Subspace Methods for Visual Learning and Recognition

Horst Bischof

Inst. f. Computer Graphics and Vision, Graz Univesity of Technology

Austria

bischof@icg.tu-graz.ac.at

Aleš Leonardis

Faculty of Computer and Information Science, University of Ljubljana

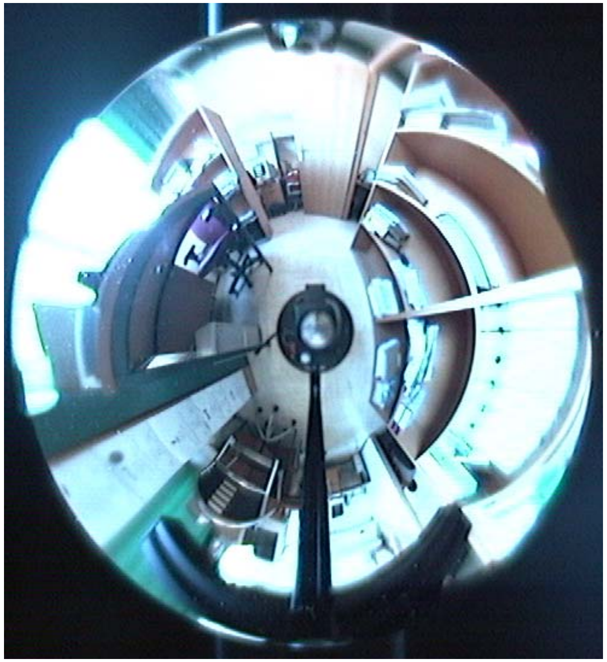
Slovenia

Ales.Leonardis@fri.uni-lj.si

Outline Part 1

- Motivation
- Appearance based learning and recognition
- Subspace methods for visual object recognition
- Principal Components Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Canonical Correlation Analysis (CCA)
- Independent Component Analysis (ICA)
- Non-negative Matrix Factorization (NMF)
- Kernel methods for non-linear subspaces

The name of the game



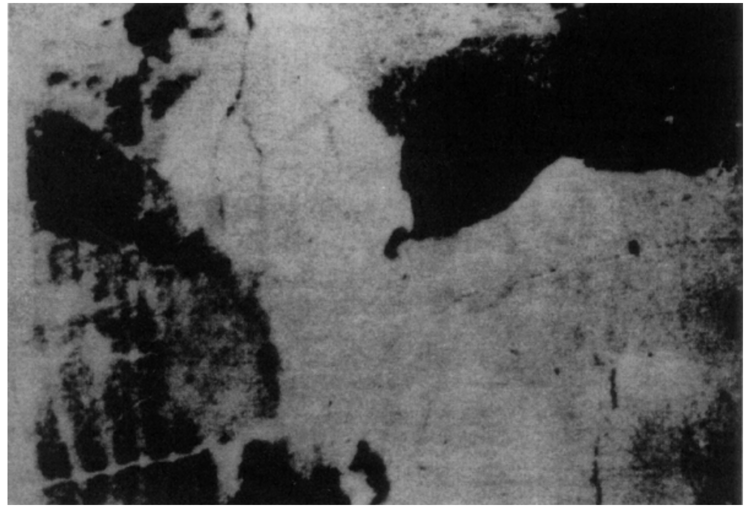
- complex objects/scenes
- varying pose (3D rotation, scale)
- cluttered background/foreground
- occlusions (noise)
- varying illumination

Object Representation

- High-Level Shape Models (e.g., Generalized Cylinders)
  - Idealized Images
  - Texture Less
- Mid-Level Shape Models (e.g. CAD models, Superquadrics)
  - More Complex
  - Well-defined geometry
- Low-level Appearance Based Models (e.g. Eigenspaces)
  - Most complex
  - Complicated shapes

Problems

Segmentation:



Pose/Shape:

a

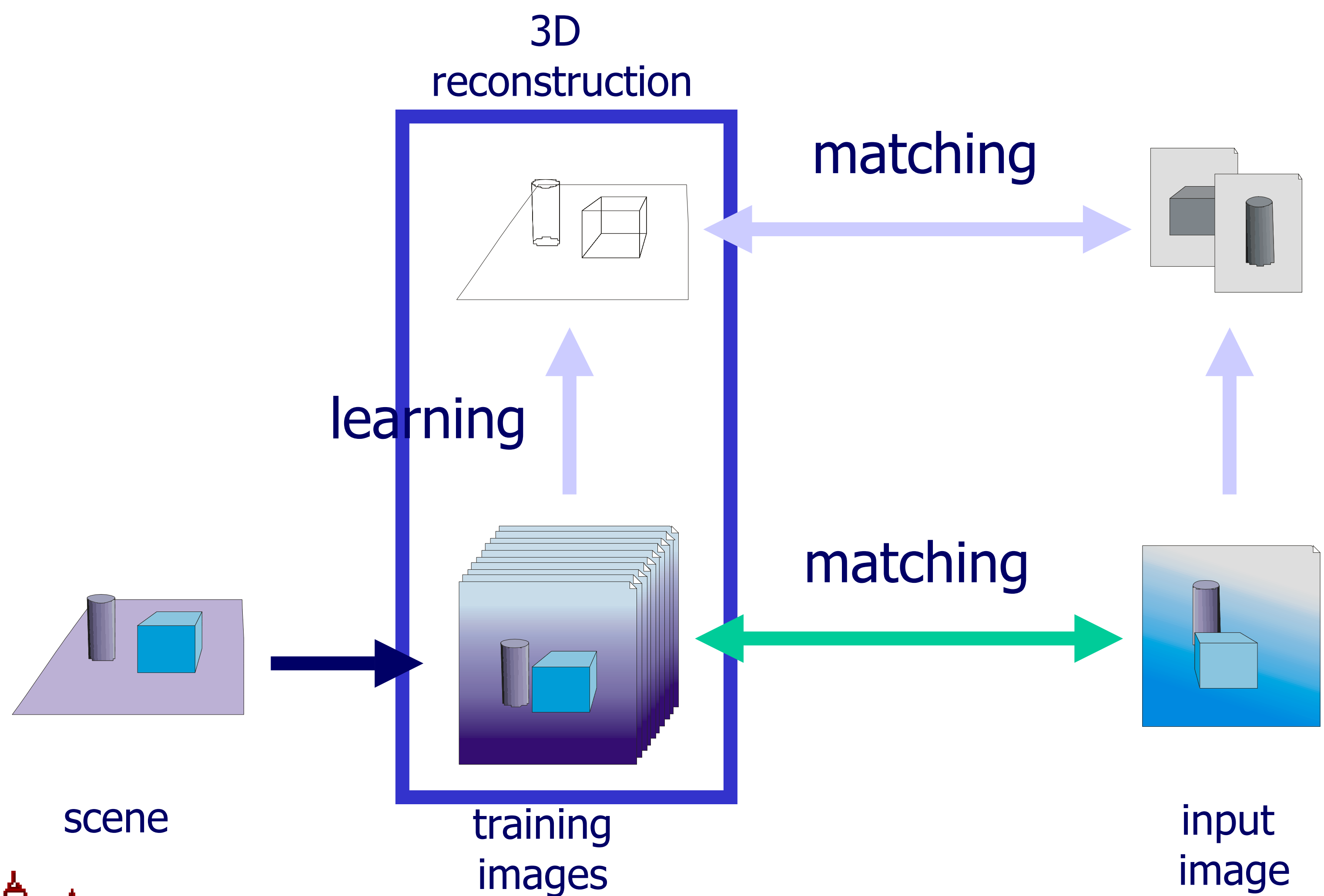
Illumination

Example

Example

Learning and recognition



Appearance-based approaches

A renewed attention in the appearance-based approaches

Encompass combined effects of:

- shape,
- reflectance properties,
- pose in the scene,
- illumination conditions.

Acquired through an automatic learning phase.

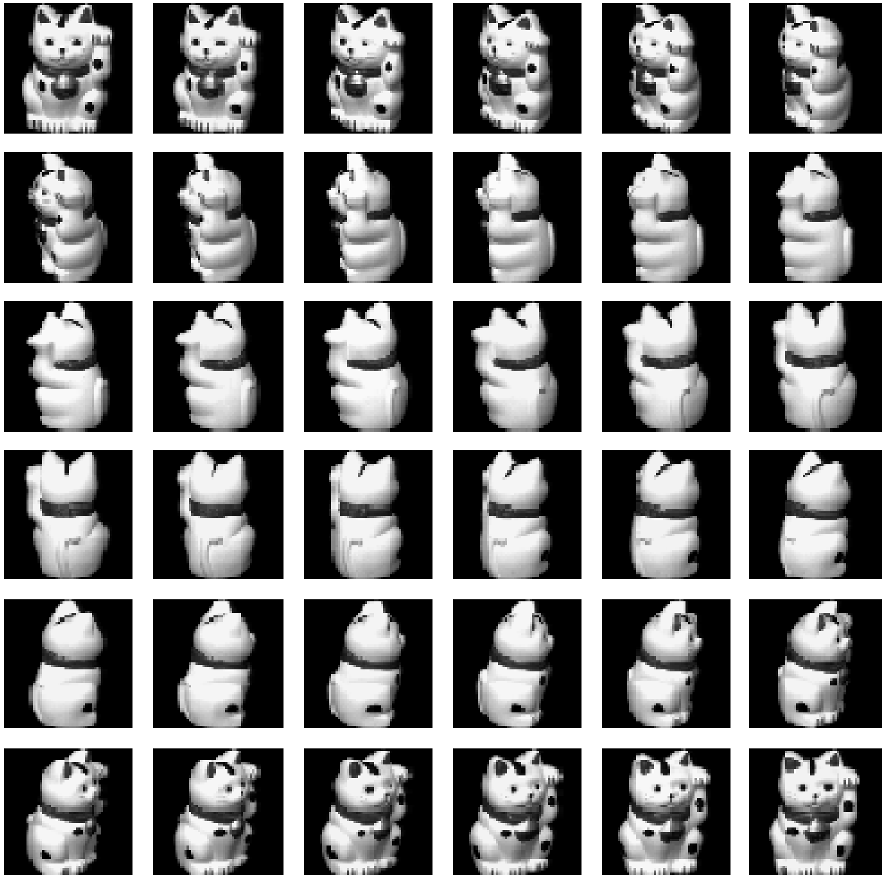
Appearance-based approaches

A variety of successful applications:

- Human face recognition e.g. [Beymer & Poggio, Turk & Pentland]
- Visual inspection e.g. [Yoshimura & Kanade]
- Visual positioning and tracking of robot manipulators, e.g. [Nayar & Murase]
- Tracking e.g., [Black & Jepson]
- Illumination planning e.g., [Murase & Nayar]
- Image spotting e.g., [Murase & Nayar]
- Mobile robot localization e.g., [Jogan & Leonardis]
- Background modeling e.g., [Oliver, Rosario & Pentland]

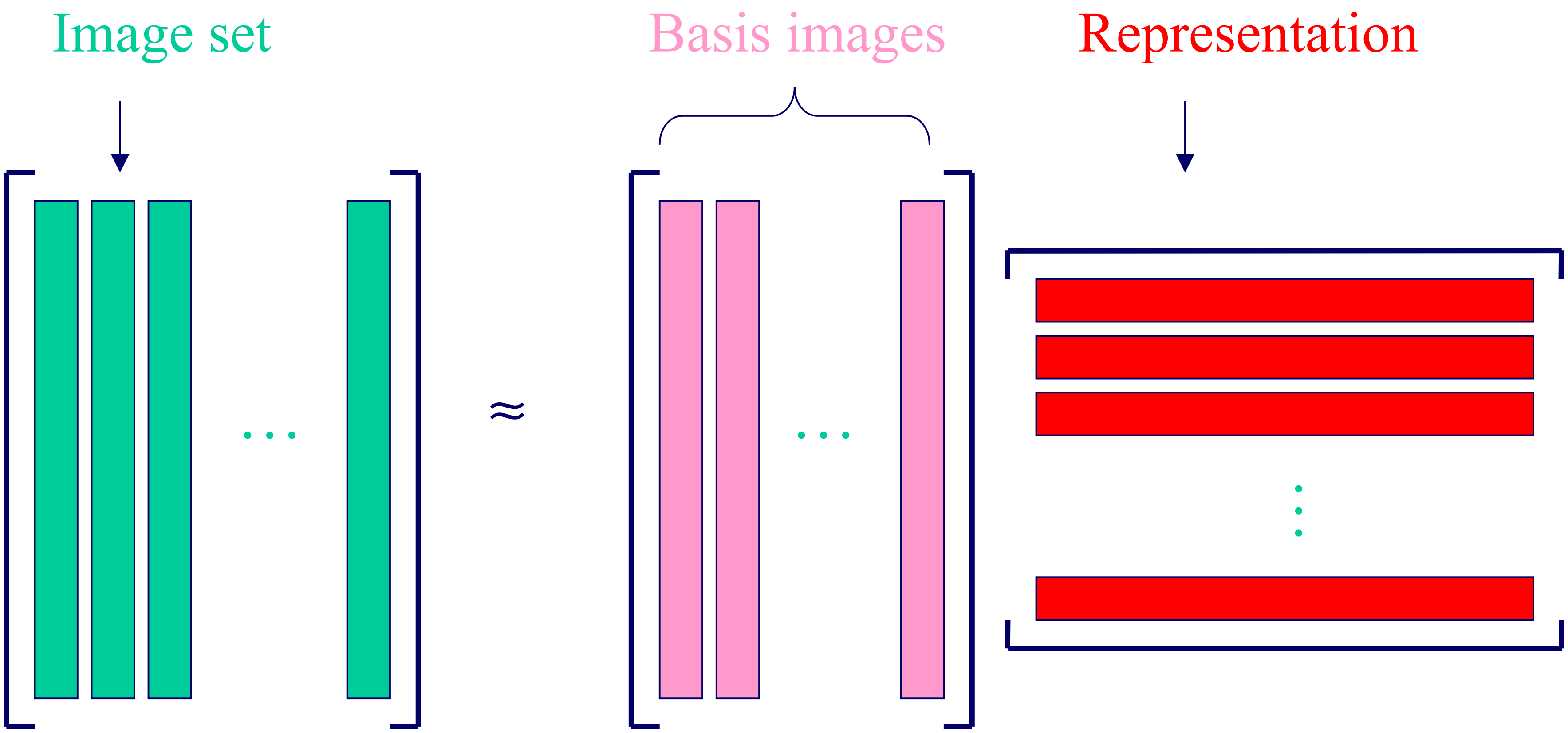
Appearance-based approaches

Objects are represented by a large number of views:



Subspace Methods

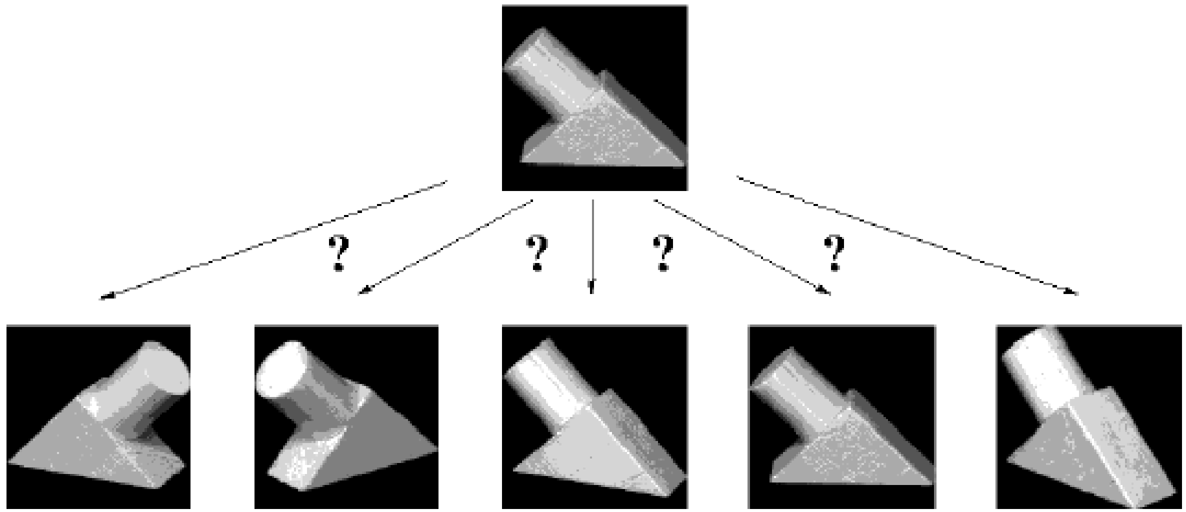
- Images are represented as points in the N-dimensional vector space
- Set of images populate only a small fraction of the space
- Characterize subspace spanned by images



Properties of the representation:

- Optimal Reconstruction $\Rightarrow$ PCA
- Optimal Separation $\Rightarrow$ LDA
- Optimal Correlation $\Rightarrow$ CCA
- Independent Factors $\Rightarrow$ ICA
- Non-negative Factors $\Rightarrow$ NMF
- Non-linear Extension $\Rightarrow$ Kernel Methods

Image Matching



$$\rho = \frac{\mathbf{x}^{T}\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} > \Theta$$

Normalized images $\|\mathbf{x}-\mathbf{y}\|^{2} < \Psi$

$\Rightarrow$ Compress images

Eigenspace representation

- Image set (normalised, zero-mean)

$$X = \begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \dots & \mathbf{x}_{n-1} \end{bmatrix}; \quad X \in \mathbb{R}^{m \times n}$$

- We are looking for orthonormal basis functions:

$$U = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \dots & \mathbf{u}_k \end{bmatrix}; \quad k << n$$

- Individual image is a linear combination of basis functions

$$\mathbf{x}_i \approx \tilde{\mathbf{x}}_i = \sum_{j=0}^{p} q_j(\mathbf{x}_i)\mathbf{u}_j$$

$$\| \mathbf{x} - \mathbf{y} \|^2 \approx \| \sum_{j=1}^{k} q_j(\mathbf{x})\mathbf{u}_j - \sum_{j=1}^{k} q_j(\mathbf{y})\mathbf{u}_j \|^2 =$$

$$\| \sum_{j=1}^{k} (q_j(\mathbf{x}) - q_j(\mathbf{y}))\mathbf{u}_j \|^2 = \| q_j(\mathbf{x}) - q_j(\mathbf{y}) \|^2$$

Best basis functions ν?

