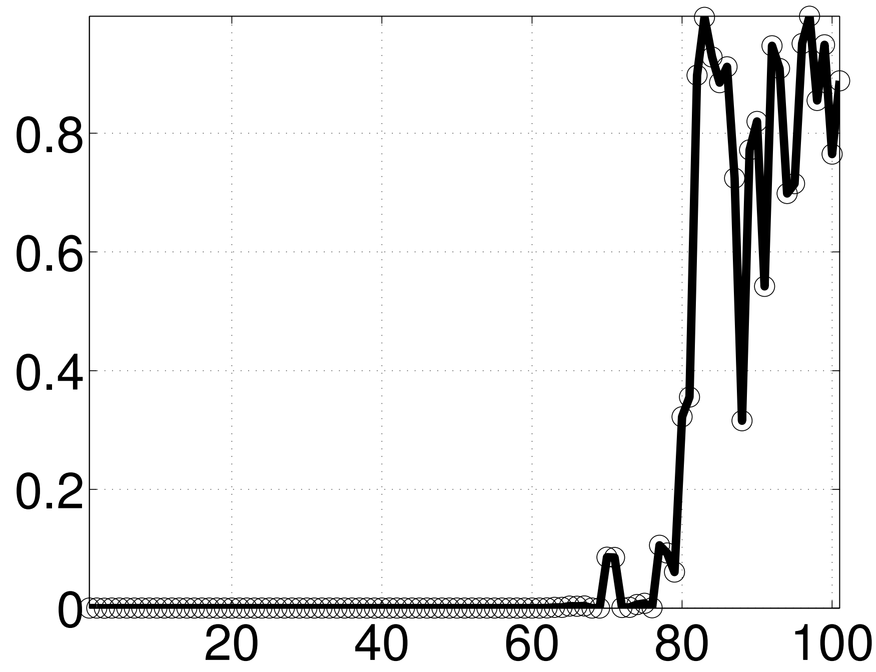
Coarse-Scale 2

5. Fast EigenSolver

Relative Errors



Eigvals



Eigvecs

5. Performance: compute 51 eigenpairs

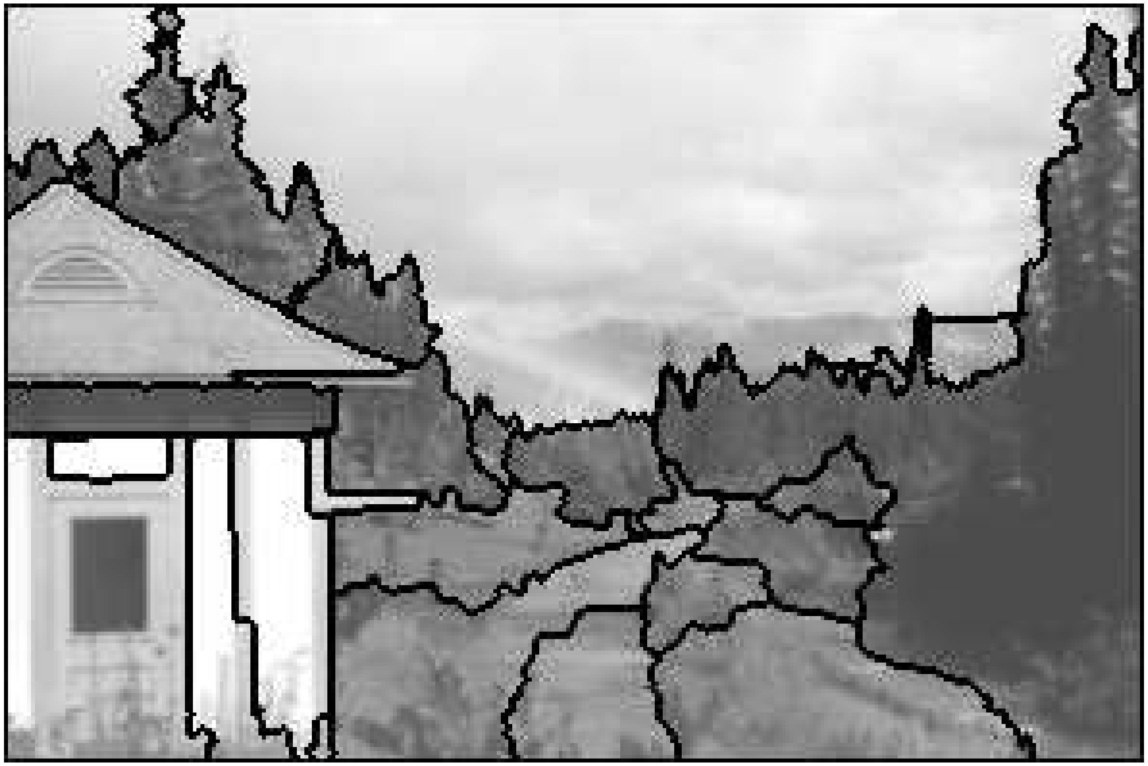
n	`svds.m`	Multi-Scale	Speedup
64^2	20.5	5.5	3.7
65^2	12.6	5.1	2.5
100^2	44.2	13.1	3.4
128^2	230.9	35.2	6.6
129^2	96.9	20.9	4.6
255^2	819.2	90.3	9.1
256^2	2170.8	188.7	11.5
257^2	871.7	93.3	9.3
512^2	20269	739.3	27.4
513^2	7887.2	461.9	17.1
800^2		1936.6	

6. Spectral Embedding and Min-Cut

[EstradaJepsonChennubhotla, BMVC'04]

6. Spectral Embedding and Min-Cut

[EstradaJepsonChennubhotla, BMVC'04]



Part–II

Sparse Principal Component Analysis

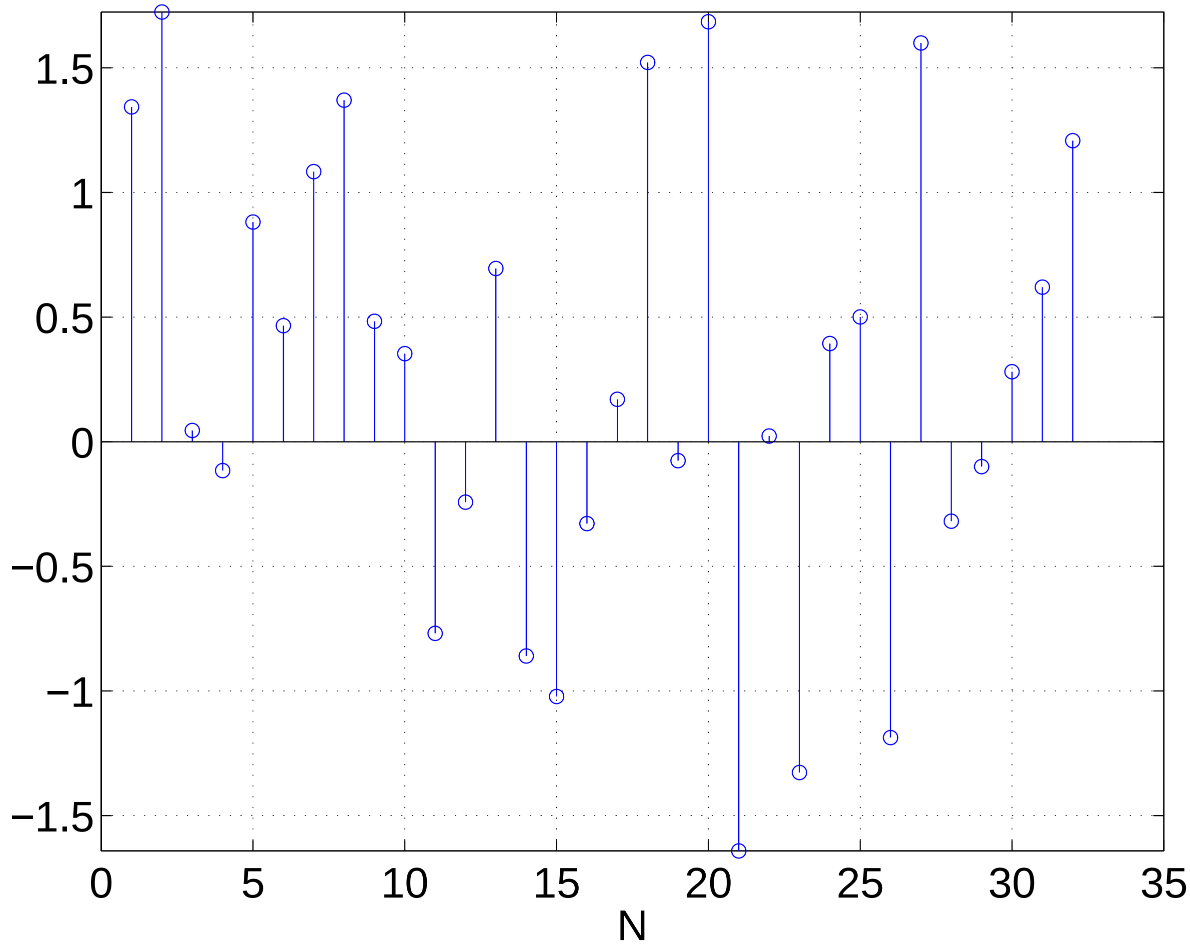
S-PCA

Dataset with Known Correlation Structure

Generative Model:

