This is an extended and modified version of the tutorial given at the ECCV'2002, Copenhagen, and ICPR'2002, Quebec City.


# Subspace Methods for Visual Learning and Recognition

Aleš Leonardis

Faculty of Computer and Information Science
University of Ljubljana
Slovenia

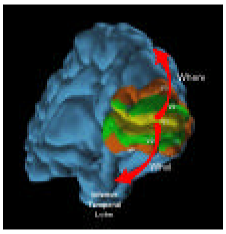
Ales.Leonardis@fri.uni-lj.si

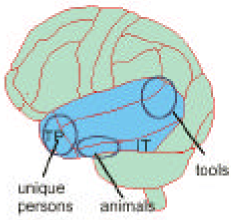
## Vision

- Vision - the most important and informative human sense
- 70% of the total information is obtained through vision
- Recognition is an essential part of human perception
- Recognition implies learning (re-cognition)
- Learning-representation-recognition (three inseparable parts of visual perception)
- Visual recognition seems to be an easy task for humans.
  - How does human brain learn and store visual information?
  - How is the recognition performed?
- Psychology, psychophysics, neuroscience
- Computer vision



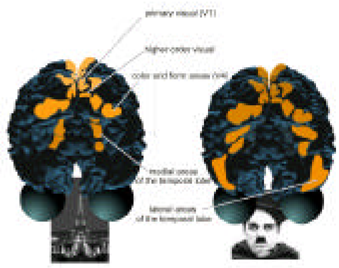
## Human perception



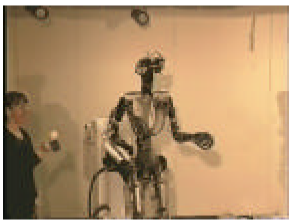



## Human perception



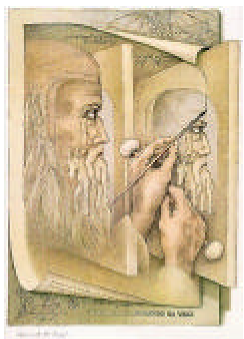



## Humanoid robot





## Complexity of Recognition



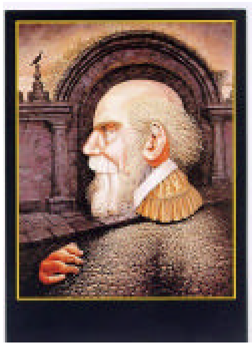
## Complexity of Recognition



## Complexity of Recognition



## Complexity of Recognition

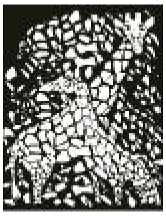




## Complexity of Recognition

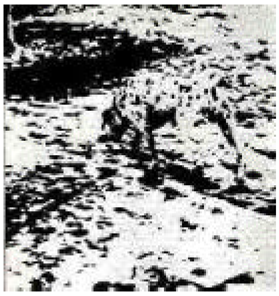


## A mosaic?

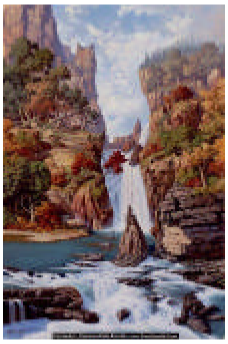




## Complexity of recognition

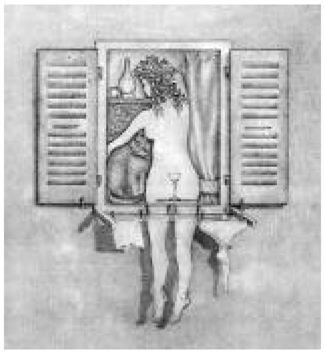


## Complexity of Recognition



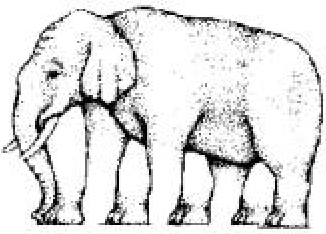


## Complexity of Recognition





## Complexity of recognition





## A duck or a rabbit?



## One or two faces?

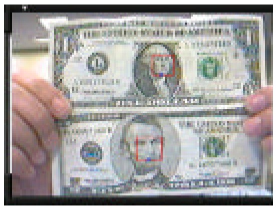


## Clinton and Gore?

## Complexity of Recognition



## Face recognition





## Mobile Robot



## Outline

- **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
- **Principal Components Analysis (PCA)**
  - Object recognition
  - Robot localization