- **Optimisation problem**

$$\sum_{i=0}^{n-1} ||\mathbf{x}_i - \sum_{j=0}^{k} q_j(\mathbf{x}_i)\mathbf{u}_j||^2 \rightarrow \min$$

- **Taking the *k* eigenvectors with the largest eigenvalues of**

$$C = XX^T = \left[\begin{array}{cccc} \mathbf{x}_0 & \mathbf{x}_1 & \dots & \mathbf{x}_{n-1} \end{array} \right] \left[\begin{array}{c} \mathbf{x}_0^\top \\ \mathbf{x}_1^\top \\ \dots \\ \mathbf{x}_{n-1}^\top \end{array} \right]$$

- **PCA or Karhunen-Loéve Transform (KLT)**

$$C\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

Efficient eigenspace computation

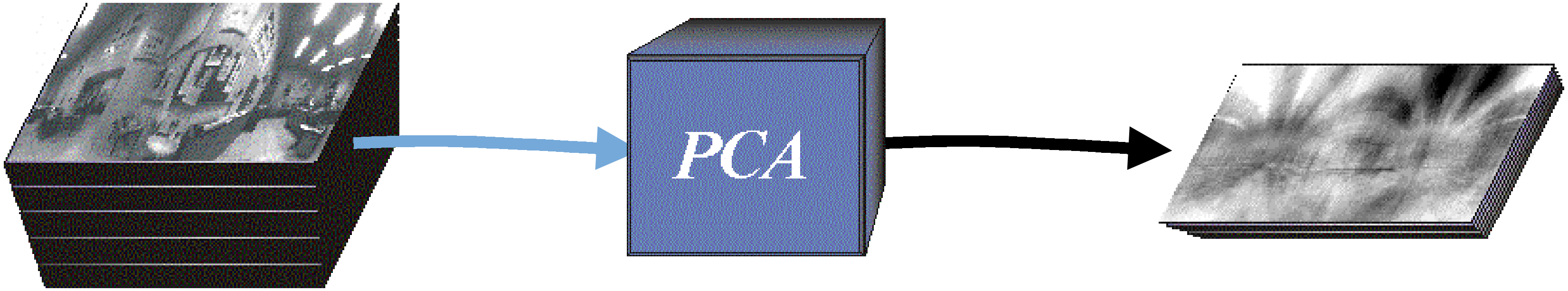
- *n* << *m*
- Compute the eigenvectors u'_i, $i = 0,\ldots,n-1$, of the inner product matrix

$$Q = X^\top X = \begin{bmatrix} \mathbf{x}_0^\top \\ \mathbf{x}_1^\top \\ \cdots \\ \mathbf{x}_{n-1}^\top \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \ldots & \mathbf{x}_{n-1} \end{bmatrix}; \quad Q \in \mathbb{R}^{n \times n}$$

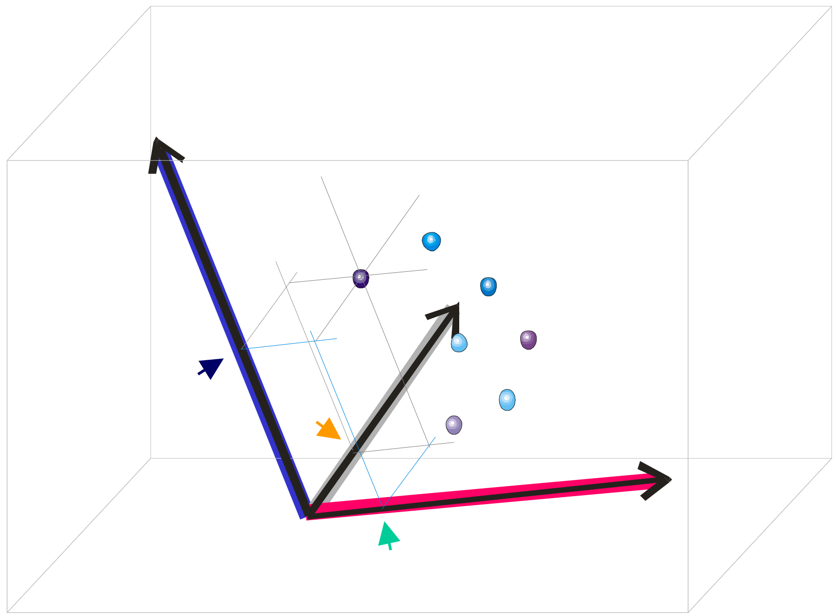
- The eigenvectors of $XX^\top$ can be obtained by using $XX^\top X v_i' = \lambda_i' X v_i'$:

$$\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i'}} X \mathbf{u}_i'$$

Principal Component Analysis



Principal Component Analysis



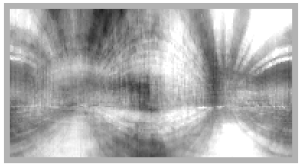
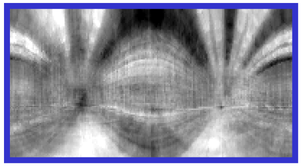
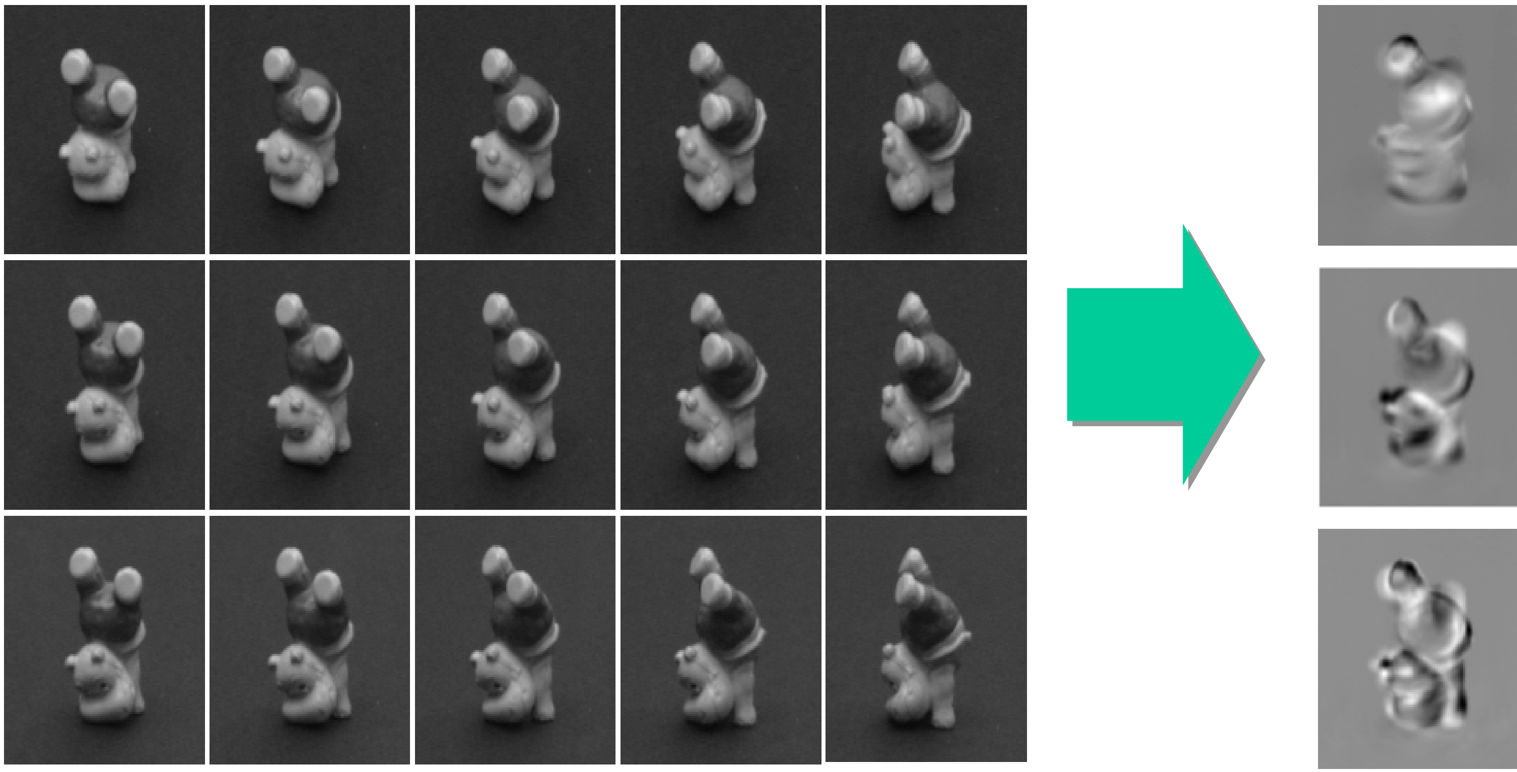
 $= q_1 \cdot$  $+ q_2 \cdot$  $+ q_3 \cdot$  $+ \ldots$

Image representation with PCA



$\mathbf{u}_1$

$\mathbf{u}_2$

$\mathbf{u}_3$

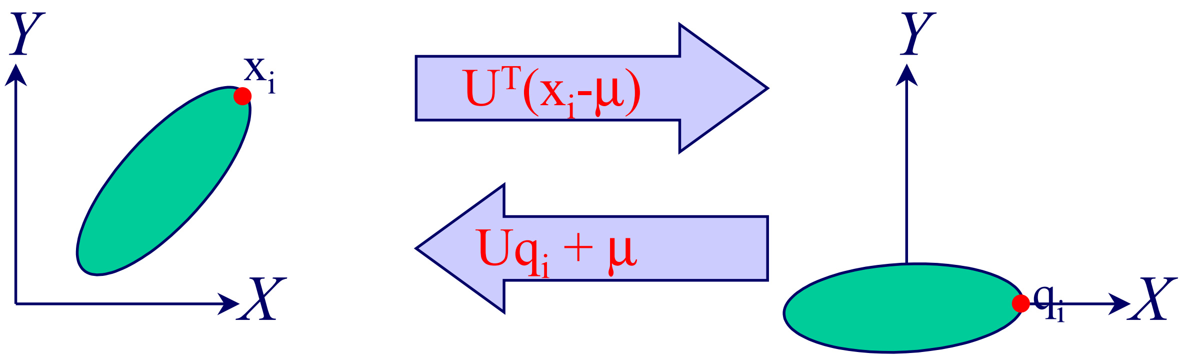
Properties PCA

- Any point x_i can be projected to an appropriate point q_i by :

$$q_i = U^T(x_i - \mu)$$

- and conversely (since $U^{-1} = U^T$)

$$Uq_i + \mu = x_i$$



Properties PCA

- It can be shown that the mean square error between x_i and its reconstruction using only m principle eigenvectors is given by the expression :

$$\boxed{\sum_{j=1}^{N} \lambda_j - \sum_{j=1}^{m} \lambda_j = \sum_{j=m+1}^{N} \lambda_j}$$

- PCA minimizes reconstruction error

- PCA maximizes variance of projection

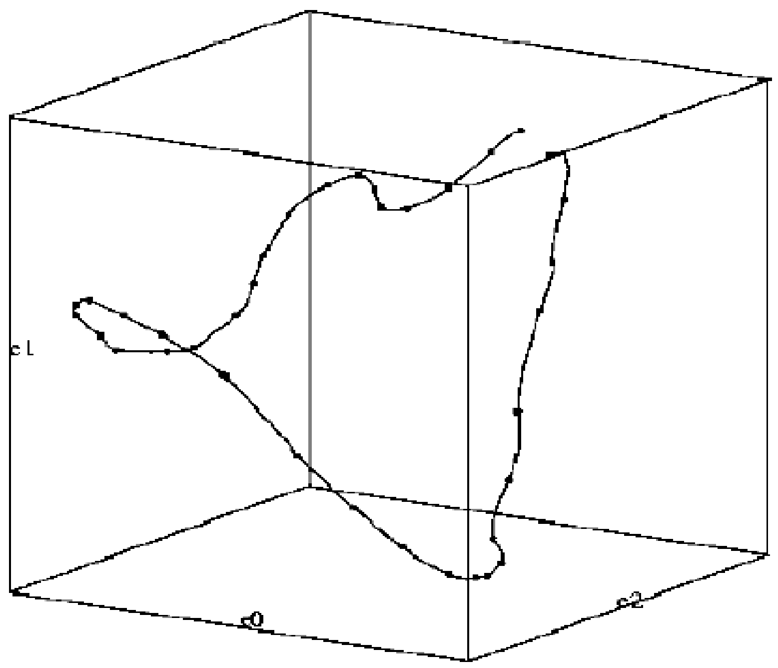
- Finds a more “natural” coordinate system for the sample data.

PCA for visual recognition and pose estimation

Objects are represented as coordinates in an n-dimensional eigenspace.

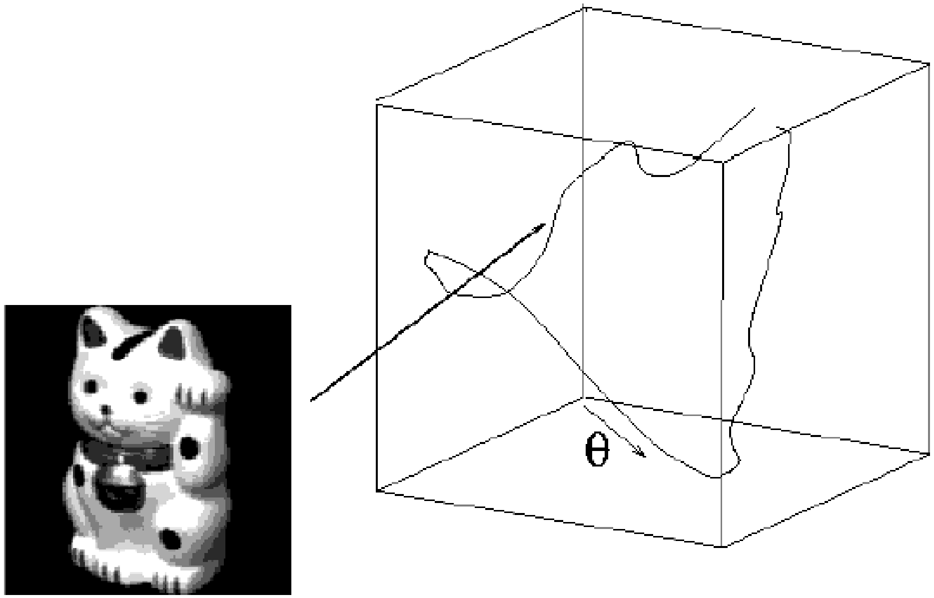
An example:

> 3-D space with points representing individual objects or a manifold representing **parametric eigenspace** (e.g., orientation, pose, illumination).



PCA for visual recognition and pose estimation

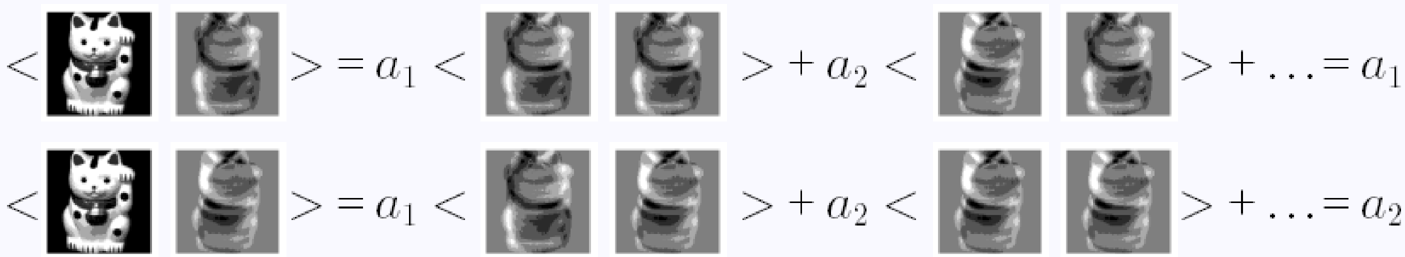
- Calculate coefficients
- Search for the nearest point (individual or on the curve)
- Point determines object and/or pose



Calculation of coefficients

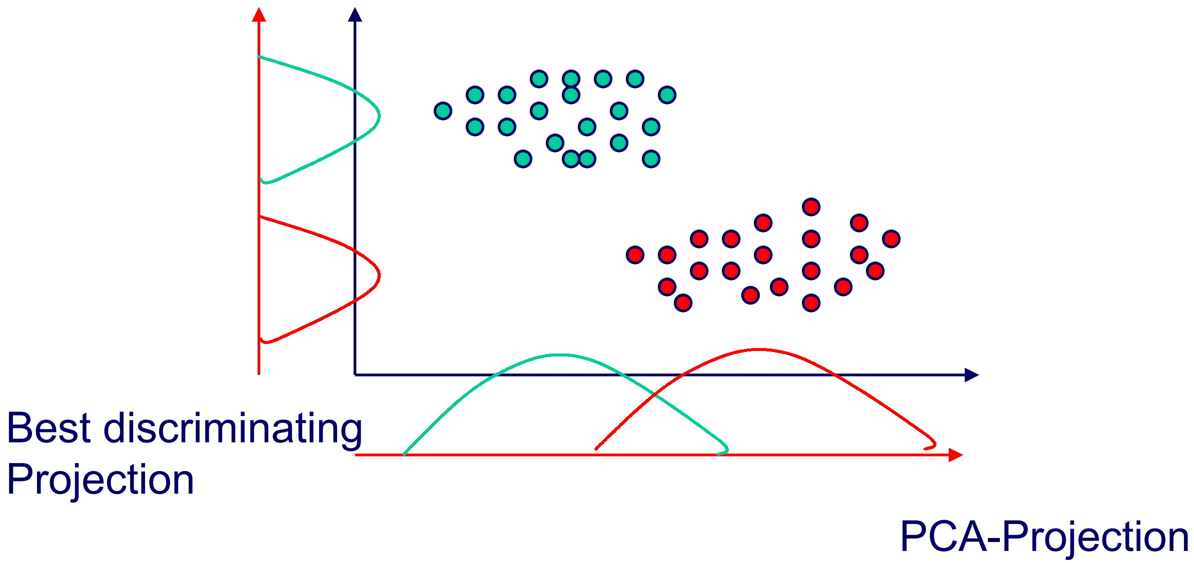
To recover a_i the image is projected onto the eigenspace

$$a_i(\mathbf{x}) = <\mathbf{x}, \mathbf{e}_i> = \sum_{j=1}^{m} x_j e_{i_j} \qquad 1 \leq i \leq p$$



- Complete image x_i is required to calculate a_i.
- Corresponds to Least-Squares Solution

- **PCA minimizes projection error**



- **PCA is „unsupervised“ no information on classes is used**
- **Discriminating information might be lost**

- **Linear Discriminance Analysis (LDA)**

  – Maximize distance between classes
  – Minimize distance within a class

⇨ Fisher Linear Discriminance

$$\rho(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S_B} \mathbf{w}}{\mathbf{w}^T \mathbf{S_W} \mathbf{w}}$$

LDA: Problem formulation

- **n Sample images:** $\{x_1, \cdots, x_n\}$
- **c classes:** $\{\chi_1, \cdots, \chi_c\}$
- **Average of each class:** $$\mu_i = \frac{1}{n_i} \sum_{x_k \in \chi_i} x_k$$
- **Total average:** $$\mu = \frac{1}{n} \sum_{k=1}^{N} x_k$$

LDA: Practice

- **Scatter of class i:**

$$S_i = \sum_{x_k \in \chi_i} (x_k - \mu_i)(x_k - \mu_i)^T$$

- **Within class scatter:**

$$S_W = \sum_{i=1}^{c} S_i$$

- **Between class scatter:**

$$S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$$

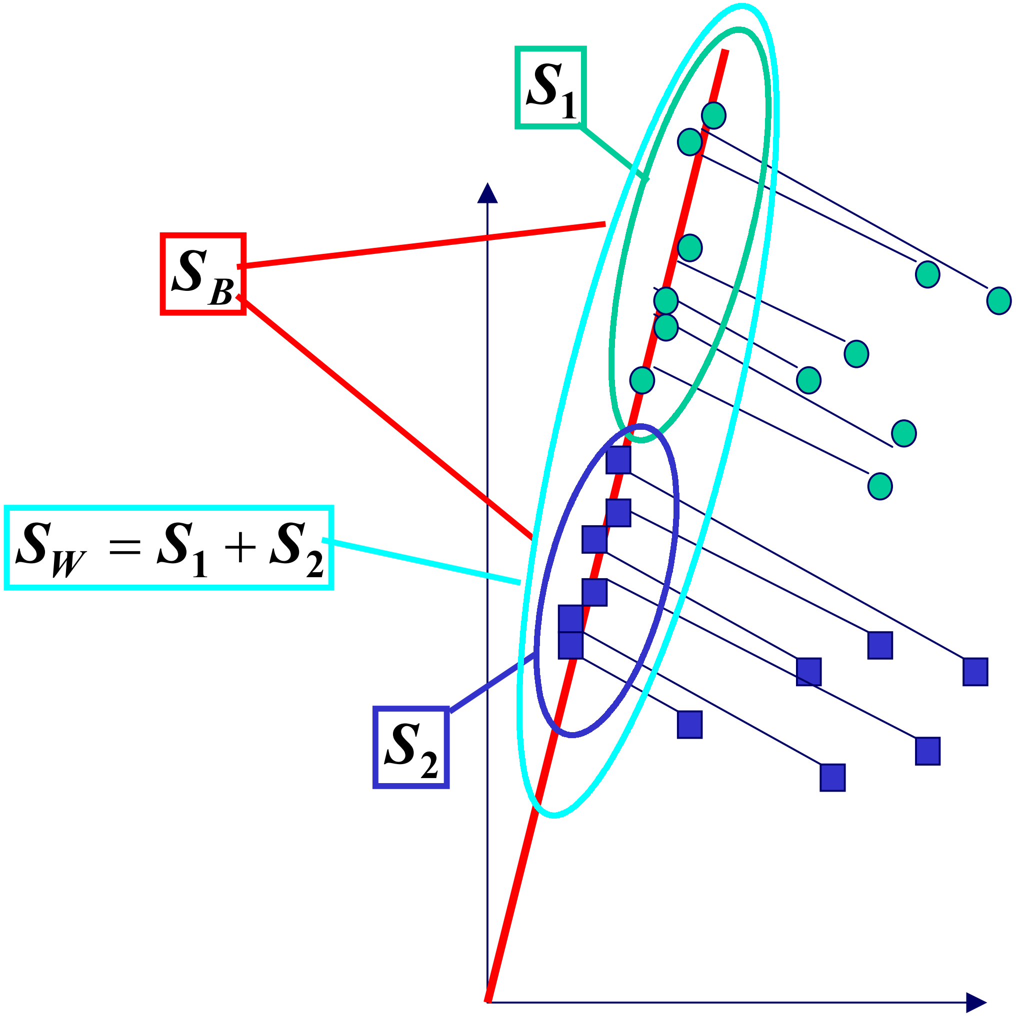
- **Total scatter:**

$$S_T = S_W + S_B$$

LDA: Practice

- **After projection:** $$y_k = W^T x_k$$
  - Between class scatter (of y's): $$\tilde{S}_B = W^T S_B W$$
  - Within class scatter (of y's): $$\tilde{S}_W = W^T S_W W$$