$\vec{y} = \vec{x}$

- 32-pixels
- $10,000$ images

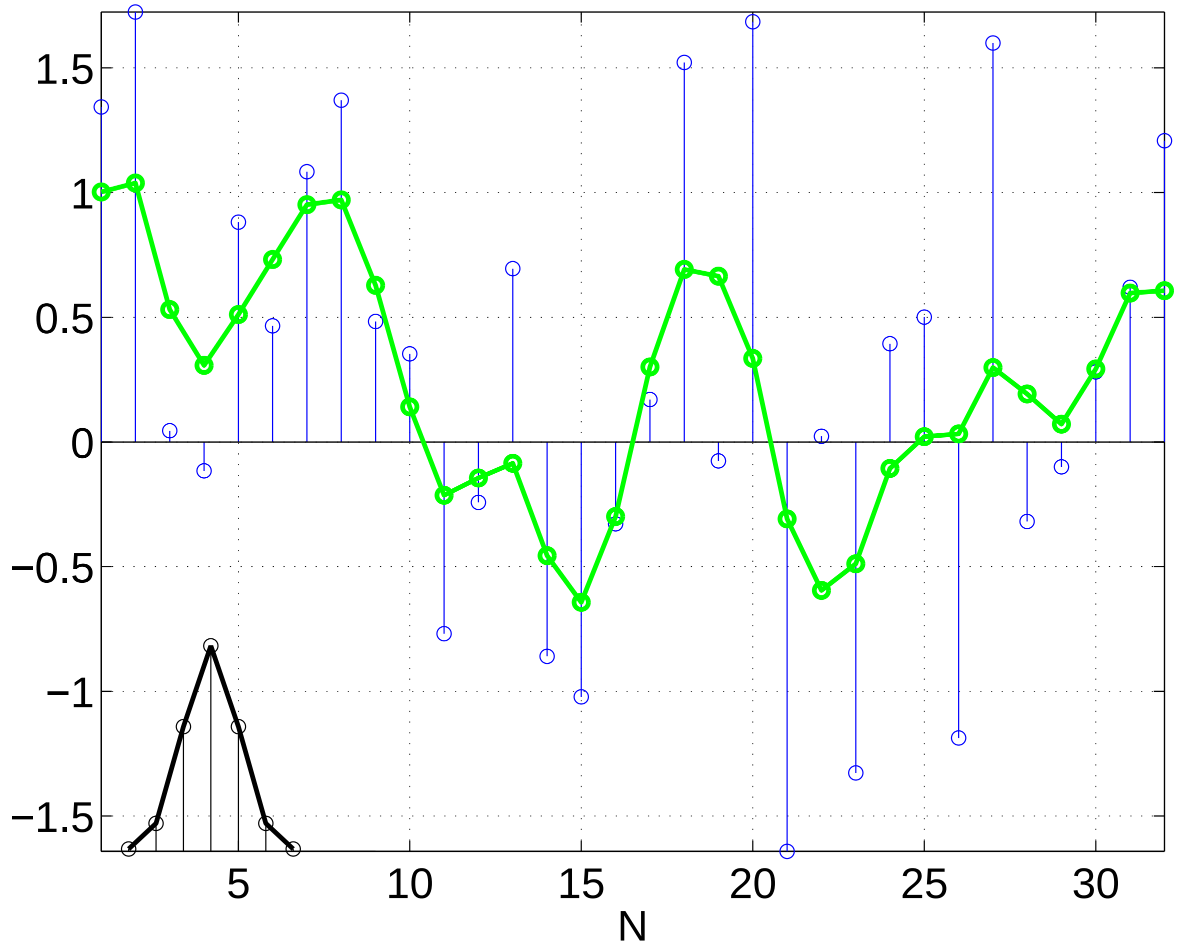


Dataset with Known Correlation Structure

Generative Model:

$$\color{red}{\vec{y}} = \underbrace{\vec{g}}_{\color{red}{\text{Low-pass}}} \otimes \color{blue}{\vec{x}}$$

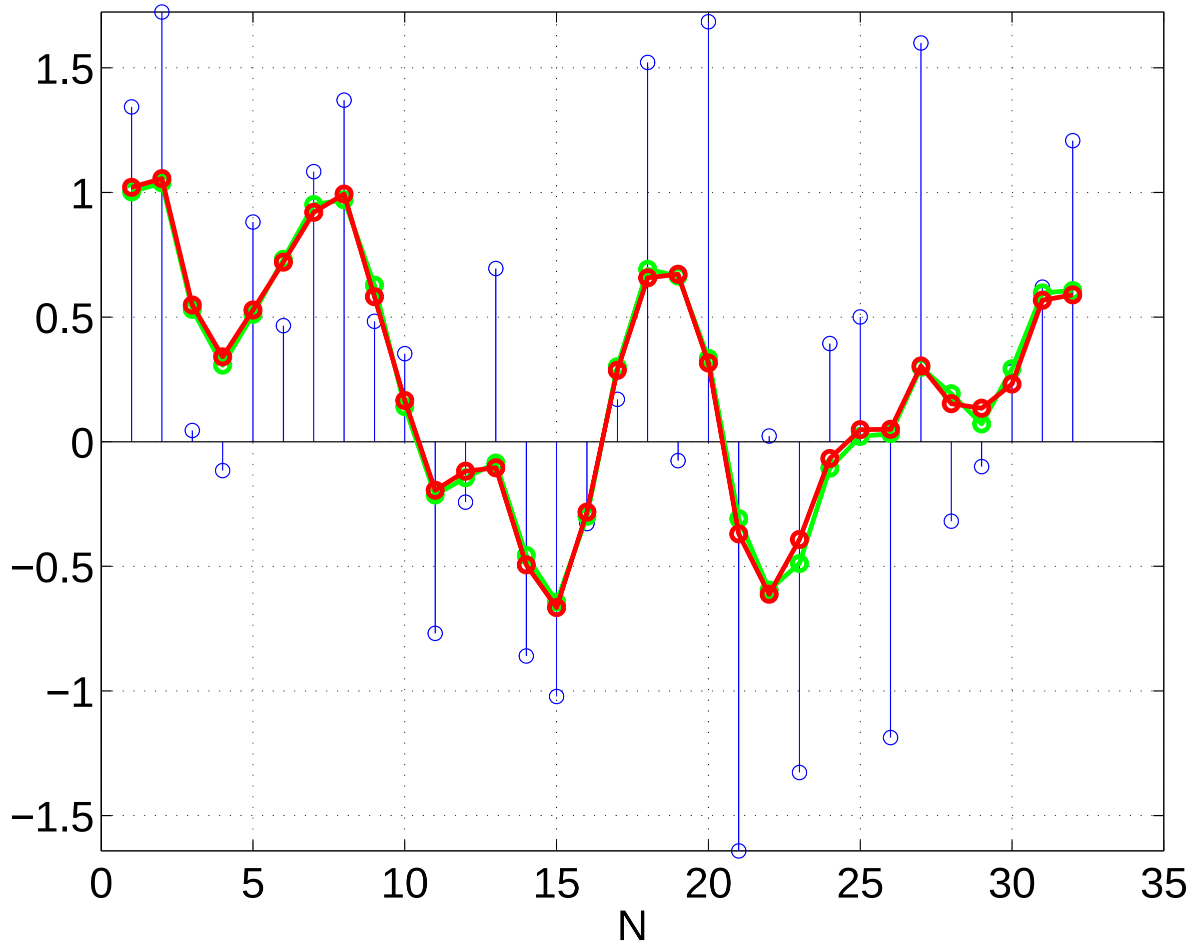
- $10,000$ images



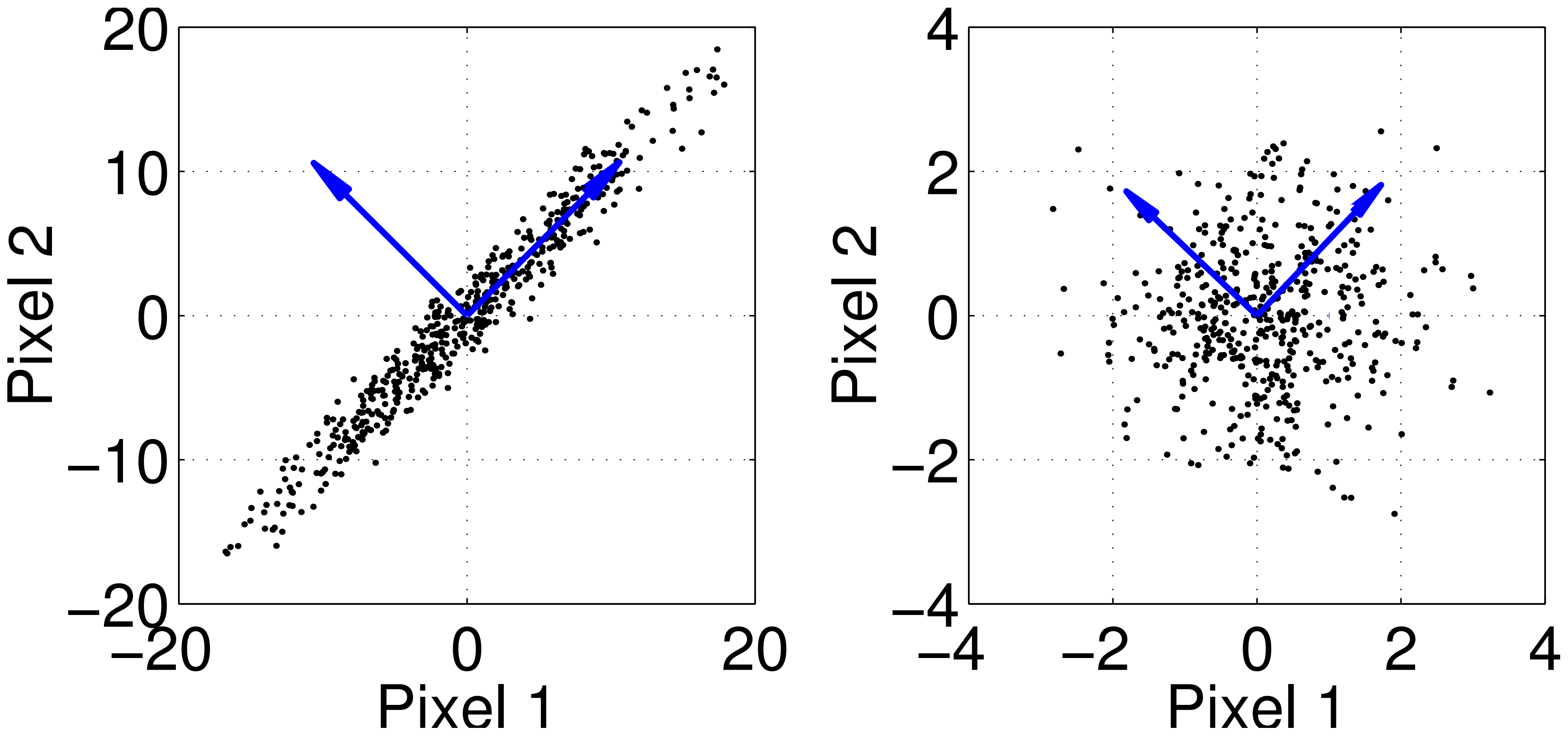
Dataset with Known Correlation Structure

Generative Model:

$$\vec{y} = \underbrace{\vec{g}}_{\text{Low-pass}} \otimes \vec{x} + \underbrace{\kappa\vec{n}}_{\text{Noise}}$$



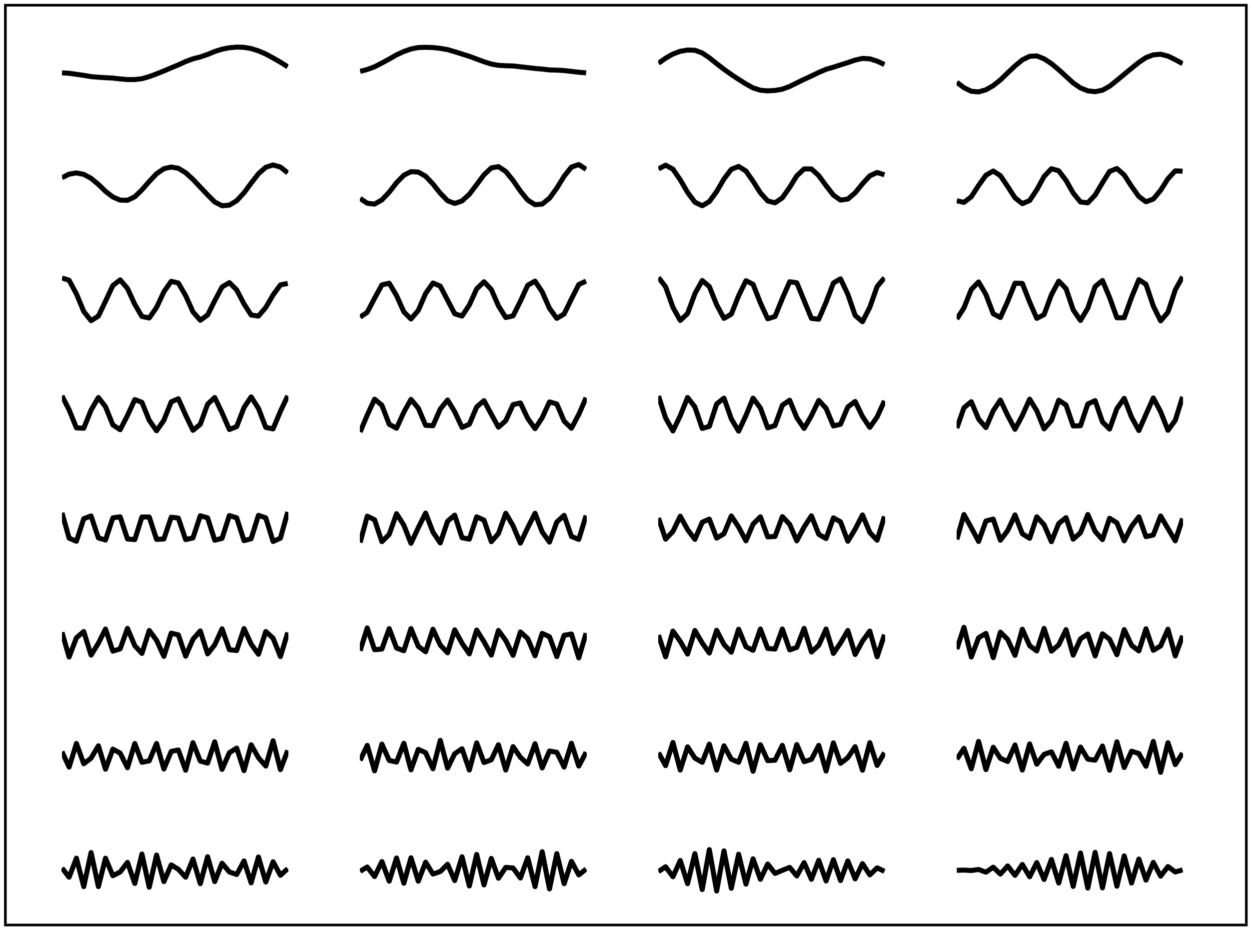
Principal Component Analysis (PCA)