LDA



Good separation

LDA

- Maximization of

$$\rho(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S_B} \mathbf{w}}{\mathbf{w}^T \mathbf{S_W} \mathbf{w}}$$

- is given by solution of generalized eigenvalue problem

$$\mathbf{S_B}\mathbf{w} = \lambda \mathbf{S_w}\mathbf{w}$$

- For the c-class case we obtain (at most) c-1 projections as the largest eigenvalues of

$$\mathbf{S_B}\mathbf{w}_i = \lambda \mathbf{S_w}\mathbf{w}_i$$

LDA

- **How to calculate LDA for high-dimensional images?**

- **Problem: S_W is always singular**
  - Number of pixels in each image is larger than the number of images in the training set

1. **Fischerfaces → Reduce dimension by PCA and then perform LDA**

2. **Simultaneous diagonalization of S_W and S_B**

LDA

- Fischerfaces (Belhumeur et.al. 1997)

- Reduce dimensionality to n-c with PCA

$$\mathbf{U}_{pca} = \arg\max_{U} |\mathbf{U}^{\mathbf{T}}\mathbf{Q}\mathbf{U}| = [\mathbf{u_1}\ \mathbf{u_2}\ \ldots\ \mathbf{u_{n-c}}]$$

- Further reduce to c-1 with FLD

$$\mathbf{W}_{fld} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W^T W_{pca}^T S_B W_{pca} W}|}{|\mathbf{W^T W_{pca}^T S_W W_{pca} W}|} = [\mathbf{w_1}\ \mathbf{w_2}\ \ldots\ \mathbf{w_{c-1}}]$$

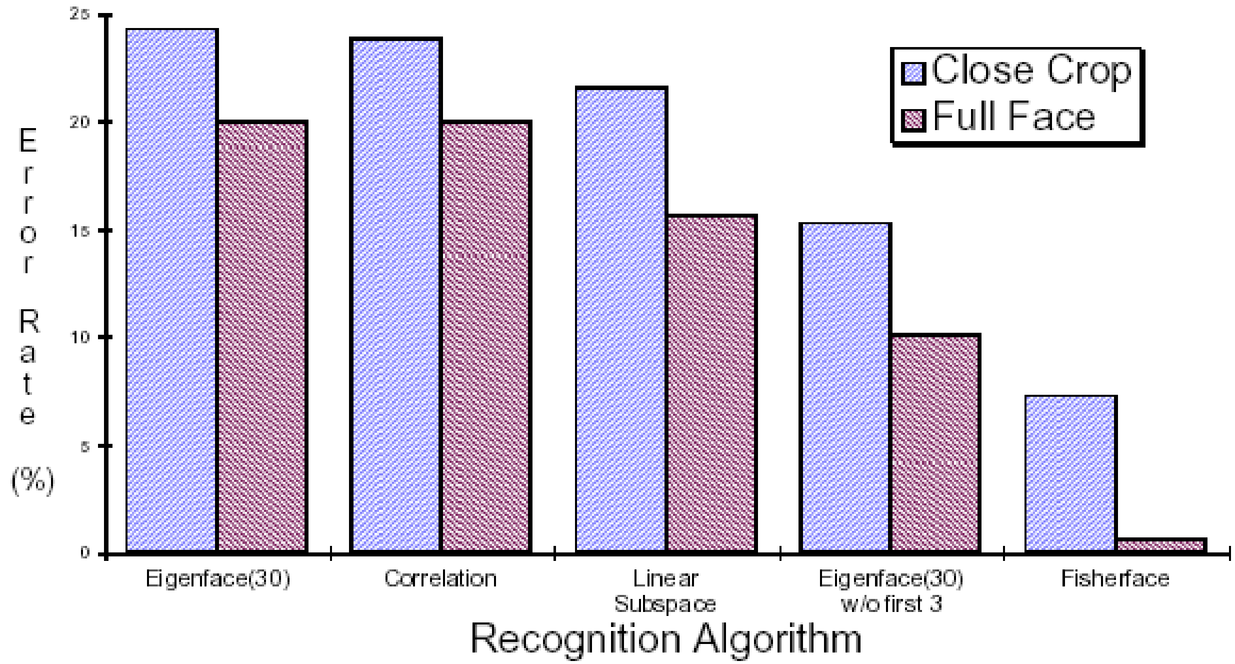
- The optimal projection becomes

$$\mathbf{W}_{opt} = \mathbf{W}_{fld}^{\mathbf{T}}\mathbf{U}^{\mathbf{T}}$$

- Example Fisherface of recognition Glasses/NoGlasses (Belhumeur et.al. 1997)

LDA

- **Example comparison for face recognition (Belhumeur et.al. 1997)**



- **Superior performance than PCA for face recognition**
- **Noise sensitive**
- **Requires larger training set, more sensitive to different training data [Martinez&Kak2001]**

Canonical Correlation Analysis (CCA)

- **Also „supervised“ method but motivated by regression tasks, e.g. pose estimation.**
- **Canonical Correlation Analysis relates two sets of observations by determining pairs of directions that yield maximum correlation between these sets.**
- **Find a pair of directions (canonical factors)** $w_x \in \Re^p$, $w_y \in \Re^q$, **so that the correlation of the projections c =** $w_x^T x$ **and d =** $w_y^T y$ **becomes maximal.**

What is CCA?

Canonical
Correlation
$0 \leq r \leq 1$

$$\rho = \frac{E[cd]}{\sqrt{E[c^2]E[d^2]}} =$$

$$\frac{E[\mathbf{w}_x^T \mathbf{x}\mathbf{y}^T \mathbf{w}_y]}{\sqrt{E[\mathbf{w}_x^T \mathbf{x}\mathbf{x}^T \mathbf{w}_x]E[\mathbf{w}_y^T \mathbf{y}\mathbf{y}^T \mathbf{w}_y]}} =$$

Between Set
Covariance
Matrix

$$\frac{\mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y}{\sqrt{\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x \mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y}}$$

- Finding solutions

$$\mathbf{w} = \begin{pmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{pmatrix} \qquad \mathbf{A} = \begin{pmatrix} 0 & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & 0 \end{pmatrix}, \mathbf{B} = \begin{pmatrix} \mathbf{C}_{xx} & 0 \\ 0 & \mathbf{C}_{yy} \end{pmatrix}$$

Rayleigh Quotient

$$r = \frac{\mathbf{w}^T \mathbf{A} \mathbf{w}}{\mathbf{w}^T \mathbf{B} \mathbf{w}}$$

Generalized Eigenproblem

$$\mathbf{A}\mathbf{w} = \mu \mathbf{B}\mathbf{w}$$

CCA for images

- Same problem as for LDA
- Computationally efficient algorithm based on SVD

$$\mathbf{A} = \mathbf{C}_{xx}^{-\frac{1}{2}} \mathbf{C}_{xy} \mathbf{C}_{yy}^{-\frac{1}{2}}$$

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{T}$$

$$\mathbf{w}_{xi} = \mathbf{C}_{xx}^{-\frac{1}{2}} \mathbf{u}_{i}$$

$$\mathbf{w}_{yi} = \mathbf{C}_{yy}^{-\frac{1}{2}} \mathbf{v}_{i}$$

Properties of CCA

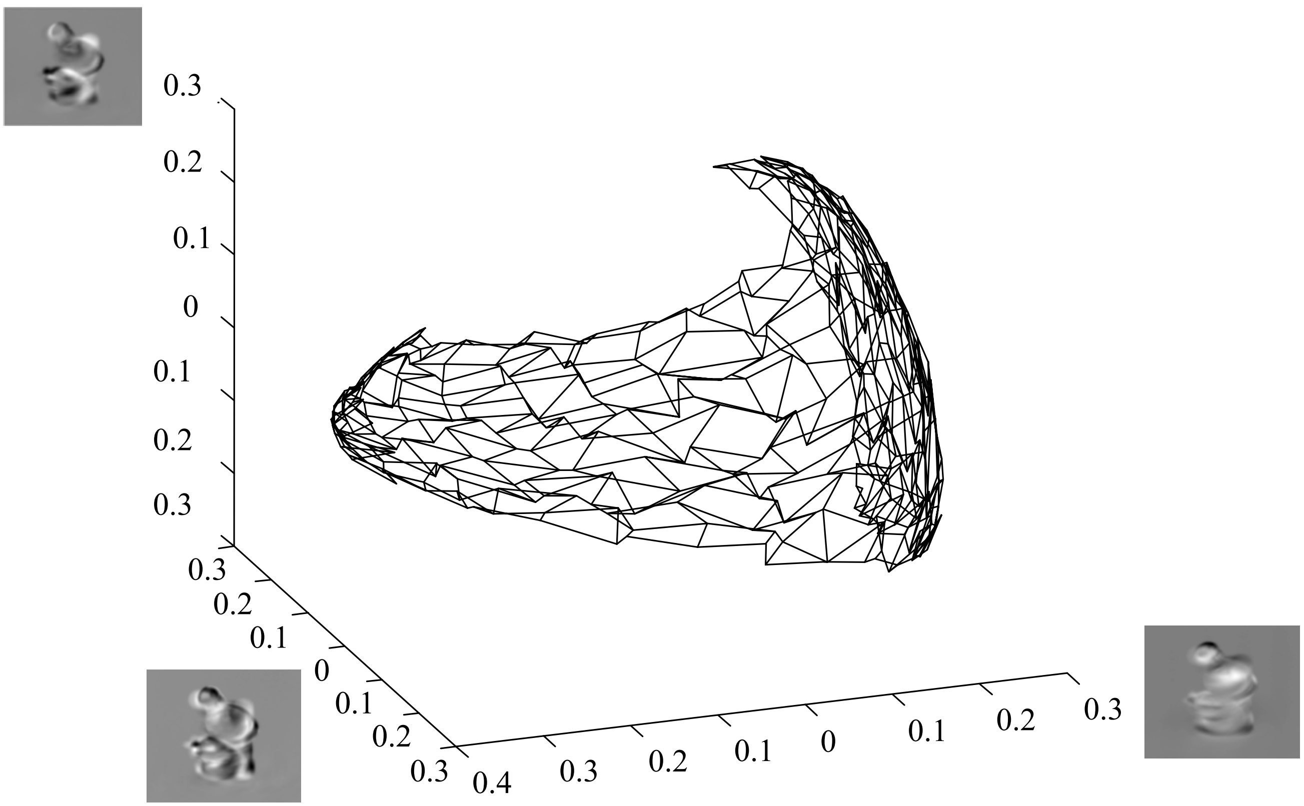
- At most min(p,q,n) CCA factors
- Invariance w.r.t. affine transformations
- Orthogonality of the Canonical factors

$$\mathbf{w}_{xi}^{T}\mathbf{C}_{xx}\mathbf{w}_{xj} = \mathbf{0}$$

$$\mathbf{w}_{yi}^{T}\mathbf{C}_{yy}\mathbf{w}_{yj} = \mathbf{0}$$

$$\mathbf{w}_{xi}^{T}\mathbf{C}_{xy}\mathbf{w}_{yj} = \mathbf{0}$$

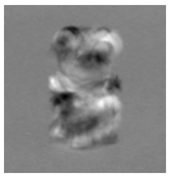
Parametric eigenspace obtained by PCA for 2DoF in pose

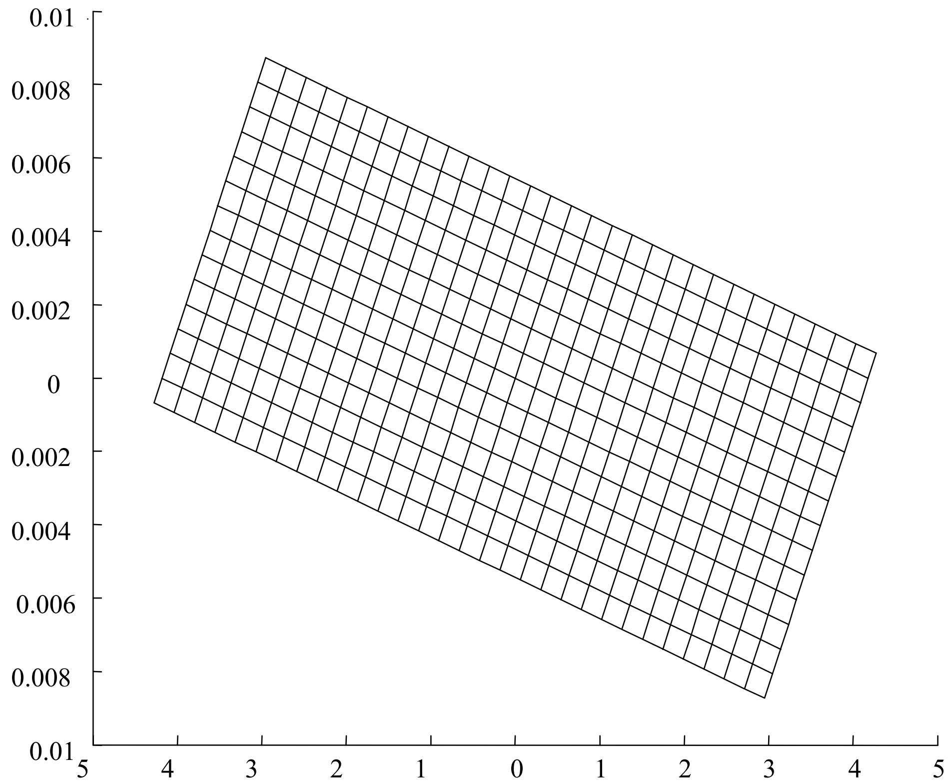


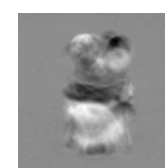
CCA Example

CCA representation

(projections of training images onto $\mathbf{w}_{x1}$, $\mathbf{w}_{x2}$)







Independent Component Analysis (ICA)

- **ICA is a powerful technique from signal processing (Blind Source Separation)**
- **Can be seen as an extension of PCA**
- **PCA takes into account only statistics up to 2nd order**
- **ICA finds components that are statistically independent (or as independent as possible)**

→ Local descriptors, sparse coding

Independent Component Analysis (ICA)

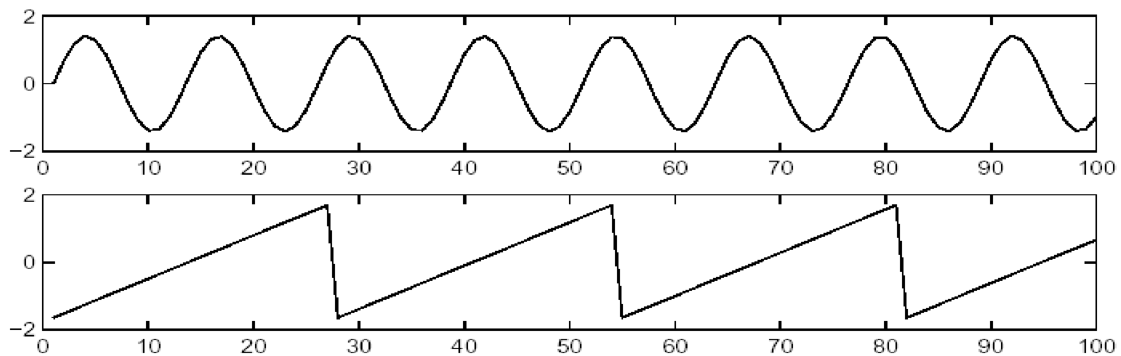
- m scalar variables $X=(x_1 \ldots x_m)^T$
- They are assumed to be obtained as linear mixtures of n sources $S=(s_1 \ldots s_n)^T$

$$\mathbf{X} = \mathbf{AS}$$

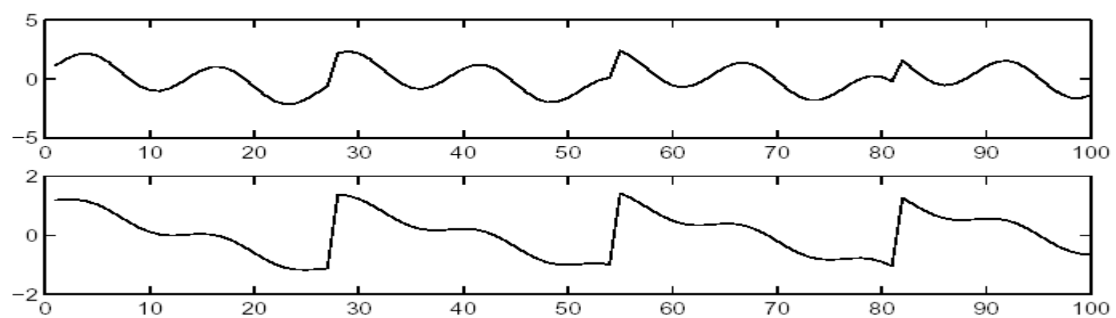
- Task: Given X find A, S (under the assumption that S are independent)

ICA Example

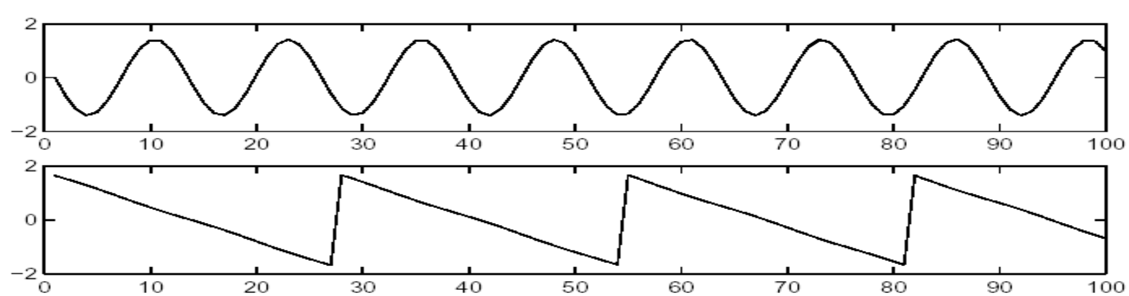
Original Sources



Mixtures

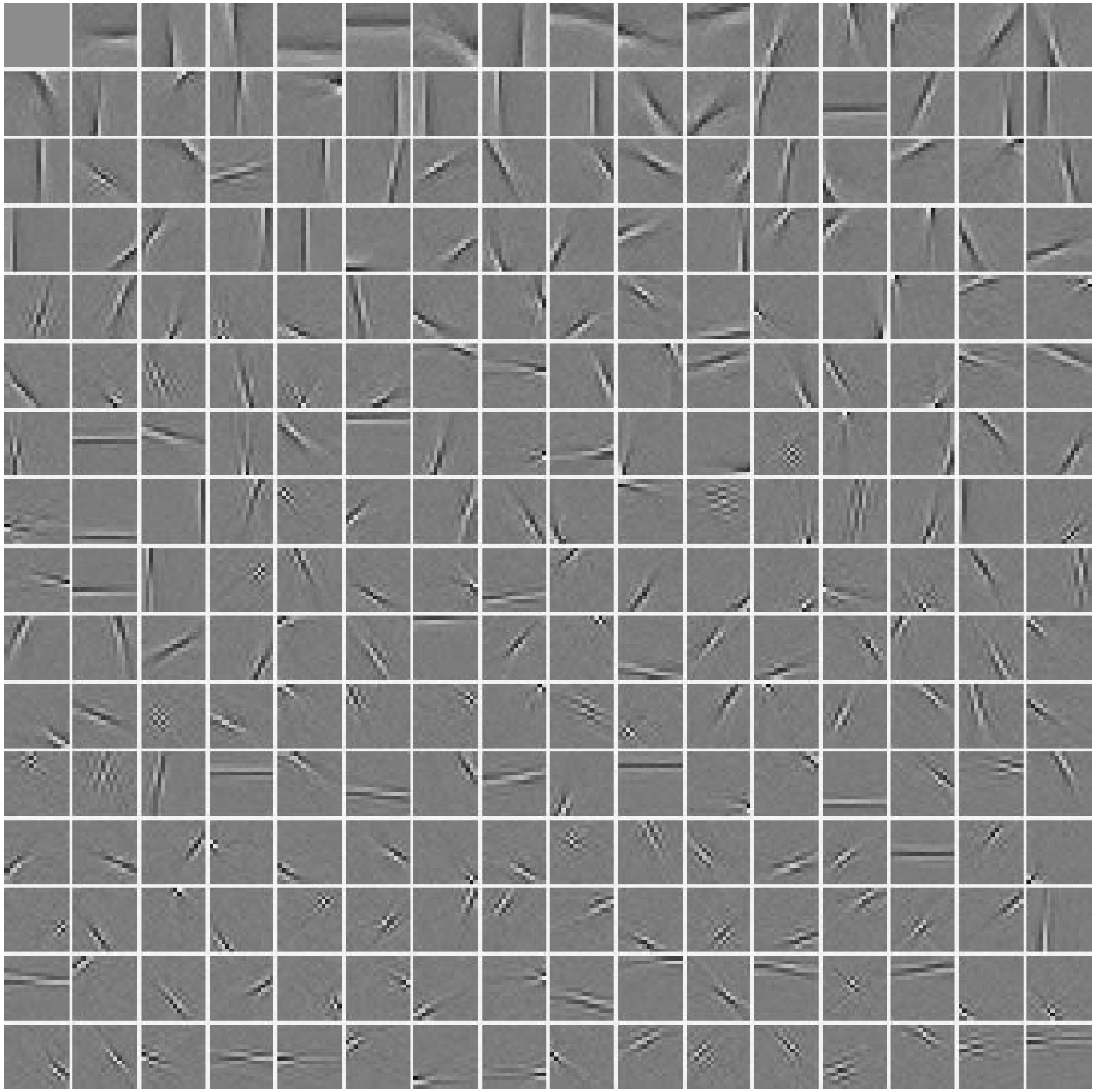


Recovered Sources



ICA Example

ICA basis obtained from 16x16 patches of natural images (Bell&Sejnowski 96)



ICA Algorithms

1. Minimize (Maximize) function
   - Complex matrix (tensor) functions

2. Adaptive Algorithms based on stochastic gradient
   - Measure of independence
   - Non-Gaussian, e.g. Kurtosis, Negentropy

   - Fast ICA Algorithm (Hyvärinen)

$$1. \qquad \mathbf{W} = \mathbf{W} / \sqrt{\| \mathbf{W}\mathbf{W}^T \|}$$

Repeat until convergence

$$2. \qquad \mathbf{W} = \frac{3}{2}\mathbf{W} - \frac{1}{2}\mathbf{W}\mathbf{W}^T\mathbf{W}$$

ICA Properties

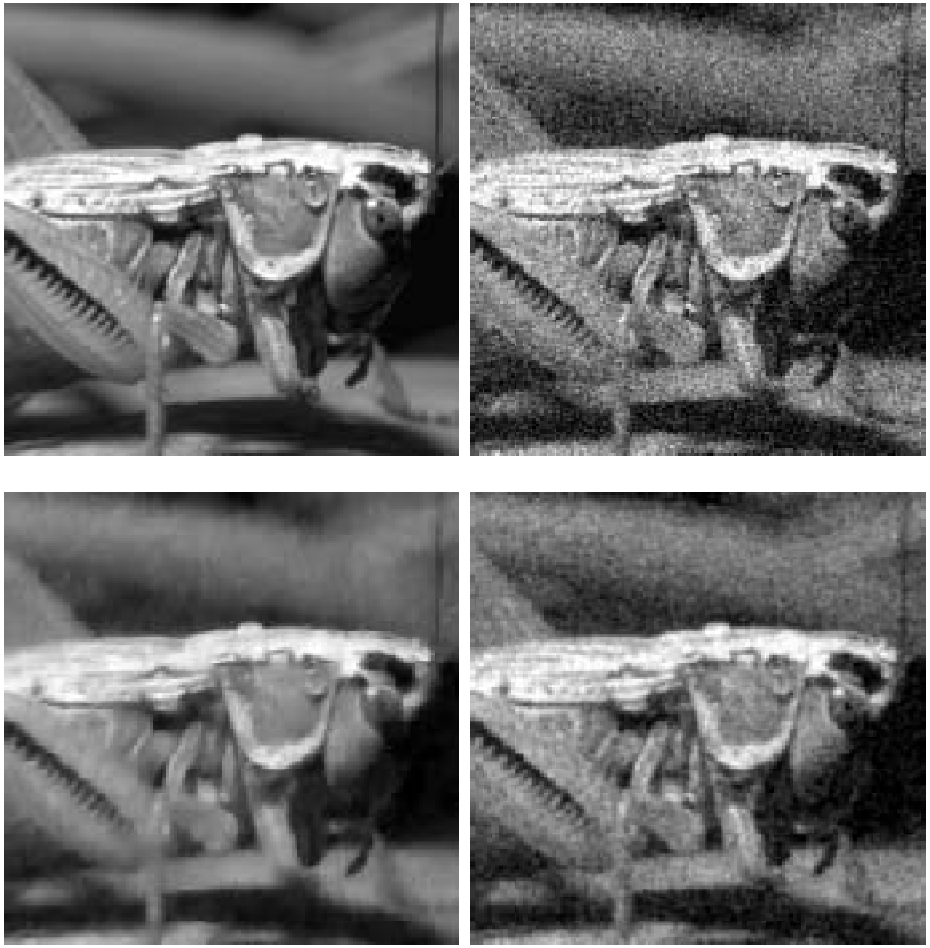
- ICA works only for Non-Gaussian Sources
- Usually centering and Whiteing of data is performed
- We can not measure the variance of the components

$$\mathbf{X} = \mathbf{A}\mathbf{P}^{-1}\mathbf{P}\mathbf{S}$$

- ICA does not provide ordering
- ICA components are not orthogonal

ICA for Noise Suppression

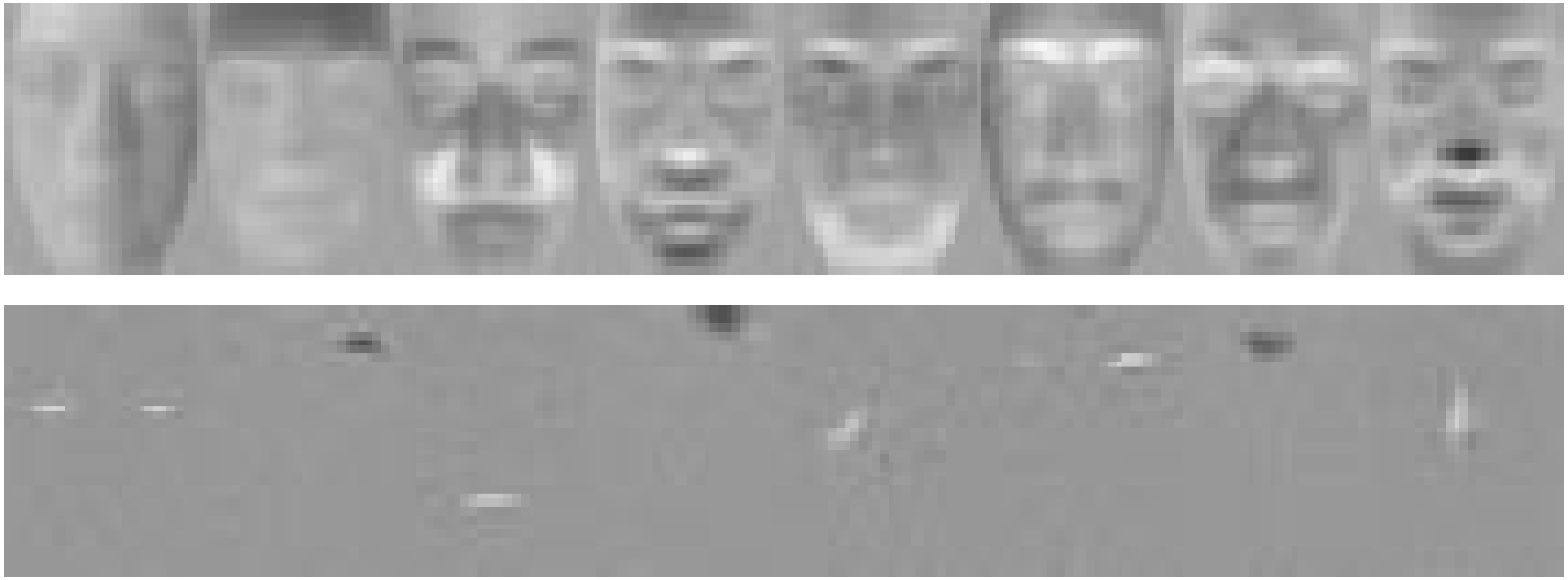
- **Sparse Code Shrinkage (similar to Wavelet Shrinkage Hyvärinen 99)**



Face Recognition using ICA

- **PCA vs. ICA on Ferret DB (Baek et.al. 02)**

PCA



ICA

Non-Negative Matrix Factorization (NMF)

- How can we obtain part-based representation?
- Local representation where parts are added
- E.g. learn from a set of faces the parts a face consists of, i.e. eyes, nose, mouth, etc.
- Non-Negative Matrix Factorization (Lee & Seung 1999) lead to part based representation

Matrix Factorization - Constraints

$V \approx WH$

- **PCA** : *W* are orthonormal basis vectors

$$W = [\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_n], \qquad \vec{w}_i \cdot \vec{w}_j = \delta_{ij}$$

- **VQ** : *H* are unity vectors

$$H = [\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_n], \qquad \vec{h}_j^{\,T} = [0,0,1,0,\cdots,0]$$

- **NMF** : *V,W,H* are non-negative

$$V_{ij}, W_{ij}, H_{ij} \geq 0 \qquad \forall i, j$$

NMF - Cost functions

- **Euclidean distance between *A* and *B***

$$\left\| A - B \right\|^2 = \sum_{ij} (A_{ij} - B_{ij})^2$$

- **Divergence of *A* from *B* (Relative entropy)**

$$D(A \| B) = \sum_{ij} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right)$$

NMF - update rules

- $\|V - WH\|^2$ **is non-increasing under**

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}} \qquad W_{i\mu} \leftarrow W_{i\mu} \frac{(V H^T)_{i\mu}}{(W H H^T)_{i\mu}}$$

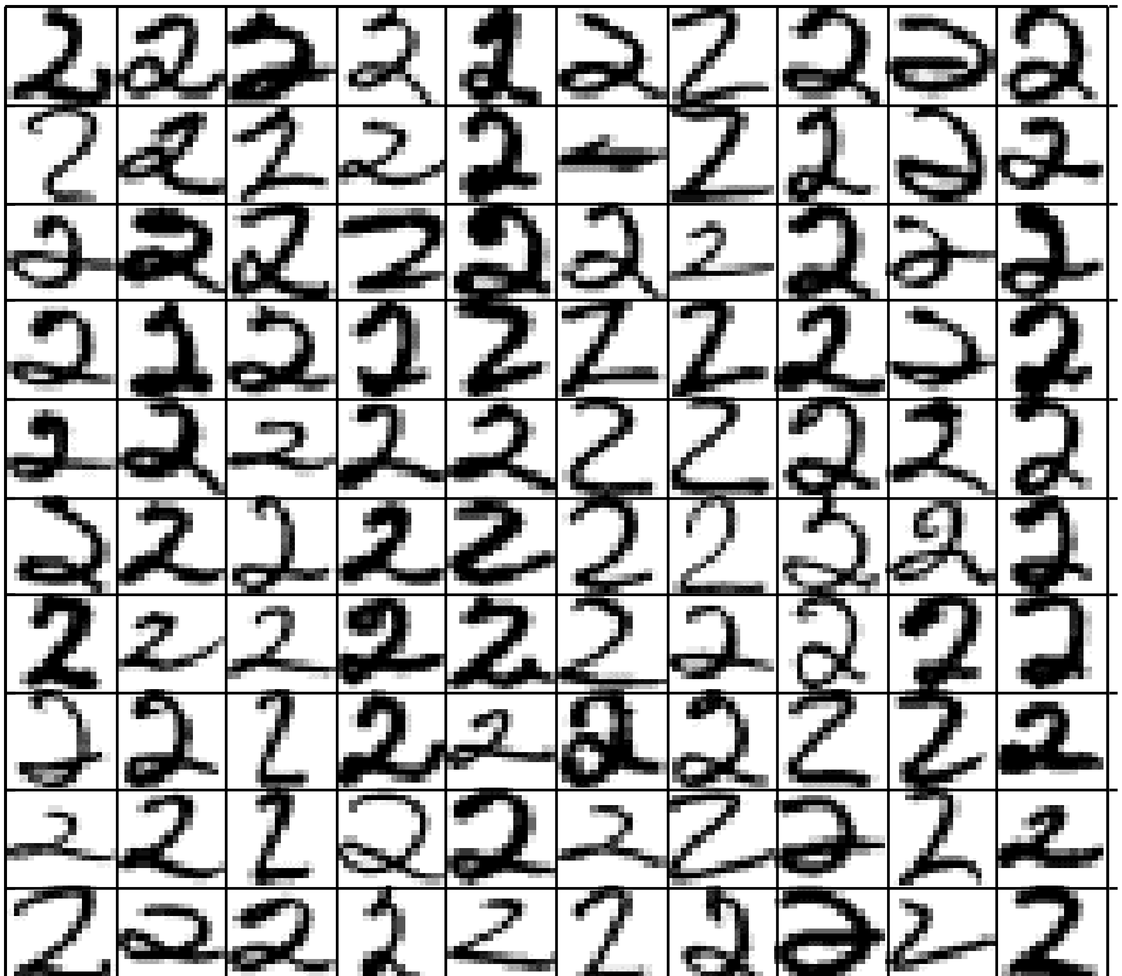
- $D(V \| WH)$ **is non-increasing under**

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu} / (WH)_{i\mu}}{\sum_k W_{ka}} \qquad W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} V_{i\mu} / (WH)_{i\mu}}{\sum_\nu W_{a\nu}}$$

- **We can start with random matrices for W and H and update each matrix iteratively until W and H are at a stationary point - the cost functions are invariant at this point.**