PCA Representation

Basis

- 32-dim
- spatial frequency specific, *global*



Sparse-PCA: Intuition

Promote Sparsity $\Rightarrow$ rotate basis directions

Sparse-PCA Framework

Minimize $\quad C(\vec{d}, U, \lambda) = C_1(\vec{d}) + \lambda C_2(U)$

$$C_1 = \text{Retains PCA directions}$$

$$C_2 = \text{Promotes Sparsity}$$

$$\vec{d} = \text{Relative Variance}$$

$$U = \text{Orthonormal Basis}$$

$$\lambda = \text{Relative importance of sparsity}$$

Sparse-PCA Framework

Minimize $$C(\vec{d}, U, \lambda) = C_1(\vec{d}) + \lambda C_2(U)$$

Relative Variance $$d_k = \frac{\sigma_k^2}{\sum_{j=1}^{M} \sigma_j^2}$$

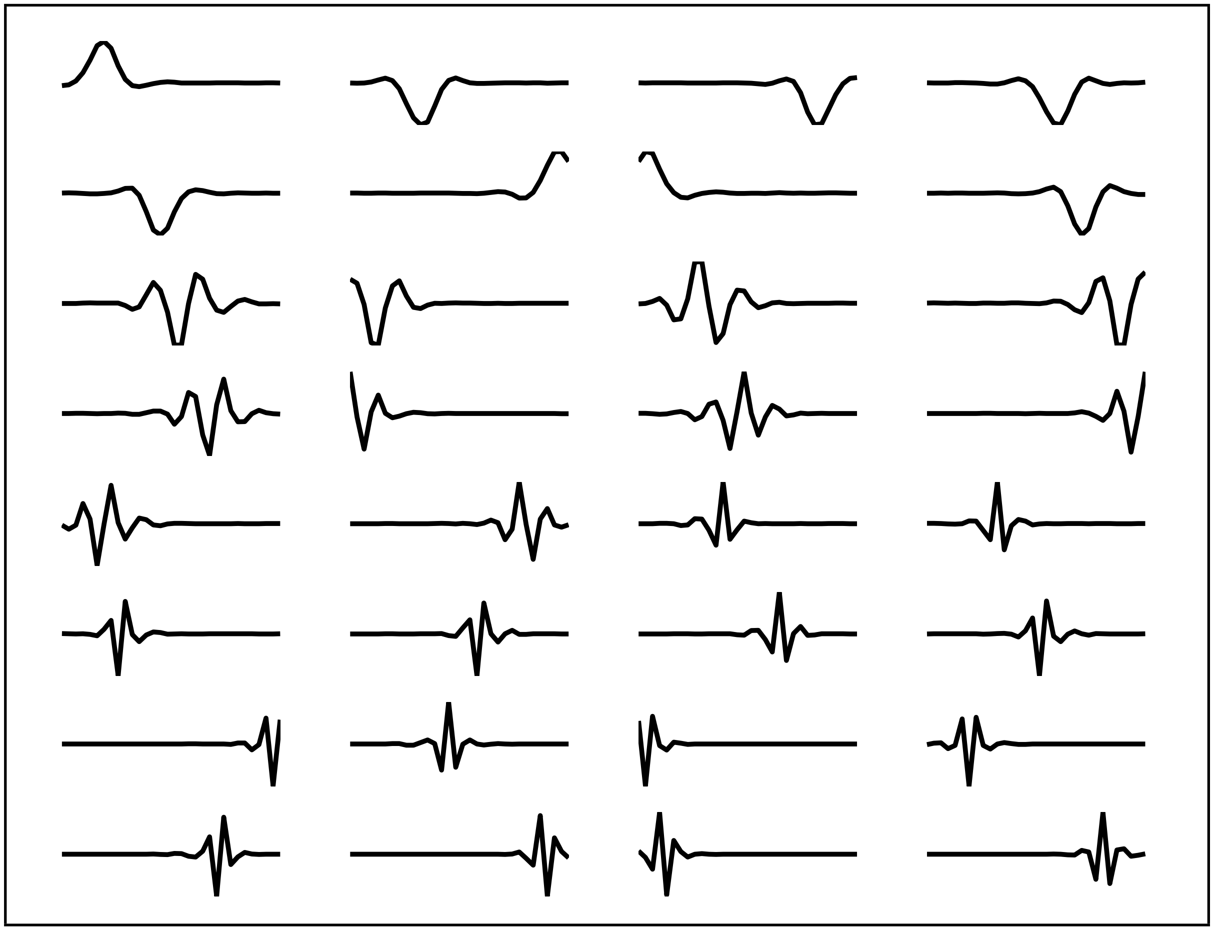
Channel Entropy $$C_1(\vec{d}) = \sum_{j=1}^{M} -d_j \log(d_j)$$

Basis Entropy $$C_2(U) = \sum_{m=1}^{M} \sum_{n=1}^{N} -u_{m,n}^2 \log(u_{m,n}^2)$$