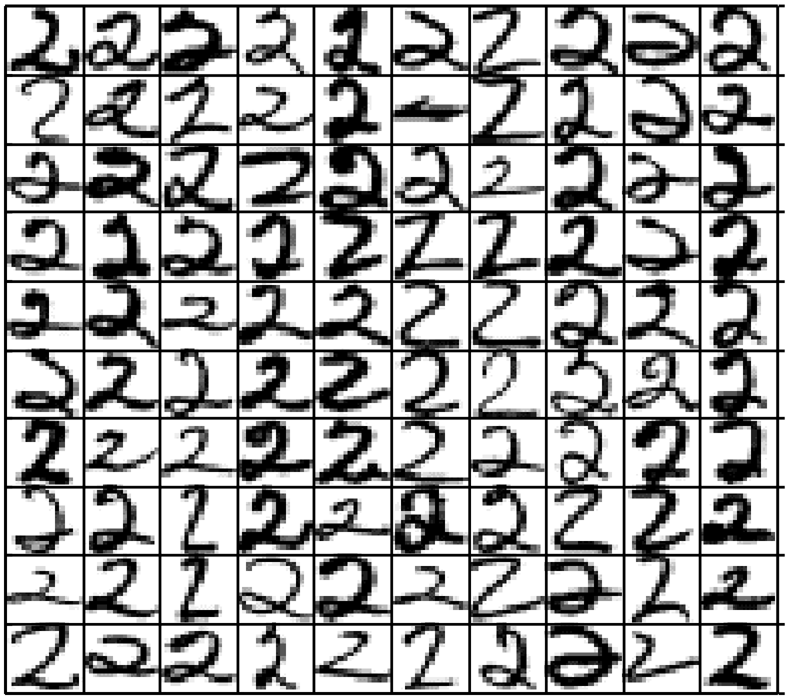
Concrete example – Handwritten Digits

- **Training data set for learning process**

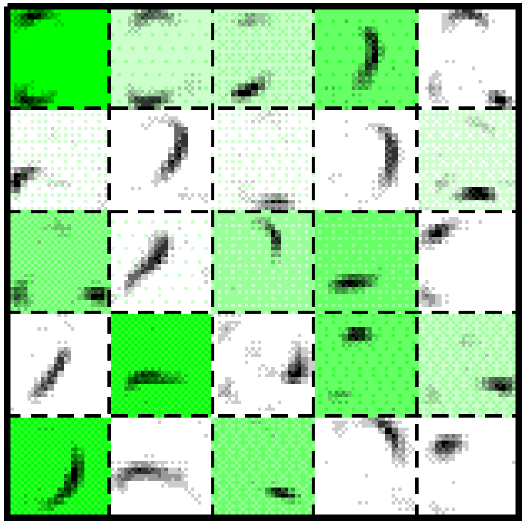


Learning

Find basis images from the training set

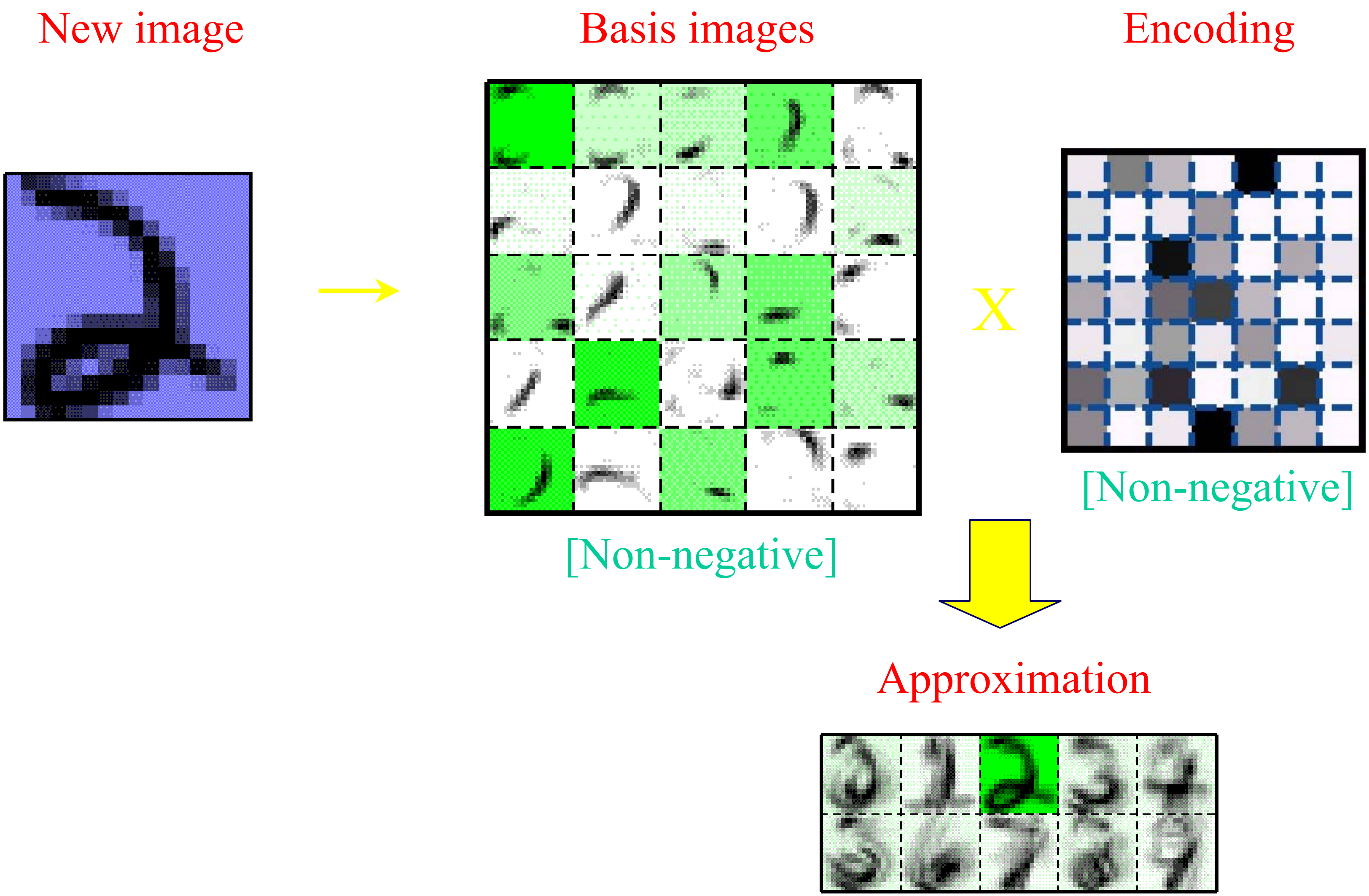


Training images

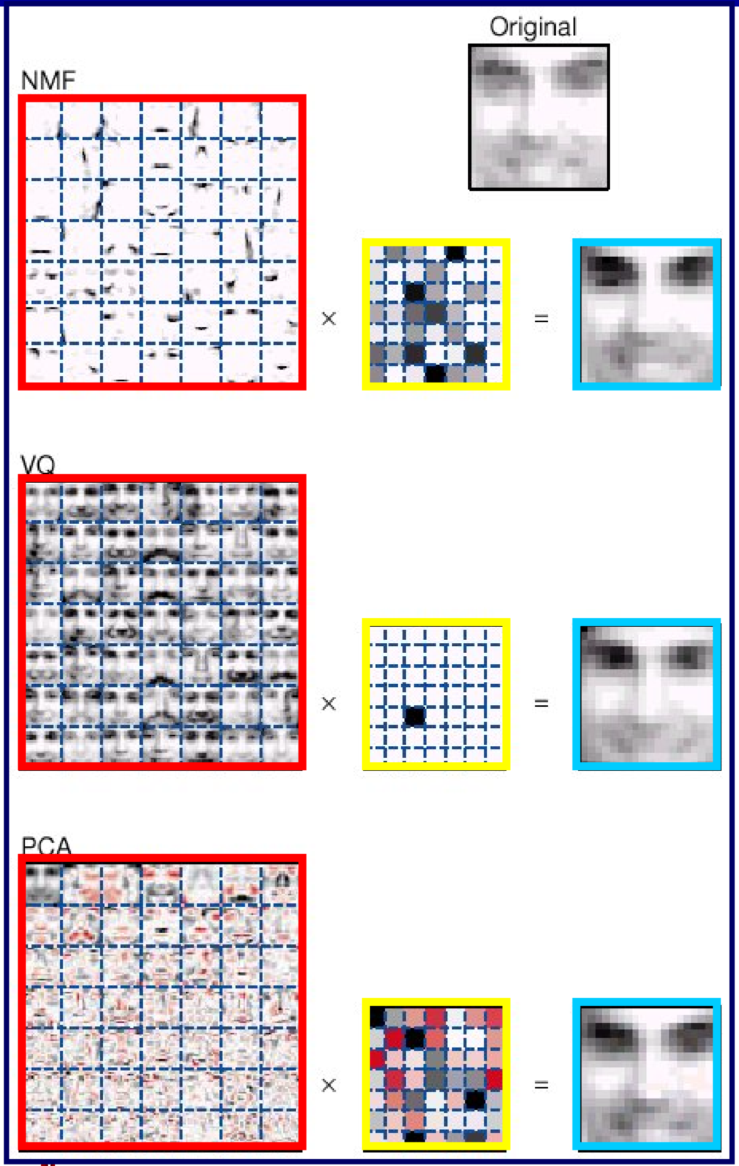


Basis images

Reconstruction



Face features



Basis images

Encoding (Coefficients)

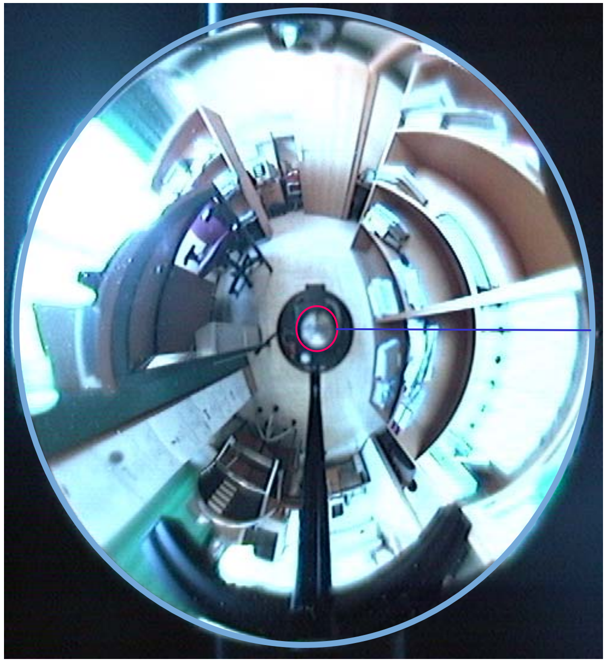
Reconstructed image

Outline Part 2

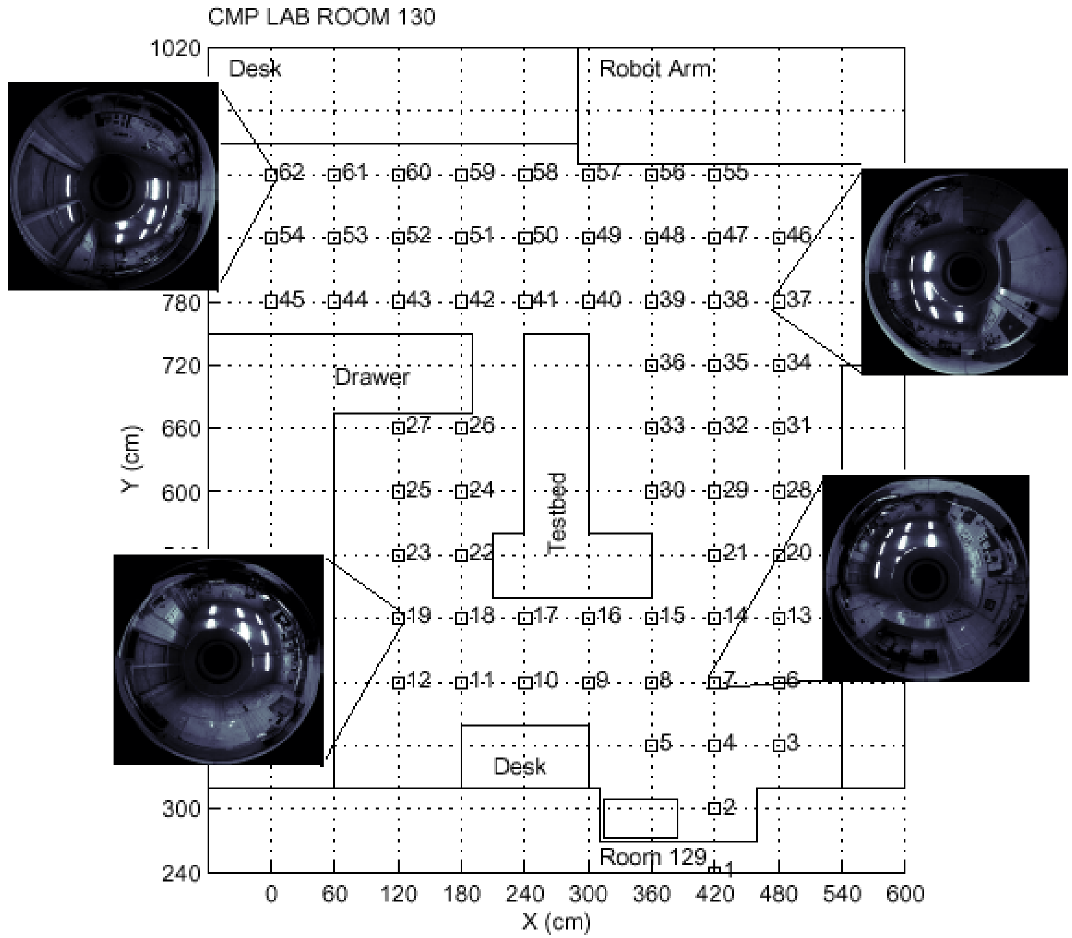
- Robot localization
- Robust representations and recognition
- Robust recognition using PCA
- Scale invariant recognition using PCA
- Illumination insensitive recognition
- Representations for panoramic images
- Incremental building of eigenspaces
- Multiple eigenspaces for efficient representation
- Robust building of eigenspaces
- Research issues

Mobile Robot

Panoramic image

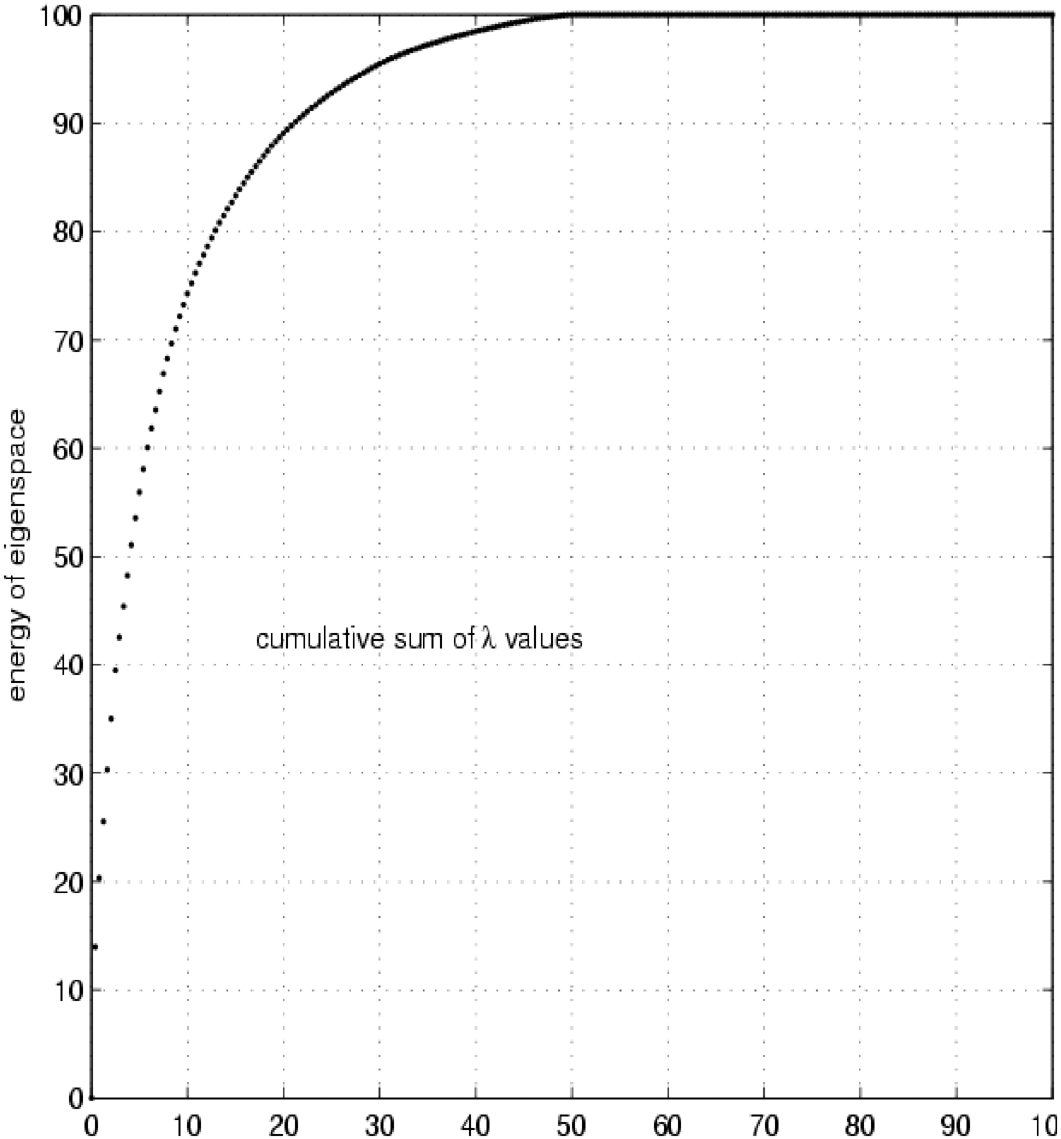


Environment map

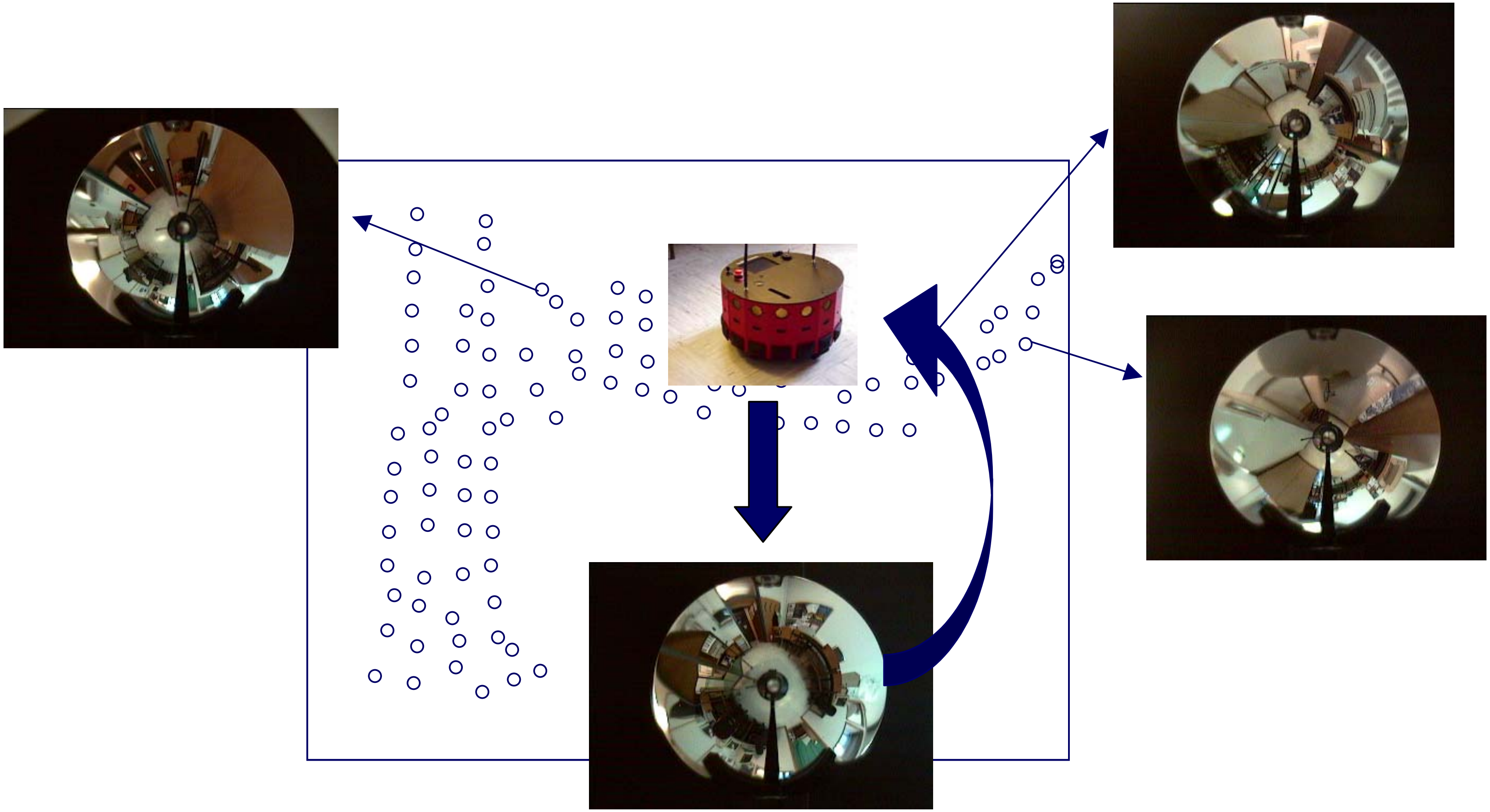


- environments are represented by a large number of views
- localisation = recognition

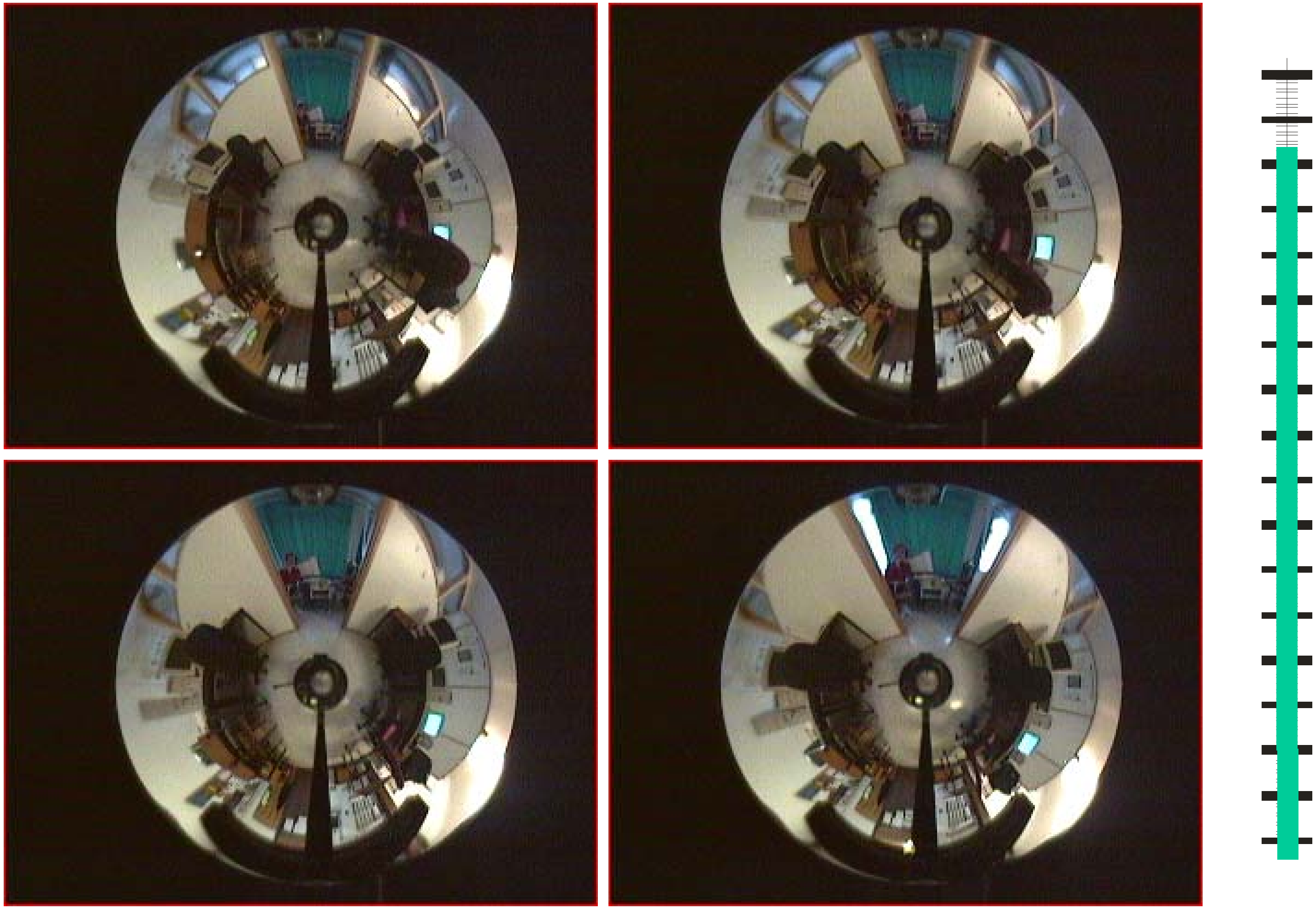
Compression with PCA



Localisation



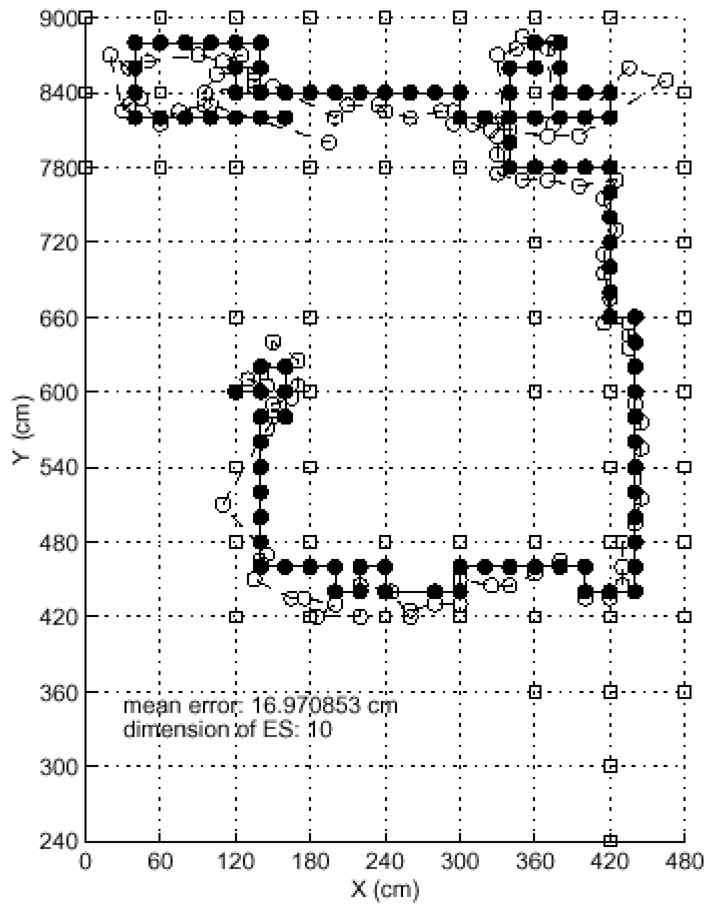
Distance vs. similarity



Robot localisation

- Interpolated hyper-surface represents the memorized environment.
- The parameters to be retrieved are related to position and orientation.
- Parameters of an input image are obtained by scalar product.