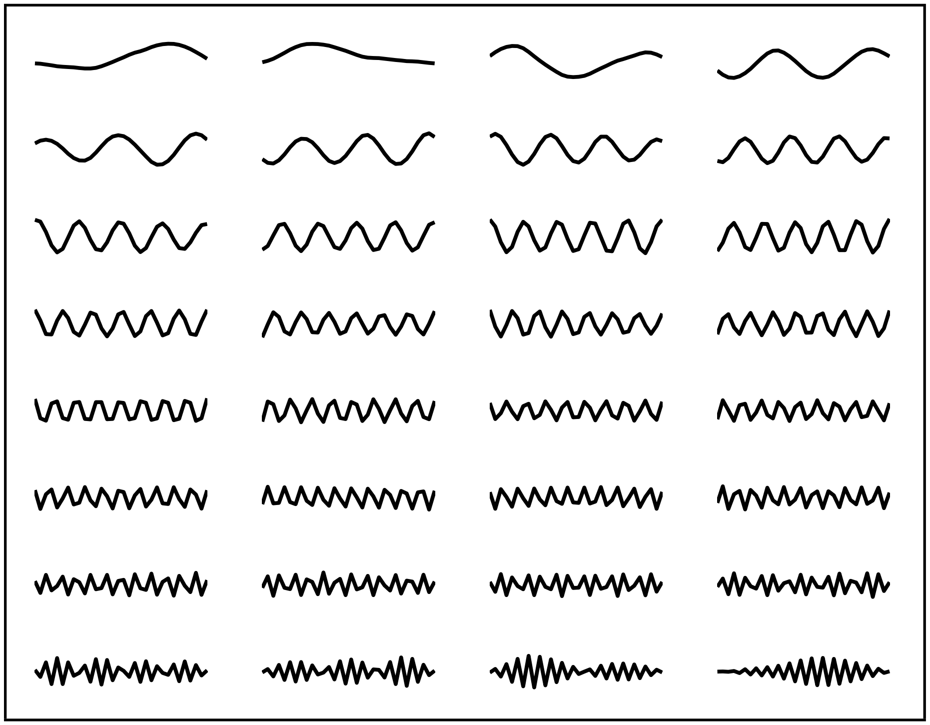
Sparse-PCA Representation

Basis

- local/multi-scale structure

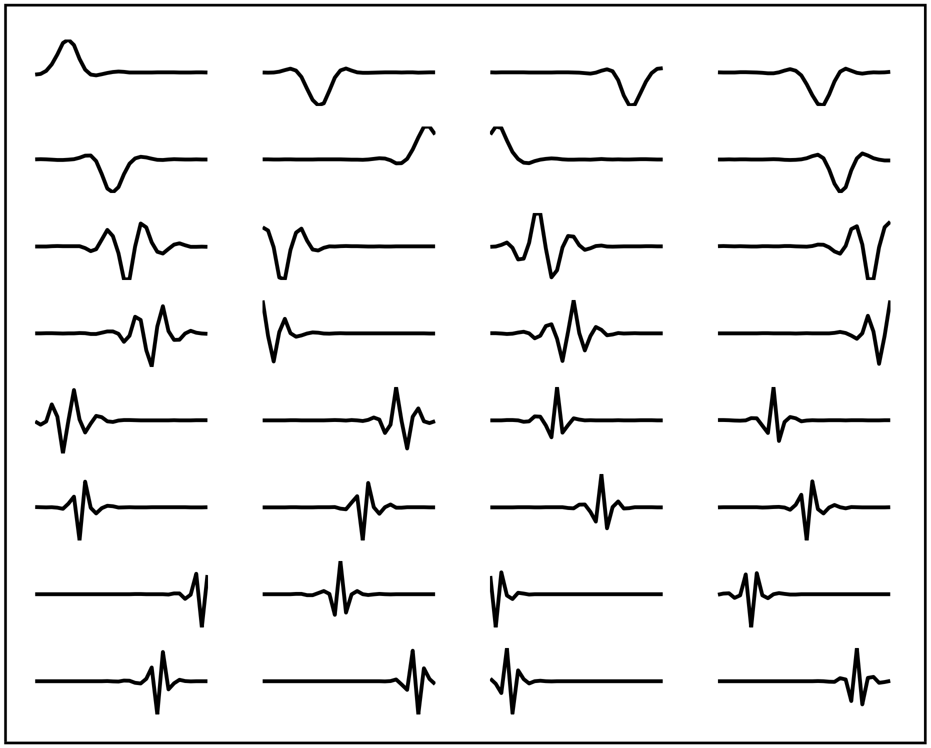


PCA



Global

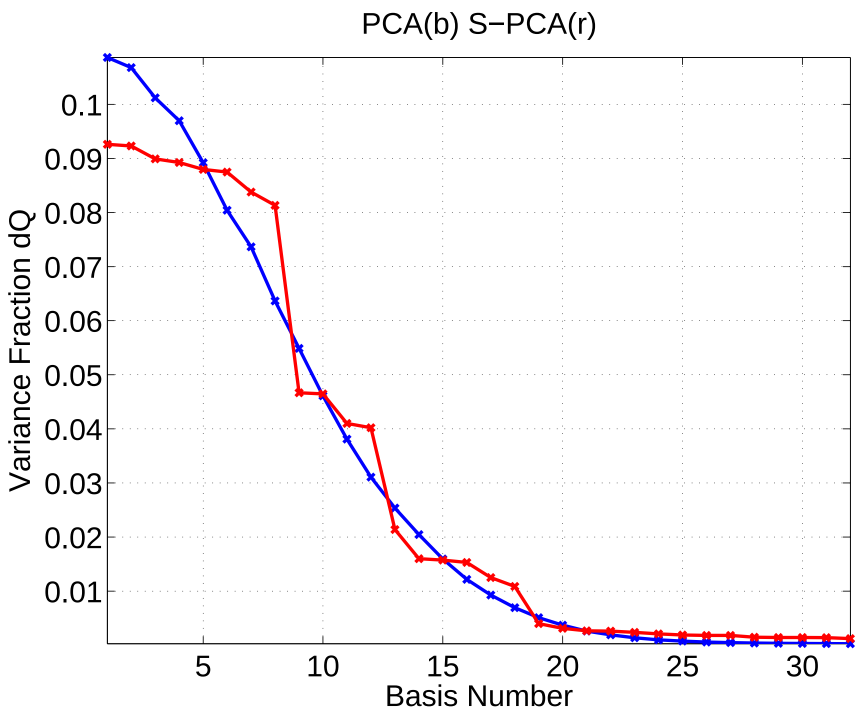
S-PCA

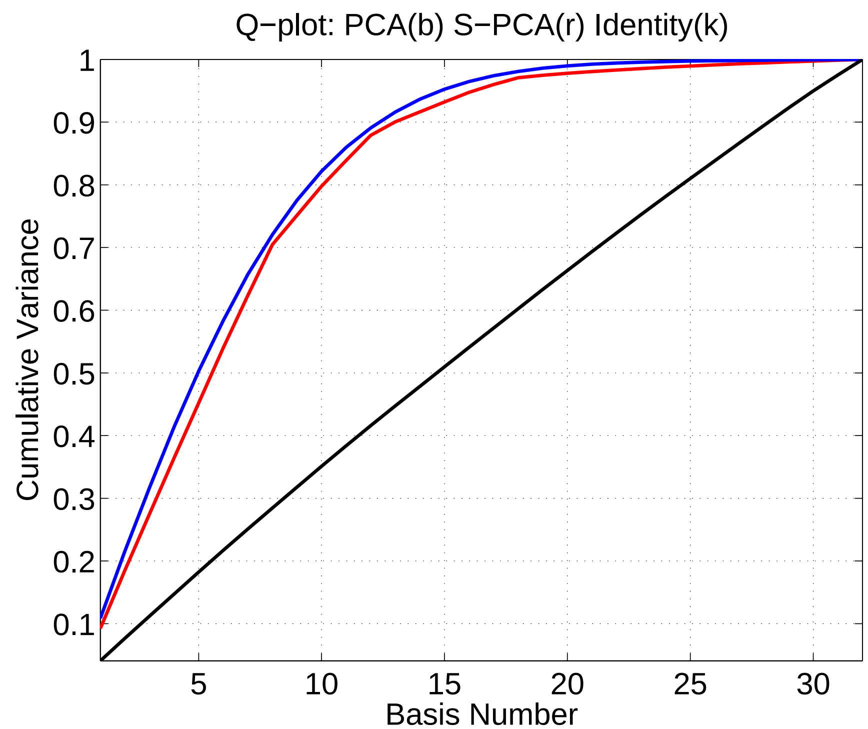


Local/Multi-Scale

- Intuitive results on other datasets with known structure

Reshaping of Variance Spectrum



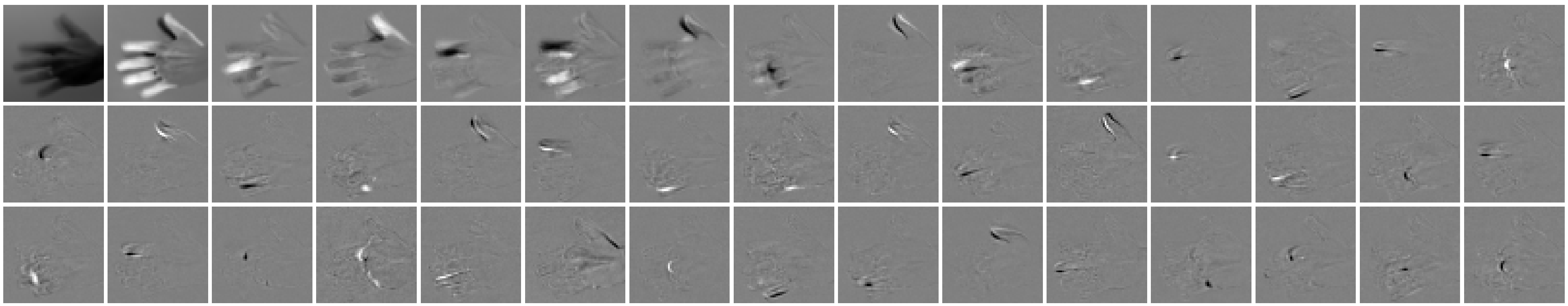


Variance Fraction

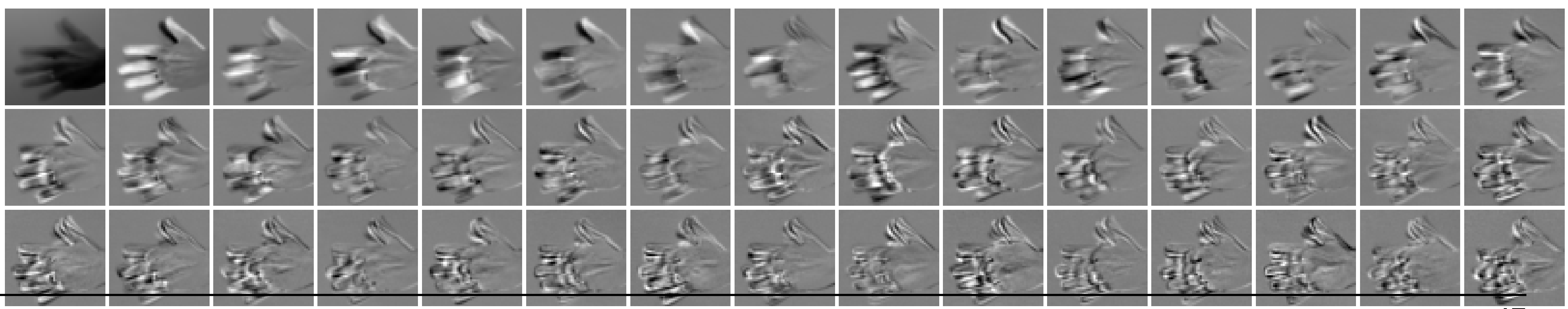
Cumulative Variance

Hand Images

S-PCA

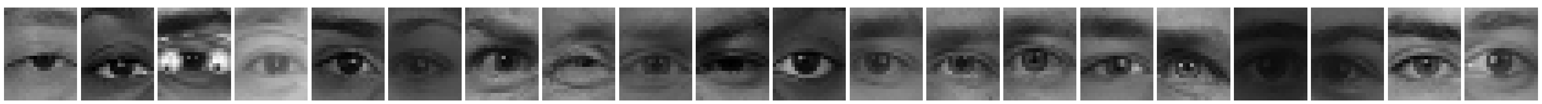


PCA

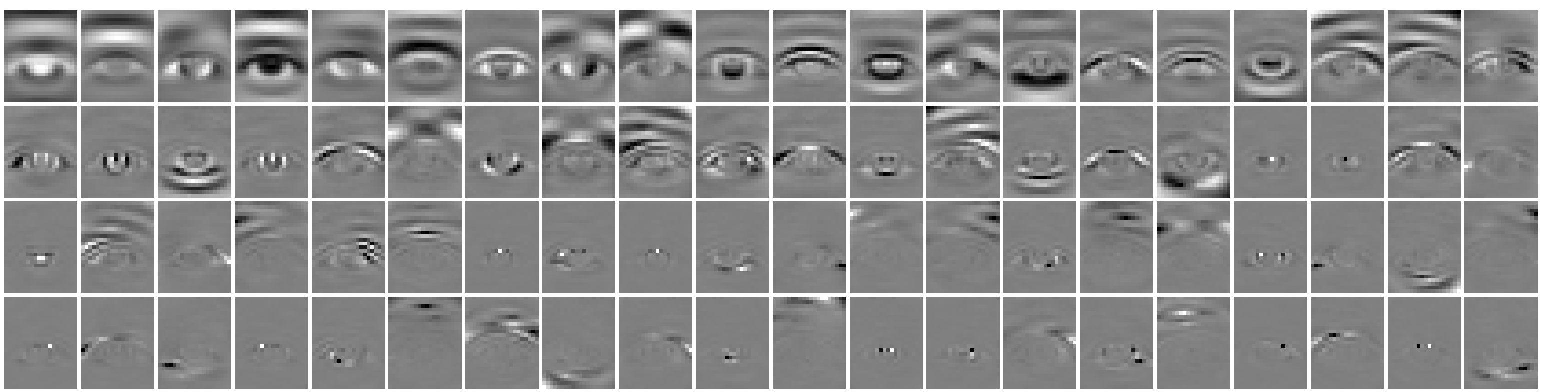


Sparse Principal Component Analysis

Eyes

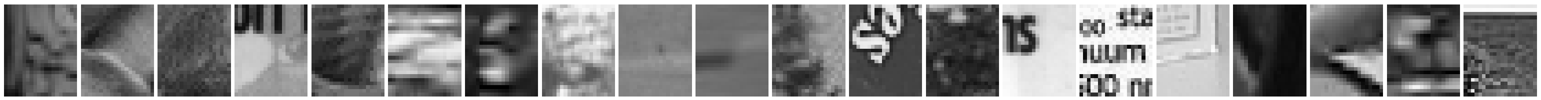


Sparse-PCA Basis

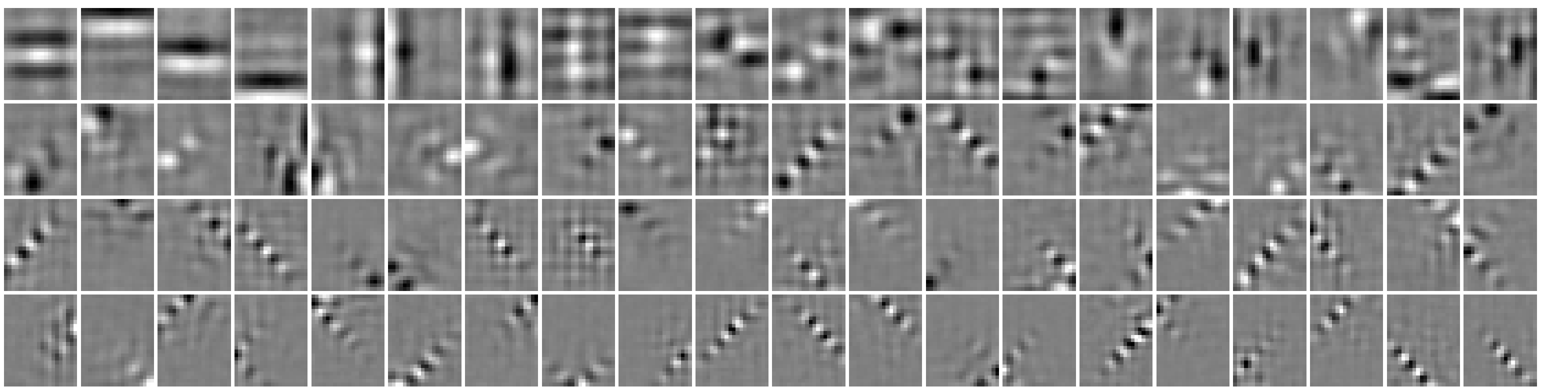


Sparse Principal Component Analysis

Generic Image Patches



Sparse-PCA Basis



S-PCA: Applications

- Contrast-Invariant Appearance Detection