Localisation

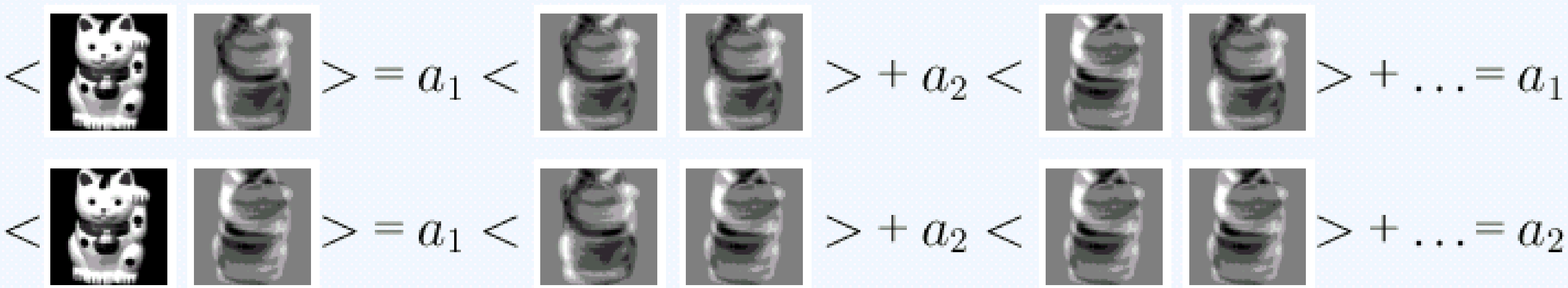


Enhancing recognition and representations

- **Occlusions, varying background, outliers**
  - Robust recognition using PCA
- **Scale variance**
  - Multiresolution coefficient estimation
  - Scale invariant recognition using PCA
- **Illumination variations**
  - Illumination insensitive recognition
- **Rotated panoramic images**
  - Spinning eigenimages
- **Incremental building of eigenspaces**
- **Multiple eigenspaces for efficient representations**
- **Robust building of eigenspaces**

To recover a_i the image is projected onto the eigenspace

$$a_i(\mathbf{x}) = <\mathbf{x}, \mathbf{e}_i> = \sum_{j=1}^{m} x_j e_{i_j} \qquad 1 \le i \le p$$

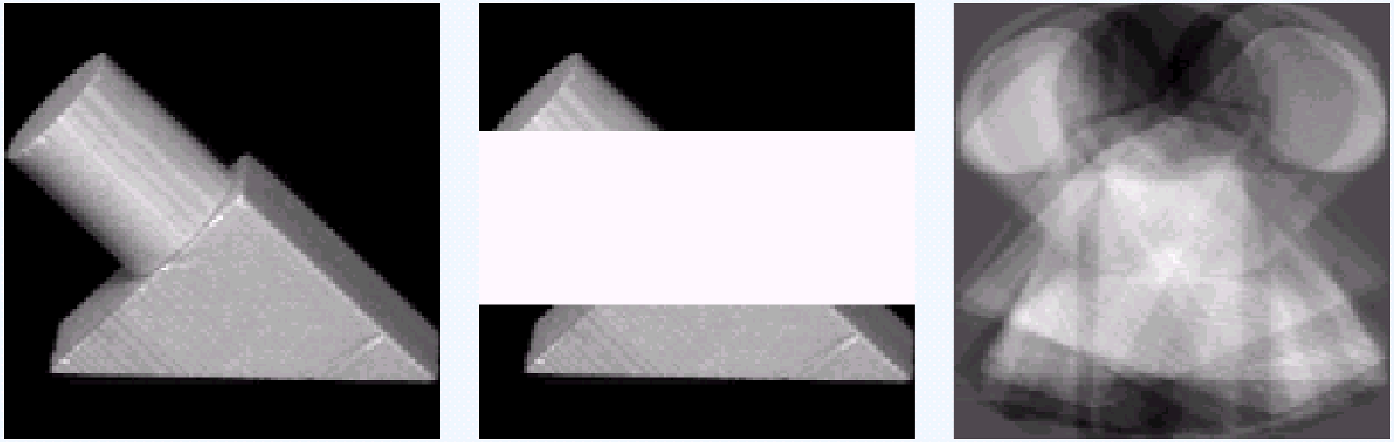


Complete image x_i is required to calculate a_i.
Corresponds to Least-Squares Solution

Non-robustness

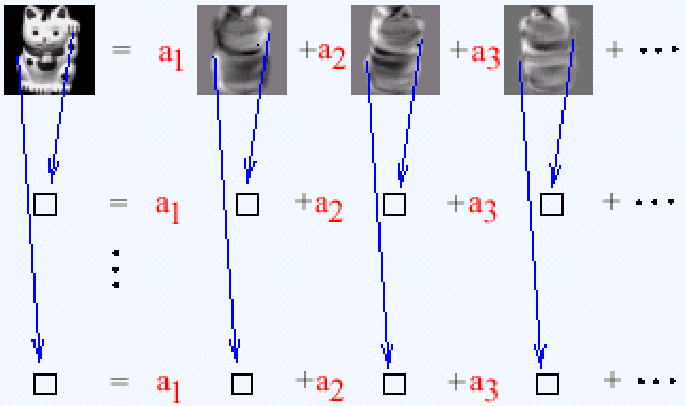
Drawbacks: Prone to errors caused by

- occlusions (outliers)
- cluttered background



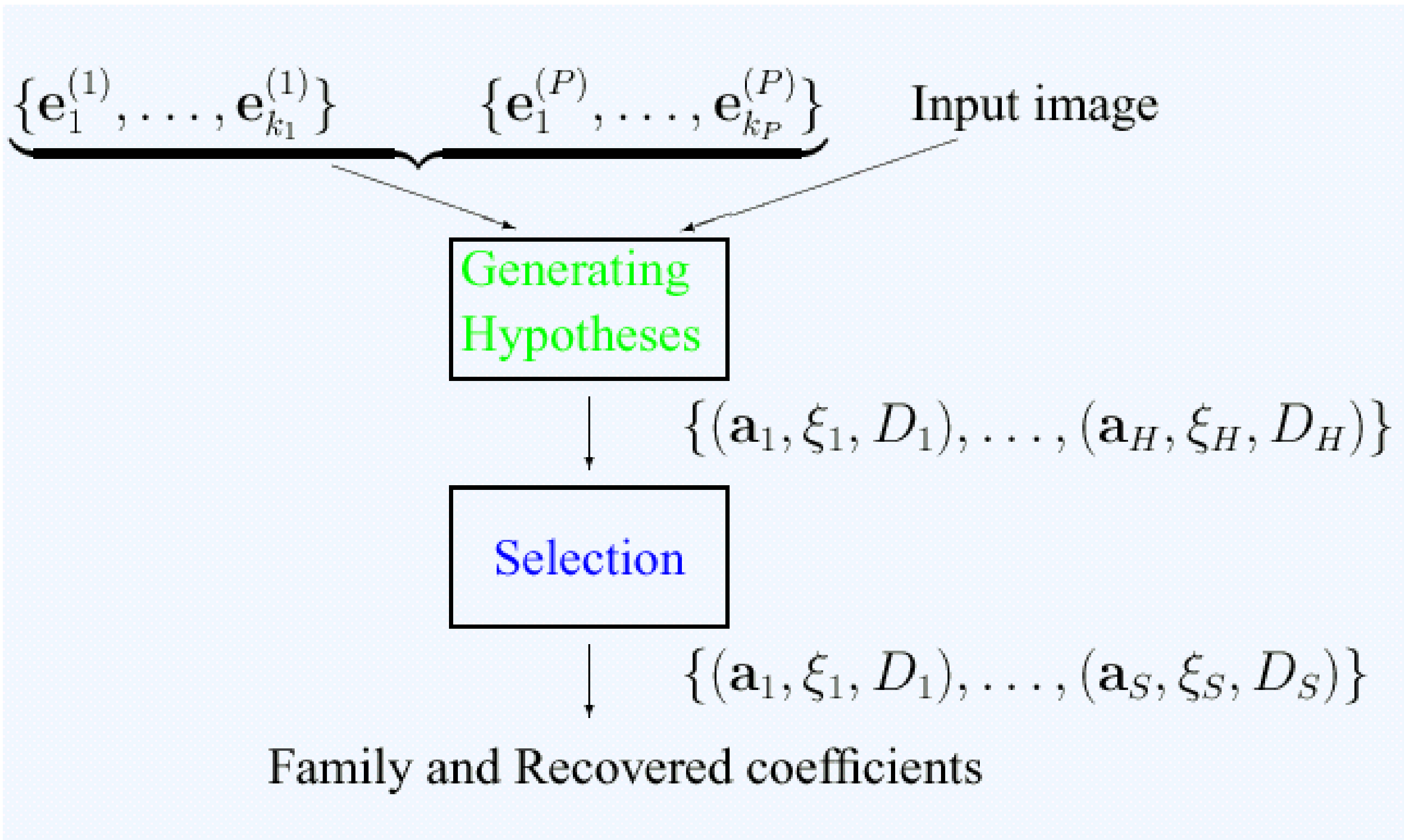
Original Occluded Reconstruction

- **Major idea:** Instead of using the standard approach we:
  - **subset of data points** $\longrightarrow$ linear system of equations
  - **Robust** solution of this system of equations
  - Perform **multiple hypotheses**



- Hypothesize-and-test paradigm
- Competing hypotheses are subject to a **selection** procedure based on the MDL principle.

Robust algorithm



Selection

Three cases:

1. **One object**: Select best match (c_{ii})
2. Multiple **non-overlapping** objects: Select local maximum (c_{ii})
3. Multiple **overlapping** objects: MDL-criterion:

The objective function:

$$F(\mathbf{h}) = \mathbf{h}^T \mathbf{C} \mathbf{h}$$

$\mathbf{h}^T = [h_1, h_2, \ldots, h_R]$ — set of hypotheses

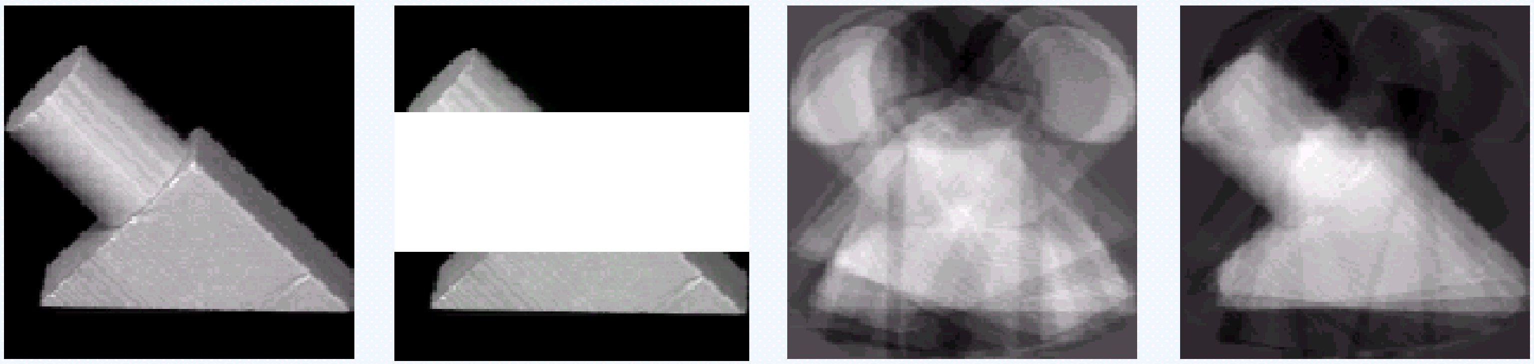
Diagonal terms of $\mathbf{C}$ express the cost-benefit value for hypothesis i

$$c_{ii} = \mathrm{K}_1 |D_i| - \mathrm{K}_2 ||\vec{\xi}_i||_{D_i} - \mathrm{K}_3 N_i$$

Off-diagonal terms handle overlapping hypotheses

$$c_{ij} = \frac{-\mathrm{K}_1 |D_i \cap D_j| + \mathrm{K}_2 \xi_{ij}}{2}$$

Robust recovery of coefficients



Original Occluded Standard Robust

Robustness – Experimental results

Experimental testing on a standard database COIL of 1440 images (20 objects under 72 orientations).

Recognition and pose estimation

Pose estimation :

Method	Salt & Pepper [%]				Gaussian Noise [σ]				Occlusions [%]			
	0	25	50	75	75	150	225	300	15	30	45	60
Standard	2	3	3	48	3	3	4	24	3	25	31	45
Robust	2	3	3	4	4	5	6	10	3	3	16	29

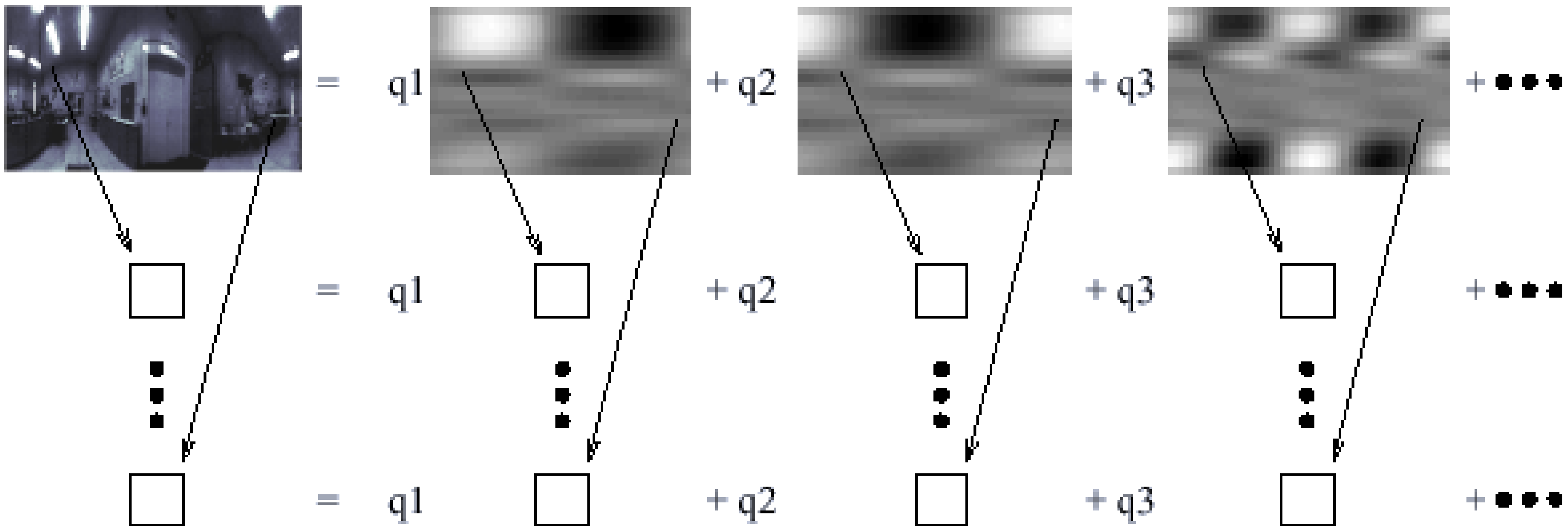
Recognition (50 % salt & pepper noise):

Method	Recognition Rate	Mean absolute orientation error
Standard	46 %	22°
Robust	75 %	6°

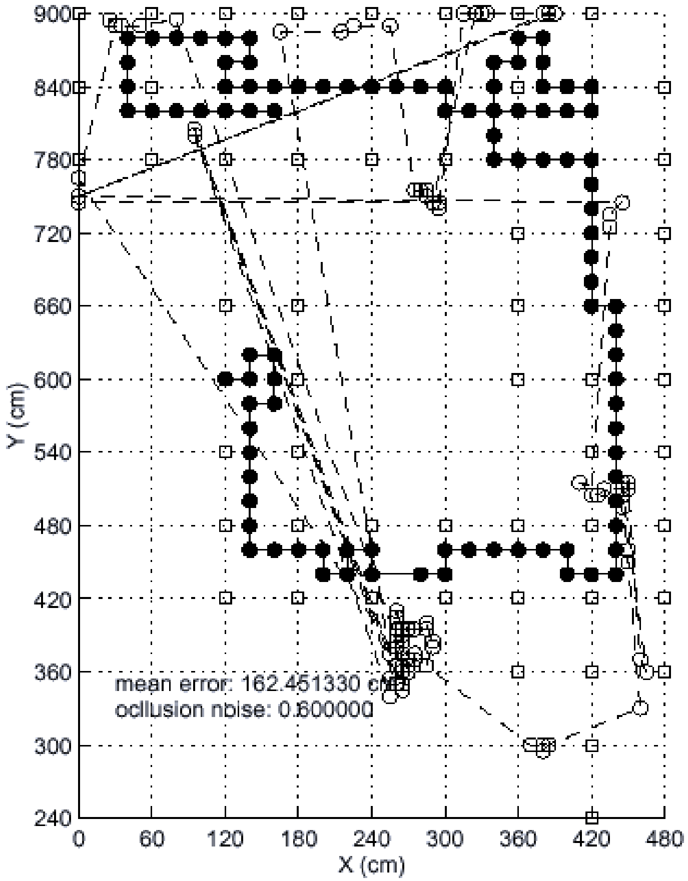
Recognition (50 % occlusion):

Method	Recognition Rate	Mean absolute orientation error
Standard	12 %	57°
Robust	66 %	29°

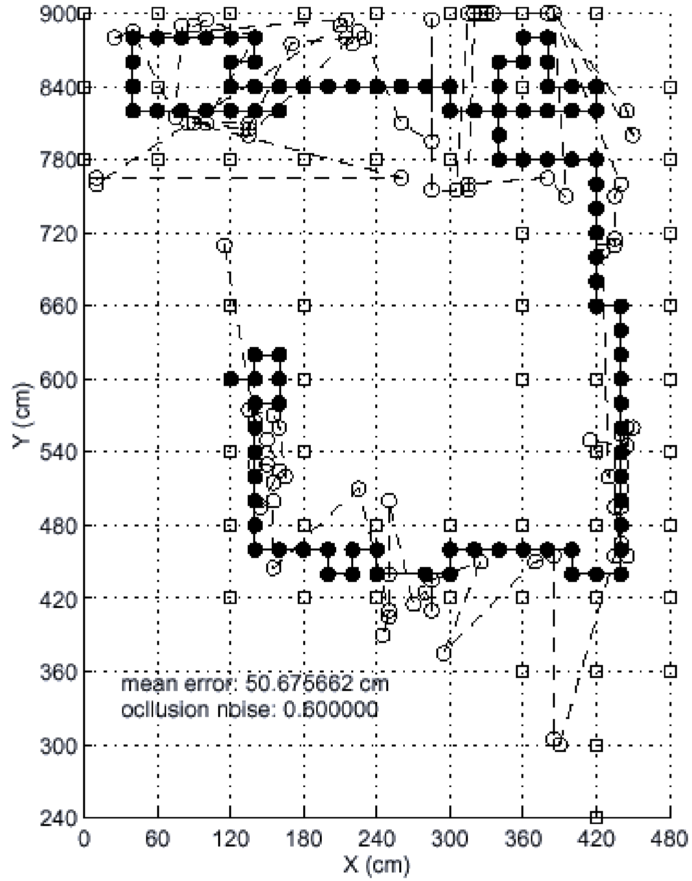
Robust localisation under occlusions



Robust localisation at 60% occlusion



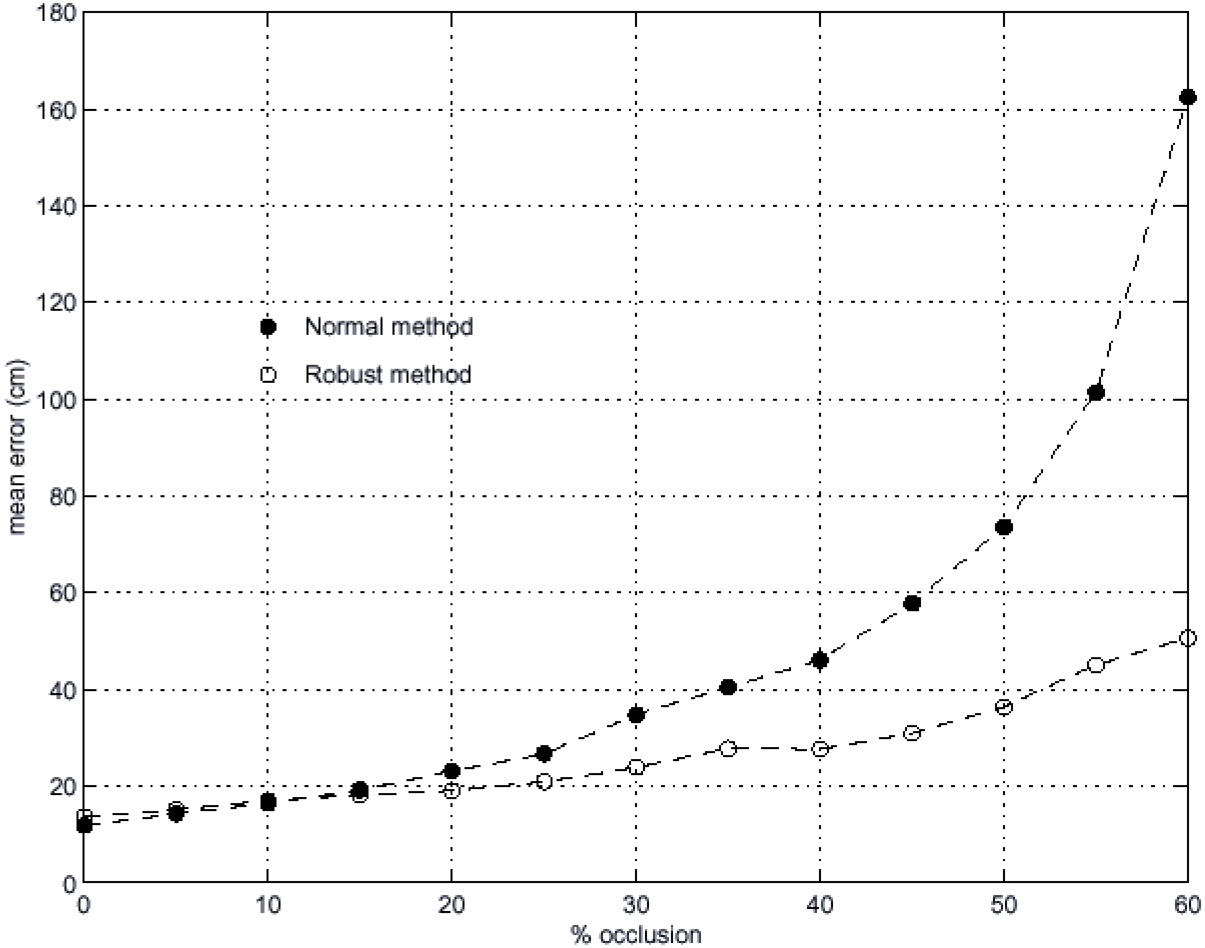
Standard approach



Robust approach

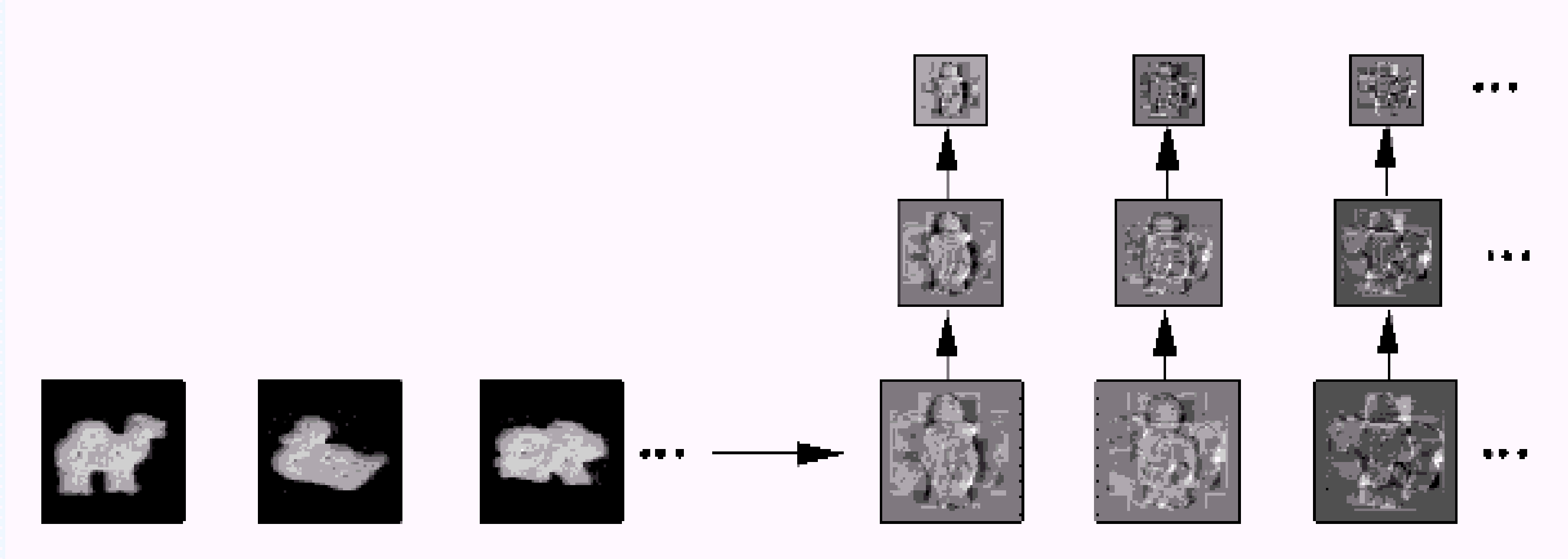
Mean error of localisation

- **Mean error of localisation with respect to % of occlusion**



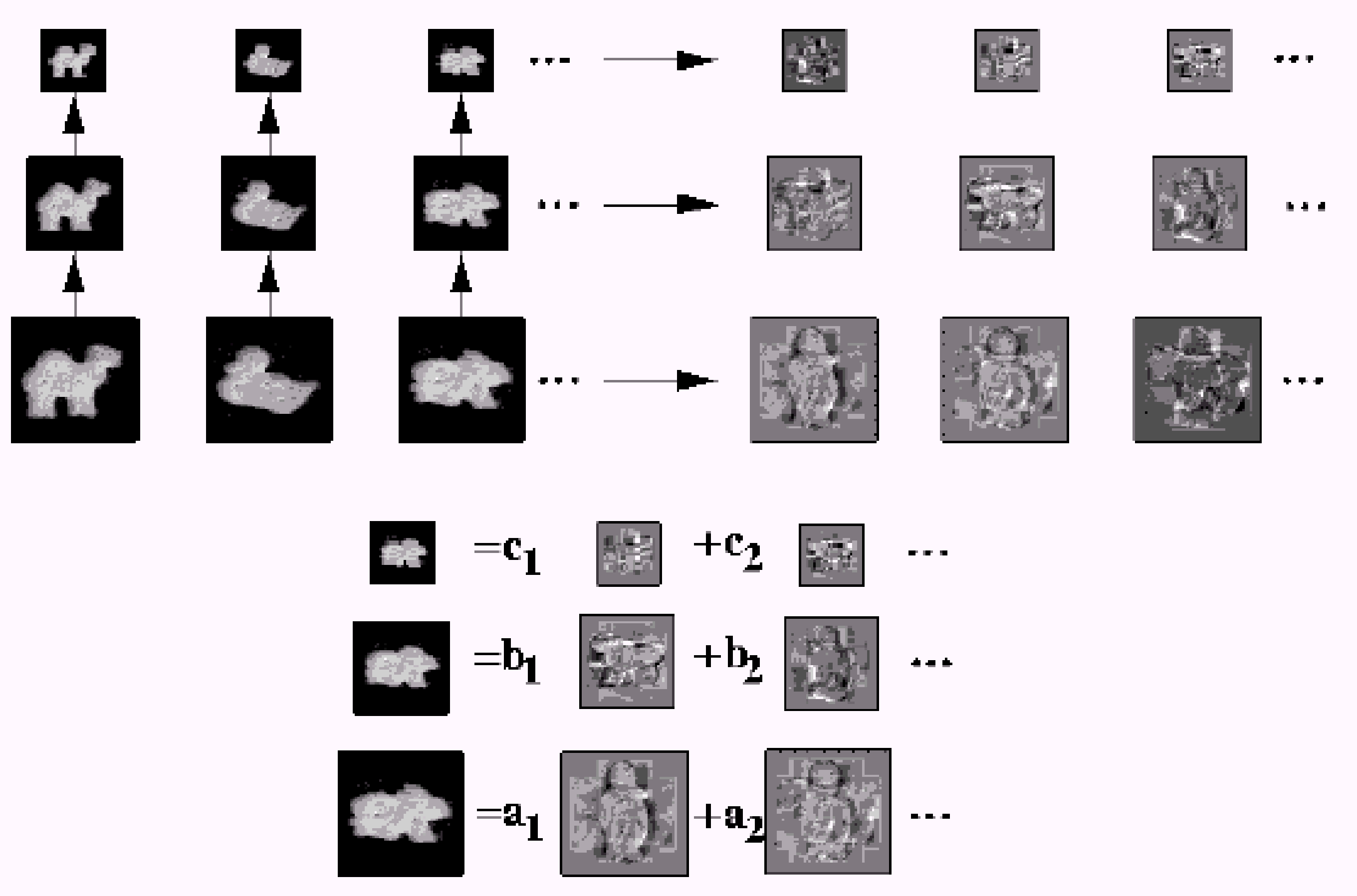
Multiresolution coefficient estimation

- **Multiresolution**
  - a well-known technique to reduce computational complexity
  - a search for the solution at the coarsest level and then a refinement through finer scales
- **Standard eigenspace method cannot be applied in an ordinary multiresolution way — it relies on the orthogonality of eigenimages.**



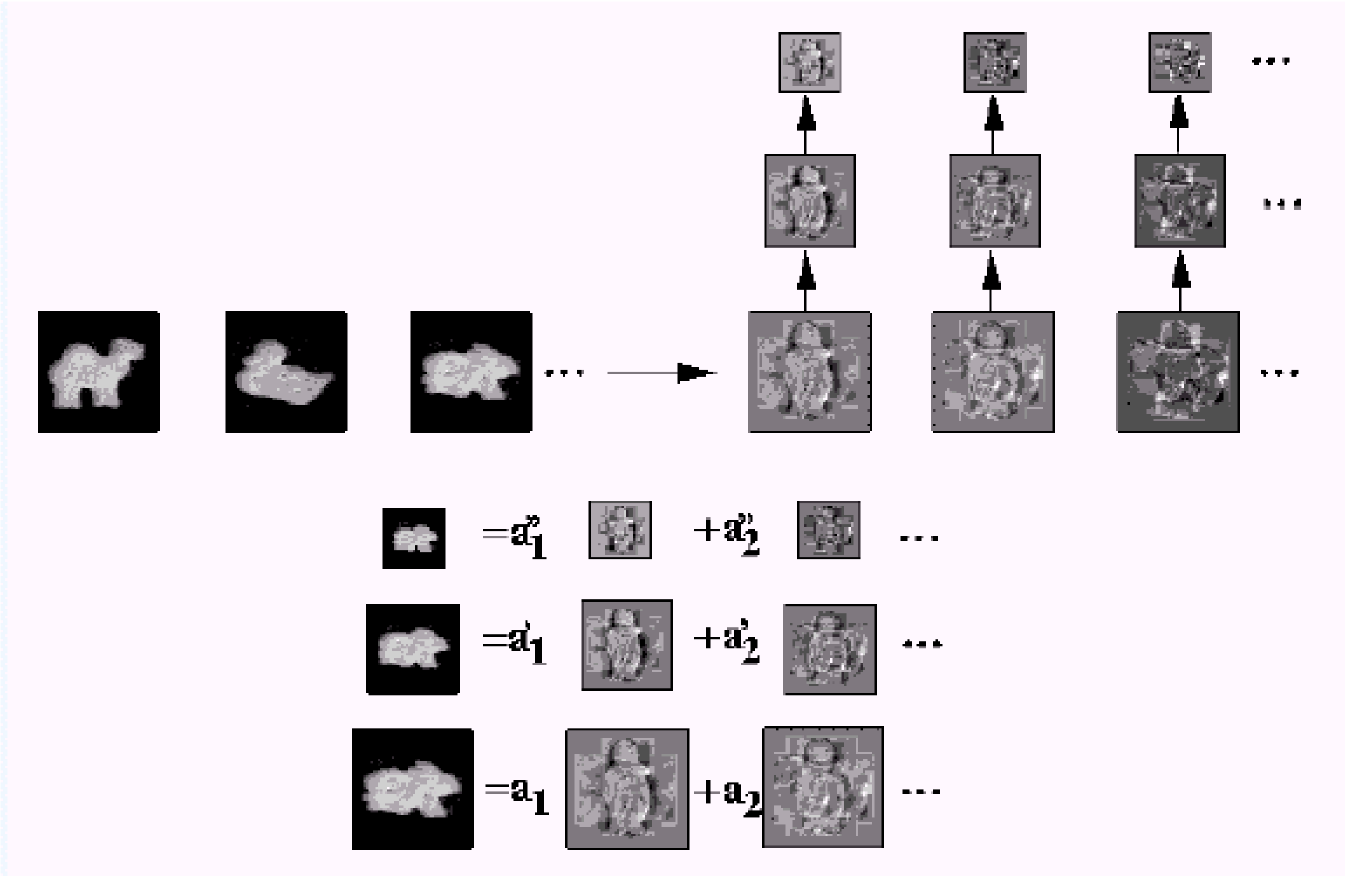
Standard multiresolution coefficient estimation

- Eigenimages in each resolution layer are computed from a set of templates in that layer
- Computationally costly and requires additional storage space



Robust multiresolution coefficient estimation

- Robust method requires only a single set of eigenimages obtained on the finest resolution.
- Linear system of equations: does not require orthogonality.



Multiresolution coefficient estimation

Linear System of Equations:

$$\tilde{x}(\mathbf{r}_j) = \sum_{i=1}^{p} a_i e_i(\mathbf{r}_j) \quad ,$$

Convolution:

$$(f * \tilde{x})(\mathbf{r}_j) = \sum_{i=1}^{p} a_i (f * e_i)(\mathbf{r}_j) \quad ,$$

Sub-sampling:

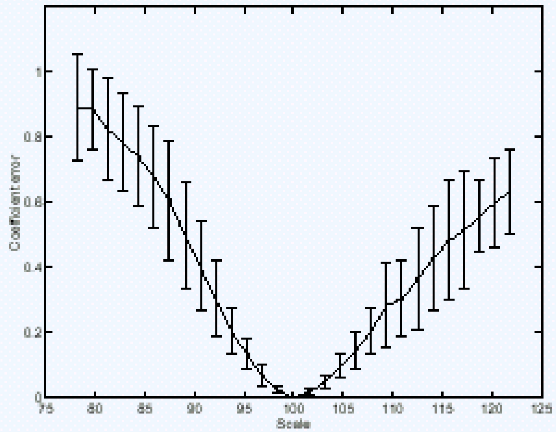
$$\tilde{x}_{\downarrow}(\mathbf{r}_j) = \sum_{i=1}^{p} a_i e_{i\downarrow}(\mathbf{r}_j) \quad ,$$

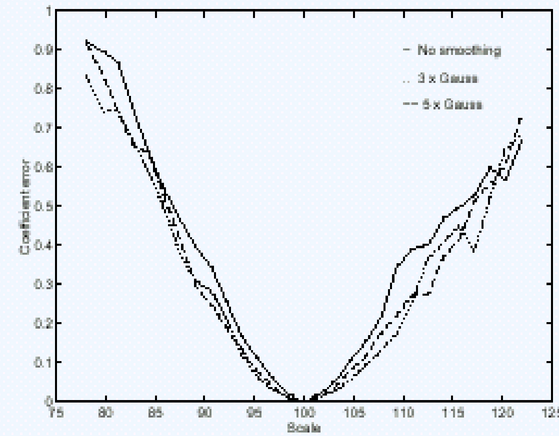
Convolution & Sub-sampling:

$$(f * \tilde{x})_{\downarrow}(\mathbf{r}_j) = \sum_{i=1}^{p} a_i (f * e_i)_{\downarrow}(\mathbf{r}_j) \quad ,$$

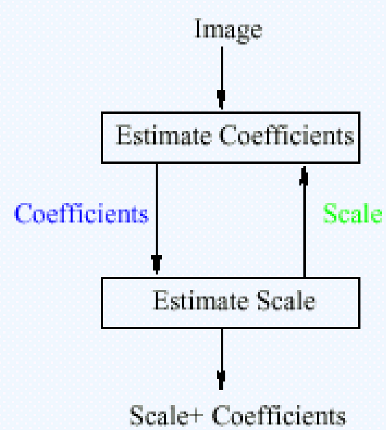
Same coefficients on convolved and sub-sampled eigenimages

Scale Sensitivity:





1. Generate multiple hypotheses at different scales.
2. **Estimate scale & coefficients simultaneously**.



Scale estimation

Minimize:

$$E_s(\alpha) = (\mathbf{s}(\mathbf{x}, \alpha) - \sum_{i=1}^{p} a_i \mathbf{e}_i)^2$$

$\mathbf{s}(\mathbf{x}, \alpha)$: image scaled by α.