  - Perceptual Image Distortion for Image Similarity using S-PCA
- Enforcing Sparsity by Coring
  - Iterative Image Reconstruction
  - flop-count for a 50-dim subspace reconstruction by S-PCA is 40% faster than PCA

Part–III

EigenCuts

A Spectral Clustering Algorithm

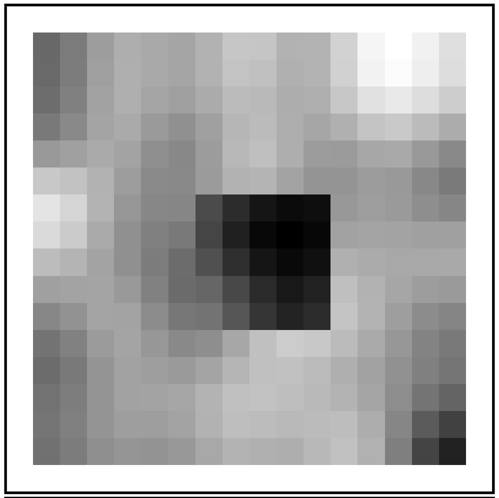
Clustering Methods

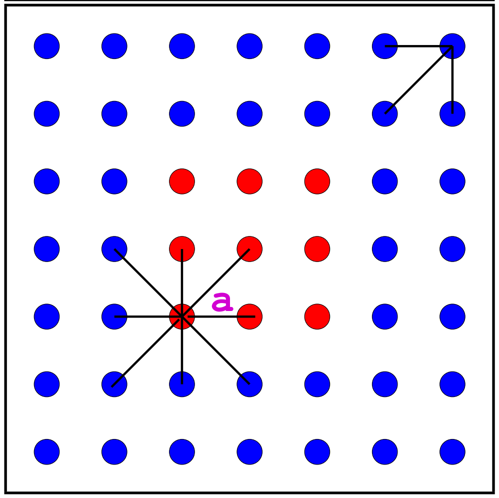
Prior Work

- Agglomerative and Divisive Clustering Algorithms
- Markov Random Field Methods
- Graph-Theoretic Approaches
  - Graph Cuts
  - Spectral Clustering (Tools from Linear Algebra)

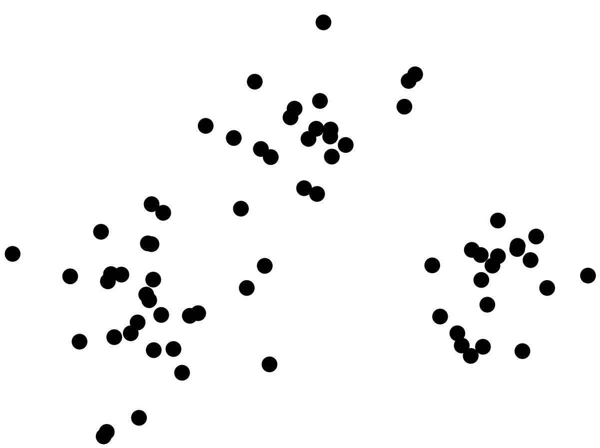
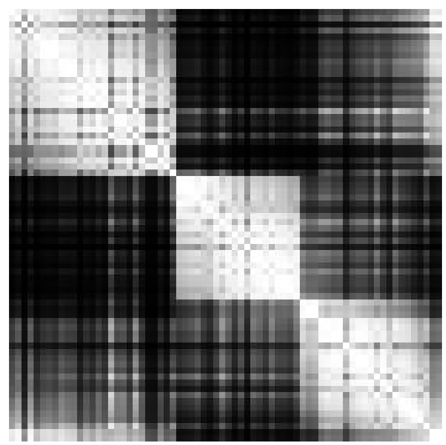
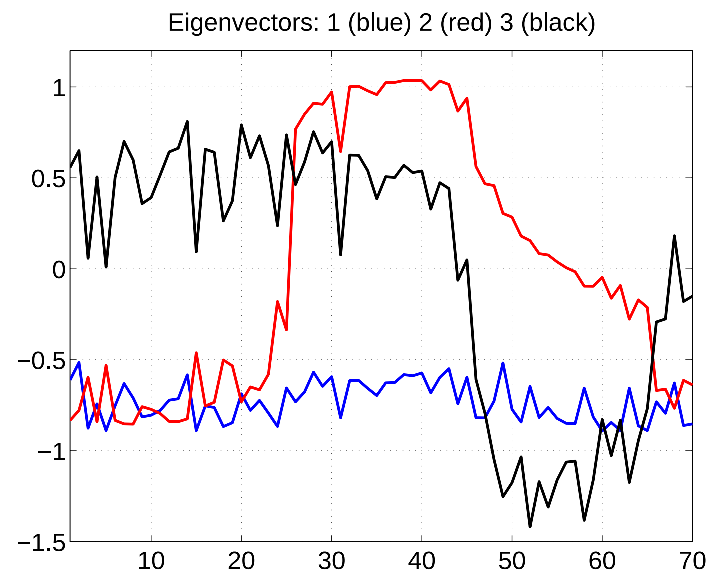
From Affinities to Markov

- Affinity Matrix $A = \{a_{i,j}\}$
- $a_{i,j} = \exp\left\{ \frac{-\left(I(\vec{x}_i) - I(\vec{x}_j)\right)^2}{2\sigma^2} \right\}$
- $M = AD^{-1}$
- $M\vec{q}_k = \lambda_k \vec{q}_k$ (eigen pairs of M)
- $|\lambda_k| \leq 1$