- **Gradient descent**[Black]:
  Taylor series expansion of $\mathbf{s}(\mathbf{x}, \alpha)$
  - Small scale changes

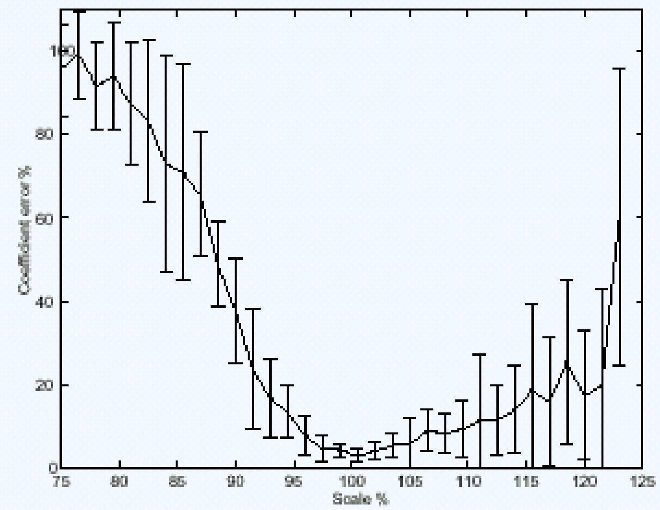
  $\longrightarrow$ High resolution

- **Coarse exhaustive search**:
  - Computationally costly

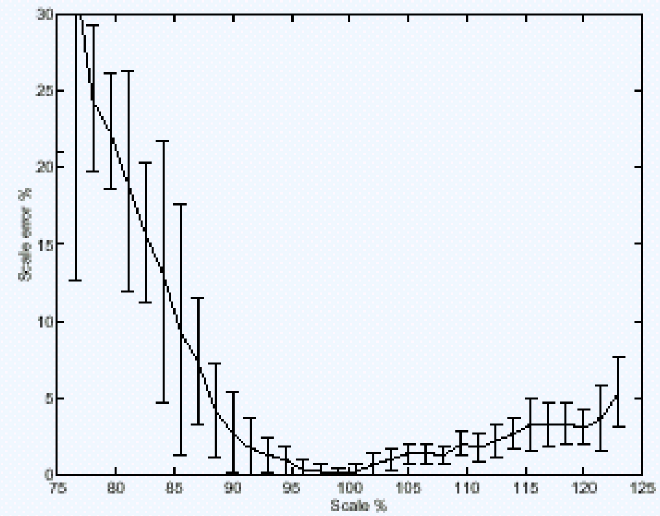
  $\longrightarrow$ Low resolution

Numerical demonstration

Coefficient Error:

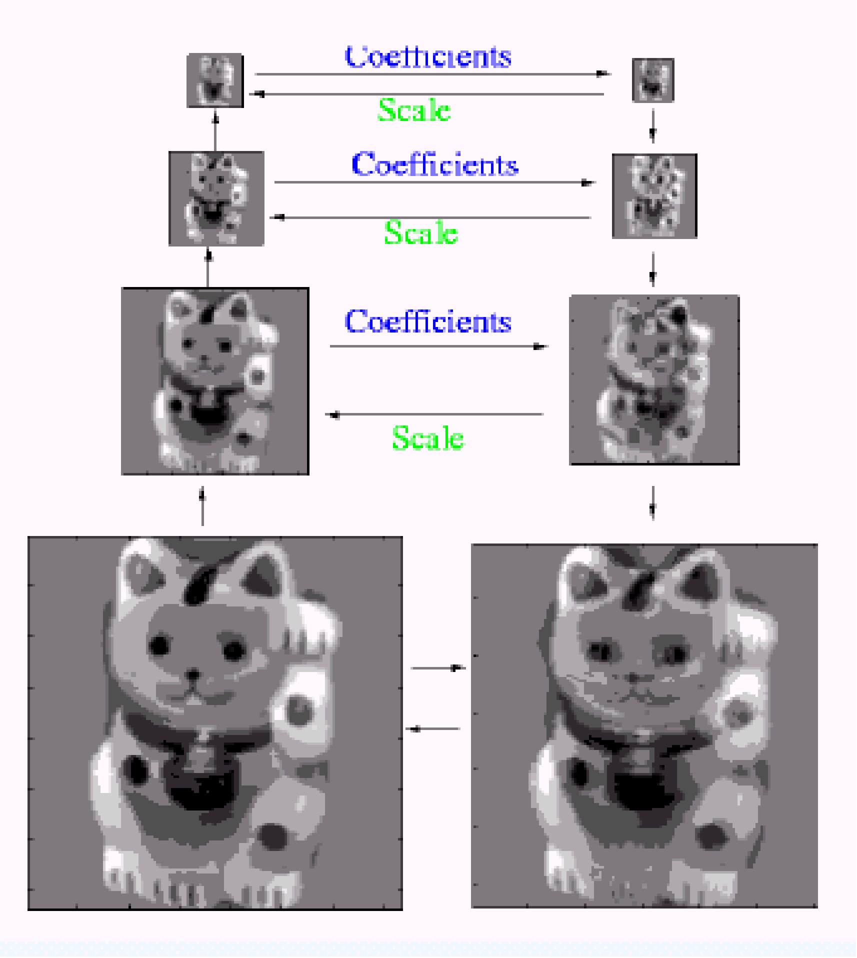


Scale Error:

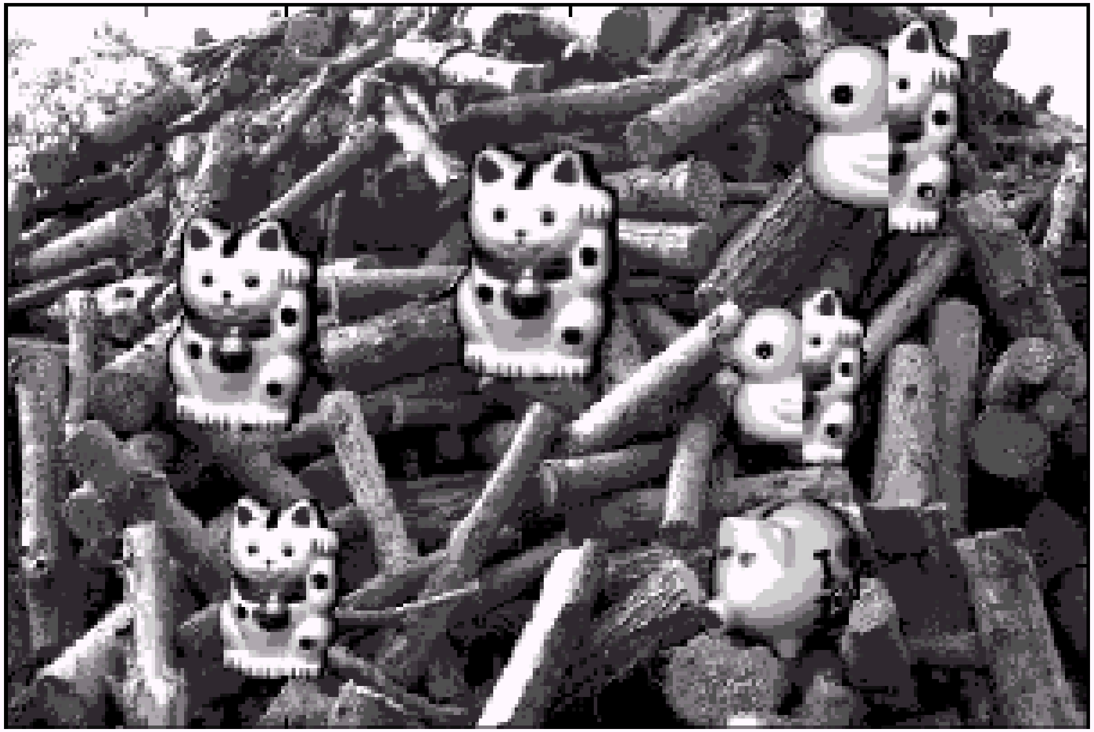


Multiresolution approach

- Estimate scale & coefficients simultaneously in the pyramid
- Efficient search structure



Experimental results – test image

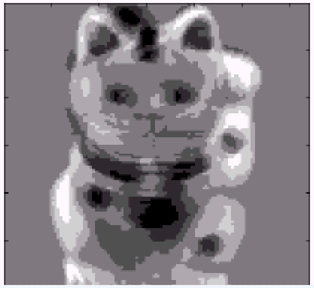


Experimental results

Cat

120% Scaled cat

Occluded cat

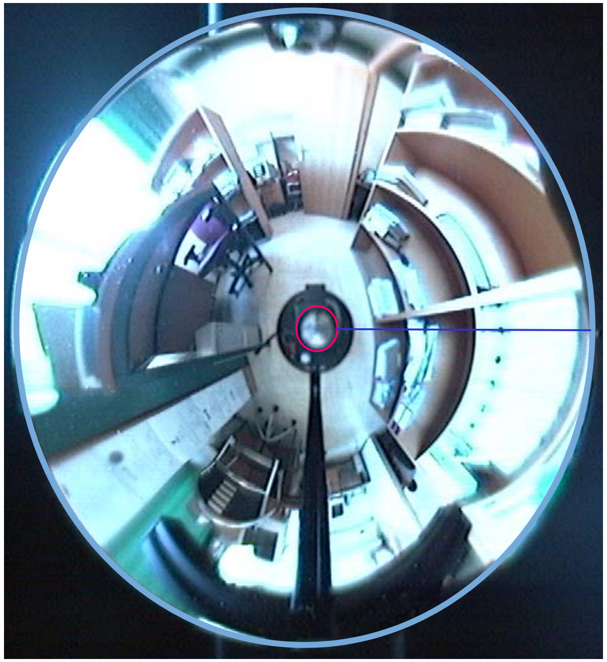


120% Scaled occluded cat

Occluded duck

120% Scaled occluded duck

Unwrapping



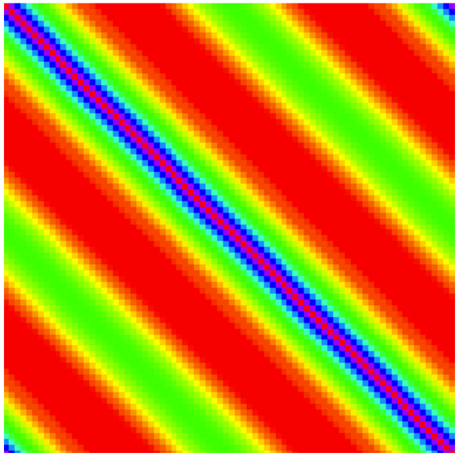
A rotated panoramic image

- **rotated/shifted *n* times**

$$X^{mn} = \begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \dots & \mathbf{x}_{n-1} \end{bmatrix}$$

- **Inner product matrix $Q = X^T X$**
- **symmetric, Toeplitz, circulant**

$$Q = \begin{bmatrix} q_0 & q_1 & \dots & q_{n-2} & q_{n-1} \\ q_{n-1} & q_0 & q_1 & \dots & q_{n-2} \\ q_{n-2} & q_{n-1} & q_0 & q_1 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ q_1 & \dots & q_{n-2} & q_{n-1} & q_0 \end{bmatrix}$$



Eigenvectors of a circulant matrix

- **Shift theorem: the eigenvectors of a general circulant matrix are the *N* basis vectors from the Fourier matrix *F*=[u_0', u_1', ... , u_{n-1}'], where**

$$\mathbf{u}_i' = \left[1, \omega^i, \omega^{2i}, \ldots, \omega^{(n-1)i}\right]^\top, \quad i = 0, \ldots, n-1$$

- **The eigenvalues can be calculated simply by retrieving the magnitude of the DFT of one row of *Q***

$$\omega = e^{-2\pi j/n}, \quad j = \sqrt{-1}$$

$$\lambda_i = \sum_{l=0}^{n-1} q_l \, \omega^{i\,l}$$

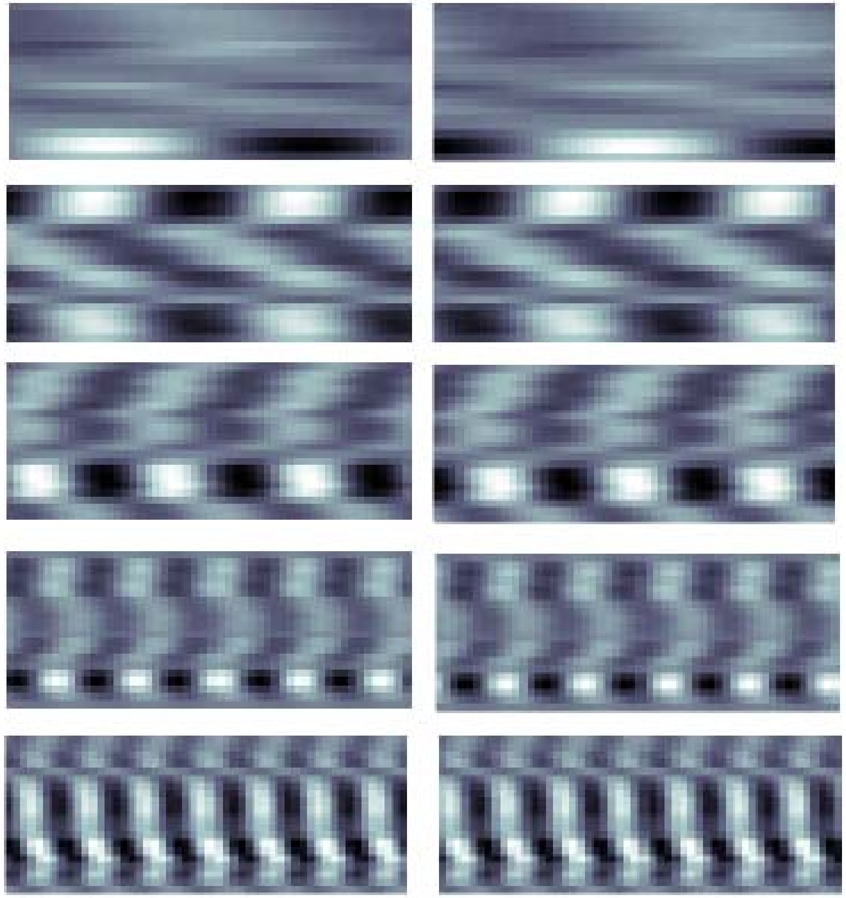
From u_i' to u_i

- **The eigenvectors of XX^T can be obtained by using $XX^T X u_i' = \lambda_i' X u_i'$:**

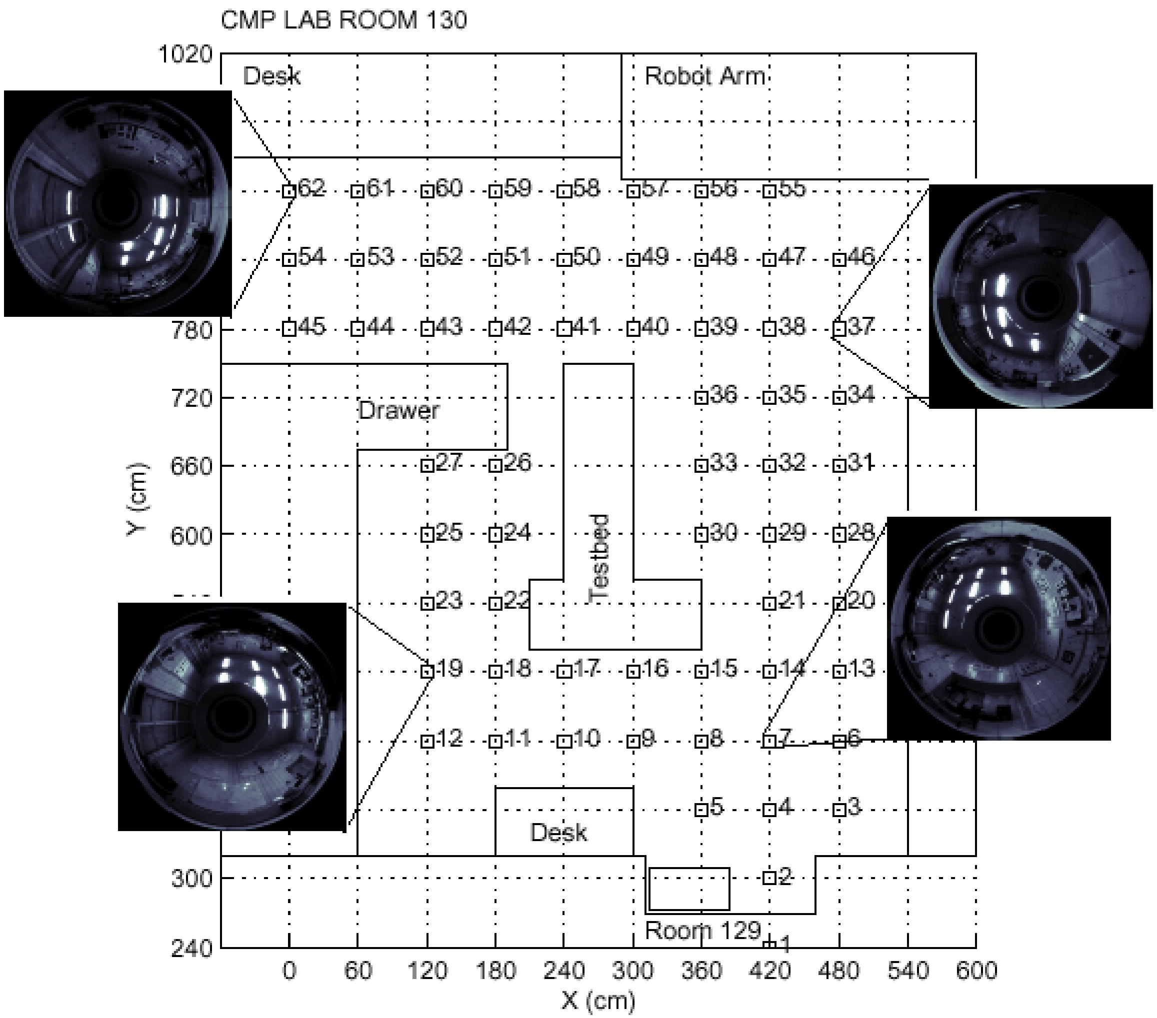
$$\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i'}} X \mathbf{u}_i'$$

- **eigenvectors** u_i
  - same frequency as $\mathbf{u}_i'$,
  - phase and amplitude may change

Eigenvectors



Generalisation to several locations



A set of rotated images

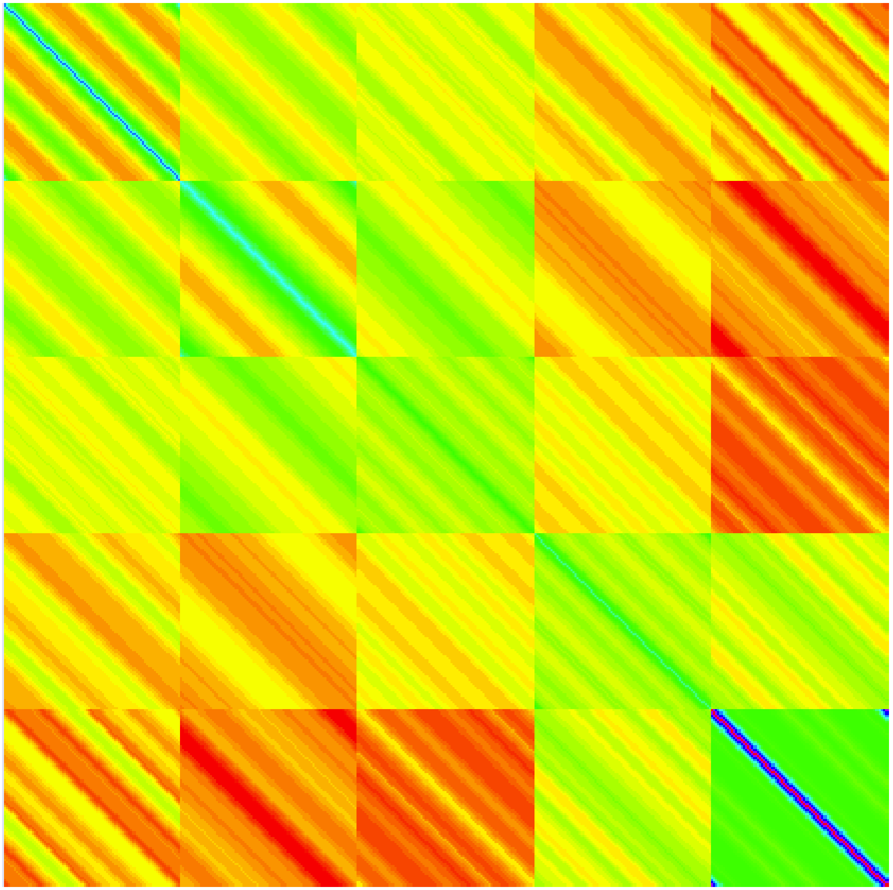
- *P* different locations, each shifted *n* times

$$A = X^T X = \begin{bmatrix} Q_{00} & Q_{01} & \dots & Q_{0,P-1} \\ Q_{10} & Q_{11} & \dots & Q_{1,P-1} \\ \dots & \dots & \dots & \dots \\ Q_{P-1,0} & Q_{P-1,1} & \dots & Q_{P-1,P-1} \end{bmatrix}$$

- Every Q_{ij} is circulant (but in general not symmetric!)
- Is it possible to exploit these properties?

It is still possible to compute the eigenvectors without performing the SVD decomposition of A.

Rotated panoramic images



Other approaches

- Images stored in arbitrary orientation
- Images stored in a reference orientation (e.g. gyrocompass)
- Autocorrelation
- FFT power spectra
- Zero Phase Representation
- Eigenspace of spinning-images