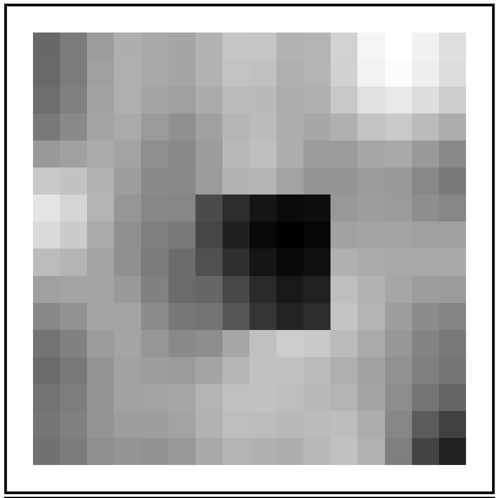


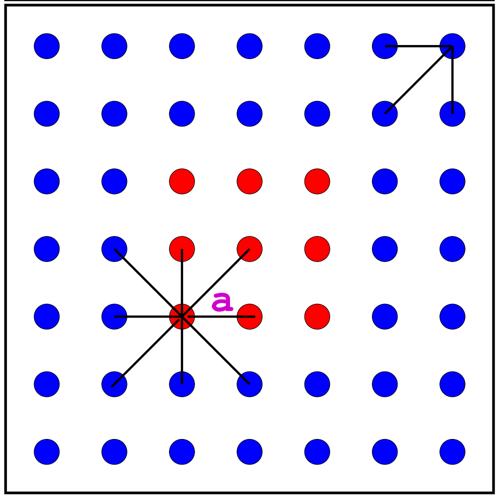
From Affinities to Markov

Weakly Separated	Affinities	Eigenvectors
		

Markov Chain Propagation

- $\vec{p}^{\,0}$ = initial distribution
- $\vec{p}^{\,\beta} = M^{\beta}\vec{p}^{\,0}$
- $\vec{p}^{\,\beta} = \vec{\pi} + \sum_{k=2}^{n} \lambda_k^{\beta}\vec{q}_k \underbrace{\left(\vec{q}_k^{\,T} D^{-1}\vec{p}^{\,0}\right)}_{r_k}$
- $\vec{\pi}$ = Stationary Distribution
- $\lambda_k^{\beta}\vec{q}_k r_k$ = Perturbation





Flow of Probability Mass

Characterizing the random walk over each link

$$\vec{p}^{\,0} = \text{initial distribution}$$

$$\text{Flow} \quad F_{i,j}(\vec{p}^{\,0}) = m_{i,j}p_j^0 - m_{j,i}p_i^0$$

$$= \text{probability xfer from } j \to i$$

$$\text{Matrix form} \quad F(\vec{p}^{\,0}) = M\,\mathrm{diag}(\vec{p}^{\,0}) - \mathrm{diag}(\vec{p}^{\,0})M^T$$

$$\text{No Flow} \quad F(\vec{\pi}) = 0$$

EigenFlows and Half-Lives

Characterizing the random walk over a group of links

Random Walk $$\vec{p}^{\,\beta} = \vec{\pi} + \sum_{k=2}^{n} \lambda_k^{\beta} \vec{q}_k r_k$$

EigenFlows $$F(\vec{p}^{\,\beta}) = \underbrace{F(\vec{\pi})}_{=0} + \sum_{k=2}^{n} \lambda_k^{\beta} F(\vec{q}_k) r_k$$

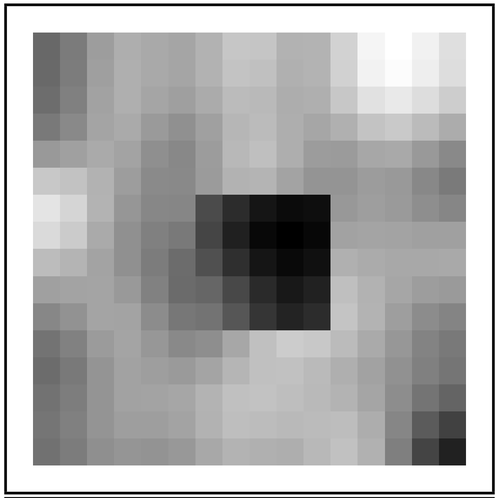
Rate of Decay $$\lambda_k^{\beta_k} = 1/2$$

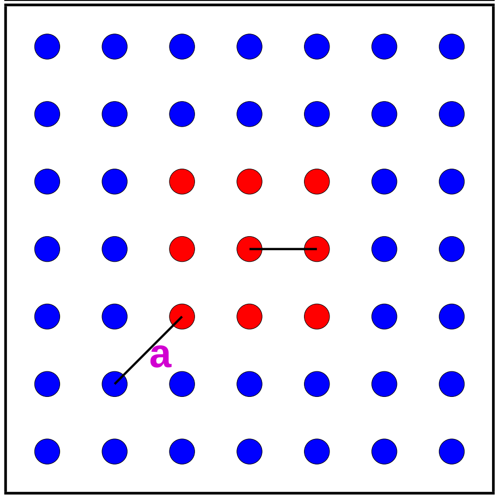
Half-Life $$\Rightarrow \beta_k = \frac{\log(1/2)}{\log(\lambda_k)}$$

Bottlenecks and Sensitivities

- $(\vec{q}_k, \lambda_k) \rightarrow \beta_k$
- Perturb: $a_{i,j} \rightarrow a_{i,j} + \alpha_{i,j}$
- Ignore $a_{i,i}$
- For $\alpha_{i,j} \rightarrow 0$, evaluate

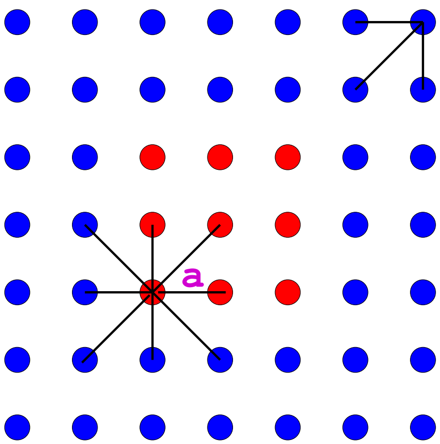
$$\frac{\partial \log(\beta_k + \beta_0)}{\partial \alpha_{i,j}}$$

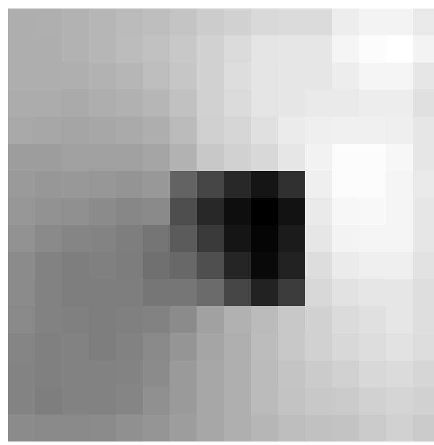


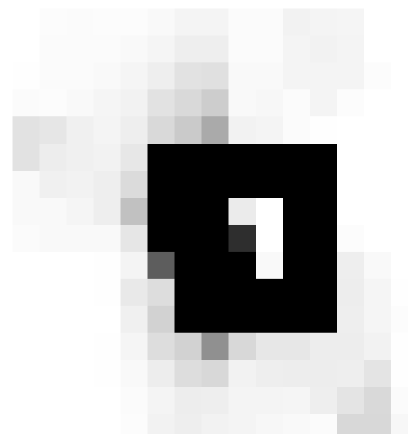


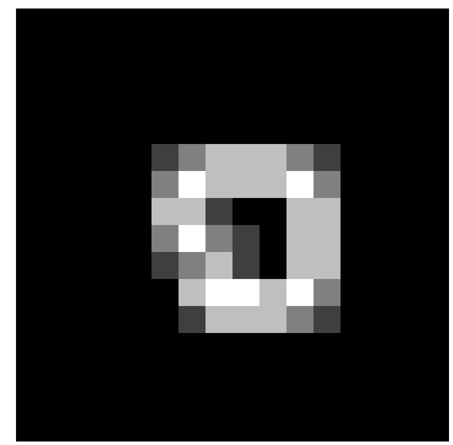
EigenCuts: Algorithm

Graph	EigenVec 2	Sensitivities	Cuts



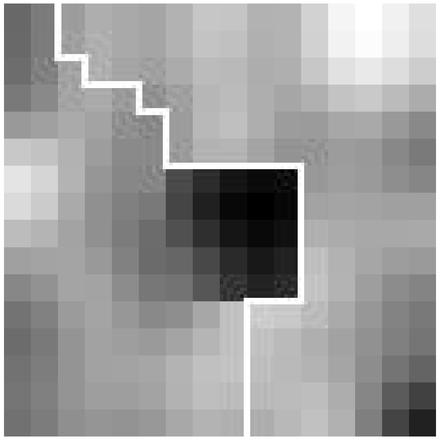
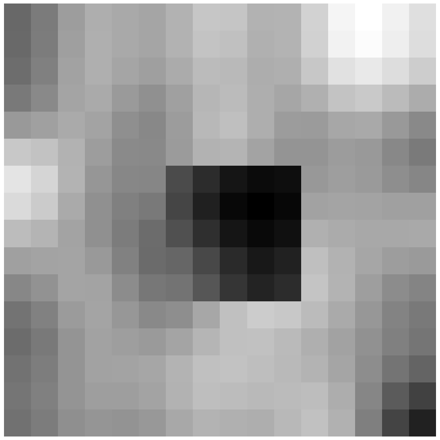
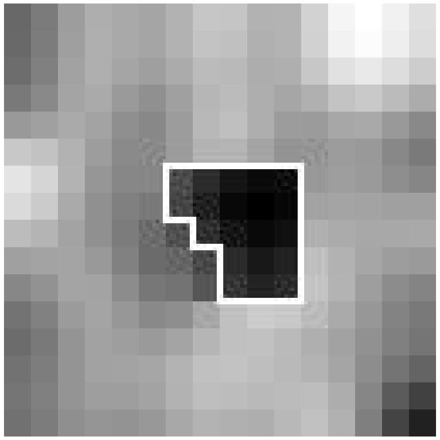






EigenCuts: Results and Comparison

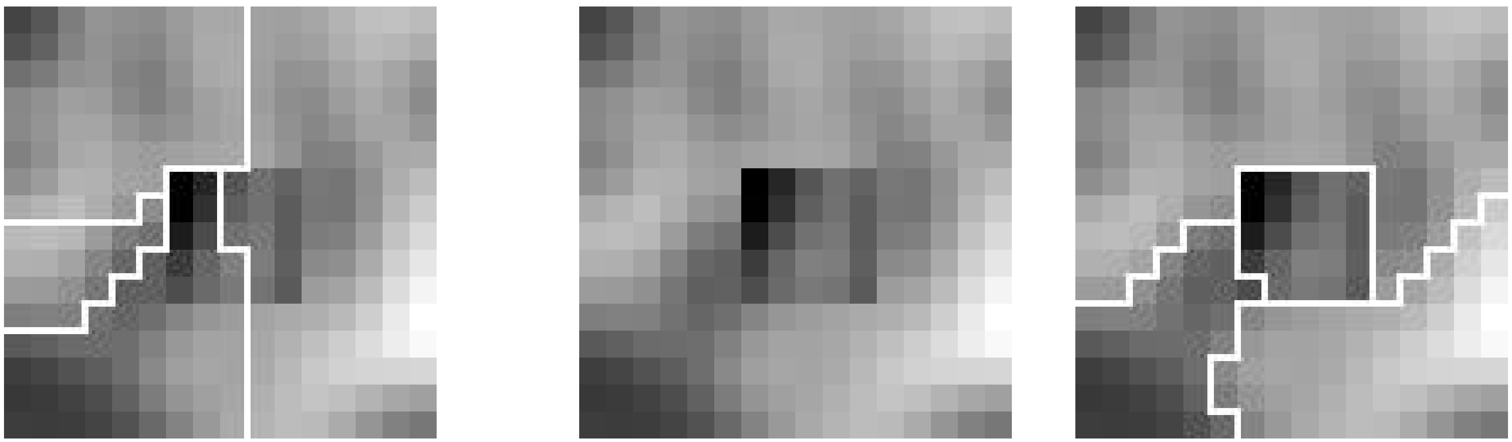
K-means-spectral	Occluder	EigenCuts

K-means constrained to give same number of segments as EigenCuts.

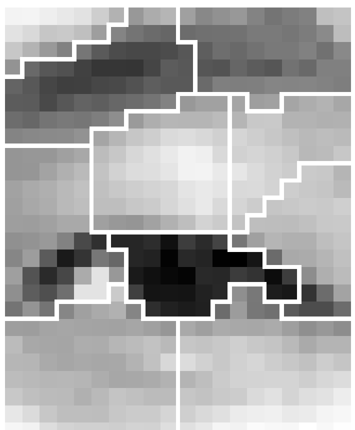
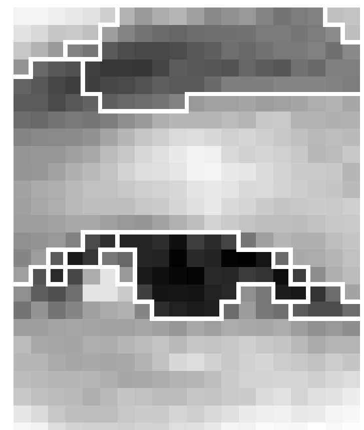
EigenCuts: Results and Comparison

K-means-spectral	Occluder	EigenCuts



K-means constrained to give same number of segments as EigenCuts.

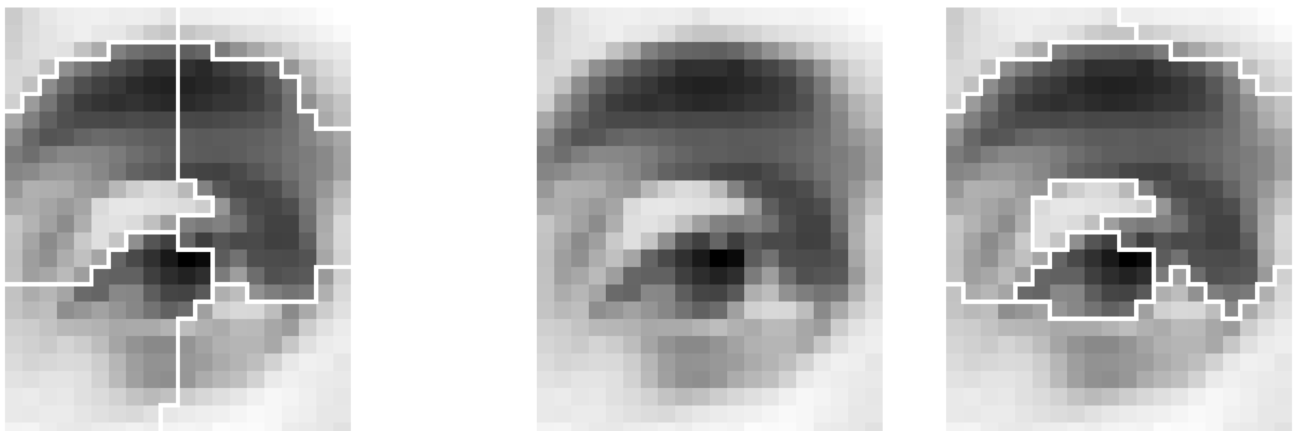
EigenCuts: Results and Comparison

K-means-spectral	Eye	EigenCuts
		

K-means constrained to give same number of segments as EigenCuts.

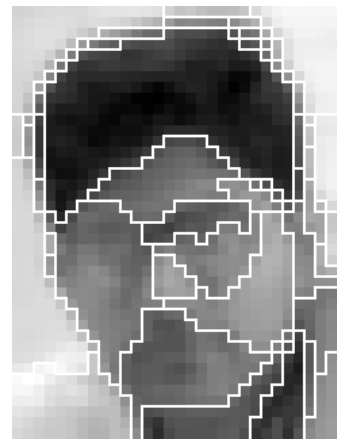
EigenCuts: Results and Comparison

K-means-spectral	Eye	EigenCuts



K-means constrained to give same number of segments as EigenCuts.

EigenCuts: Results and Comparison

K-means-spectral	Face	EigenCuts
		

K-means constrained to give same number of segments as EigenCuts.

Improving EigenCuts

- How to speed up the eigen solver?
  - Hierarchical Representation of Transition Matrices
- How to speed up cutting?
  - Multi-Scale EigenCuts

Conclusions

S-PCA:

- Inutitive model via multi-scale decomposition
- Efficient computation of output coefficients because of sparsity
- Output coefficients are slightly correlated

Tradeoff: decorrelation vs sparsity

Conclusions

EigenCuts:

- Identify bottlenecks
- Bottleneck analysis applicable to a variety of clustering problems
- Basic version computationally expensive

Future Work

- S-PCA:
  - Sparse-InfoMax to account for noise in the ensemble
  - Driving sparsity hard with a Mixture-of-Gaussian prior
- EigenCuts:
  - Running EigenCuts for just 1 iteration
  - Perturbing the elements of the Markov matrix directly

Contributions

- Sparse Principal Component Analysis
  - Multi-scale structure from second-order correlations
  - Sparse Iterative Reconstruction
  - Contrast-Invariant Appearance Detection
- EigenCuts
  - Bottleneck Analysis for Clustering
  - Fast Eigensolver
  - Multi-Scale EigenCuts

Collaborators

Toronto

- Allan Jepson
- Francisco Estrada

IBM

- Myron Flickner, IBM BlueEyes