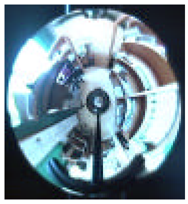


## Outline

- **Principal Components Analysis (PCA) – Extensions**
- **Robust recognition**
  - Robust PCA recognition
  - Scale invariant recognition using PCA
  - Illumination insensitive recognition
- **Representations**
  - Representations for panoramic images
  - Incremental building of eigenspaces
  - Multiple eigenspaces for efficient representation
  - Robust building of eigenspaces
- **Other subspace representations (LDA, CCA, ICA, NMF, Kernel)**
- **Research issues**



## The name of the game



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

## Recognition

- What objects are we looking at?
  - Model search needed, image region search needed
- Is this part of the image an instance of X?
  - Given model, given image region
- What is this part of the image?
  - Model search needed, given image region
- Are there any instances of X in the image?
  - Given model, image region search needed



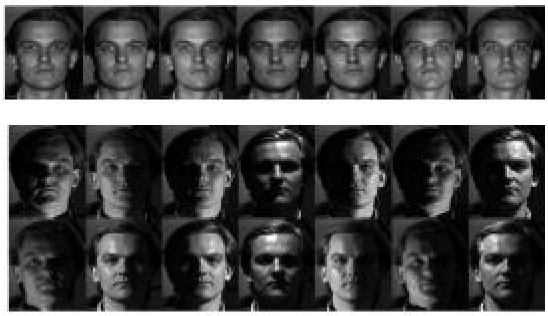
## Problems

Segmentation:

Pose/Shape:



## Illumination

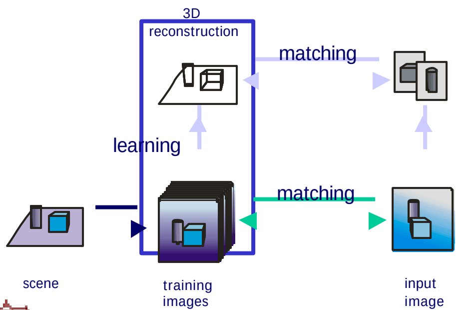




## Example



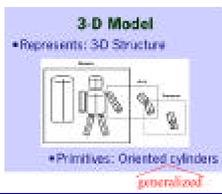
## Learning and recognition






## Recognition

- Interpretation trees
  - Given
    - The list of feature descriptors from a given object model
    - The list of feature descriptors detected in the image
    - A list of (geometric) constraints that model features must satisfy
  - Find a mapping between model features and image features such that the constraints satisfied by the model features are satisfied by the corresponding image features.



## Recognition

- **Interpretation trees**
- **Invariants**
  - Properties of geometric configurations which do not change under a certain class of transformations (projective invariants)
- **Appearance-based recognition**



## Appearance-based approaches

Attention in the appearance-based approaches

Encompass combined effects of:

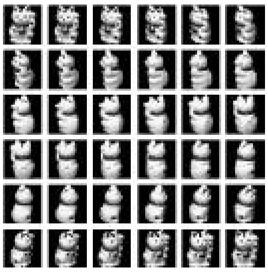
- shape,
- reflectance properties,
- pose in the scene,
- illumination conditions.

**Acquired through an automatic learning phase.**



## Appearance-based approaches

Objects are represented by a large number of views:

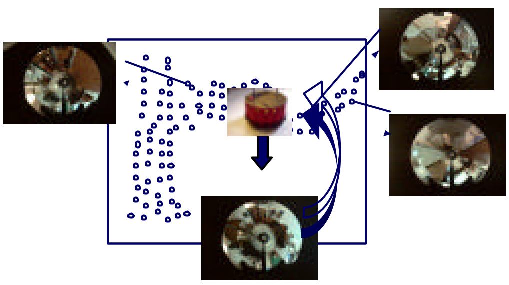




## Appearance-based approaches

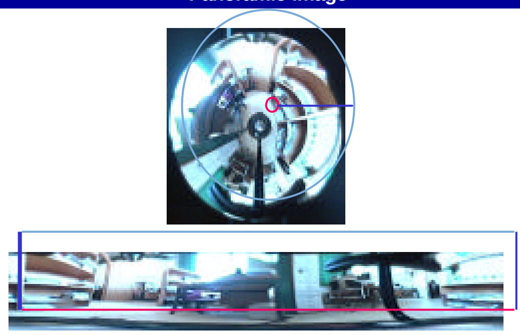


## Localisation



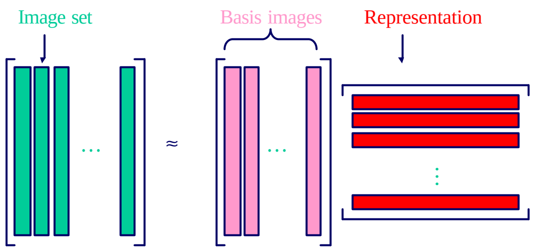


## Panoramic image



## Subspace Methods

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





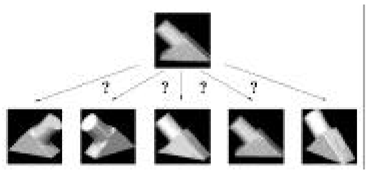

## Subspace Methods

Properties of the representation:

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



## Image Matching



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

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

⇒ Compress images



## Eigenspace representation

- **Image set (normalised, zero-mean)**

$$X = [\ \mathbf{x}_0\ \ \mathbf{x}_1\ \ \ldots\ \ \mathbf{x}_{n-1}\ ];\ X \in \mathbb{R}^{m\times n}$$

- **We are looking for orthonormal basis functions:**

$$U = [\ \mathbf{u}_0\ \ \mathbf{u}_1\ \ \ldots\ \ \mathbf{u}_k\ ];\ k << n$$

- **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 u ?

- **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**

$$C = XX^T = [\ \mathbf{x}_0\ \ \mathbf{x}_1\ \ \ldots\ \ \mathbf{x}_{n-1}\ ] \begin{bmatrix} \mathbf{x}_0^T \\ \mathbf{x}_1^T \\ \ldots \\ \mathbf{x}_{n-1}^T \end{bmatrix}$$

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

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



## Efficient eigenspace computation

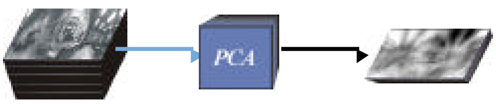
- ***n << m***
- **Compute the eigenvectors $u'_i$, $i = 0,\ldots,n-1$, of the inner product matrix**

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

- **The eigenvectors of $XX^T$ can be obtained by using $XX^TXv'_i = \lambda'_i Xv'_i$:**

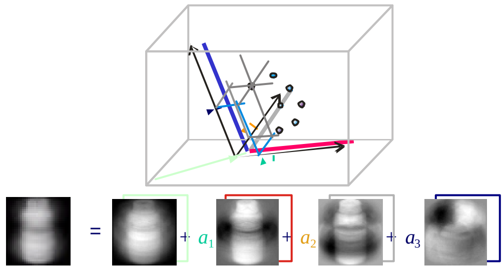
$$\mathbf{u}_i = \frac{1}{\sqrt{\lambda'_i}} X\mathbf{u}'_i$$

## Principal Component Analysis



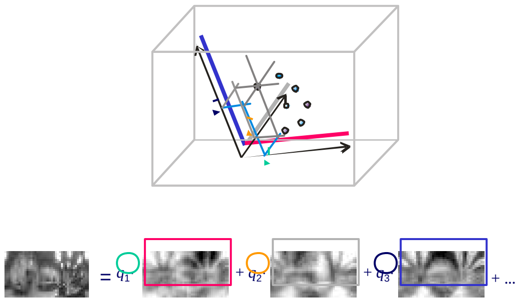



## Principal Component Analysis






## Principal Component Analysis

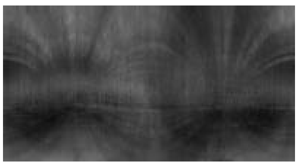





## Image presentation with PCA

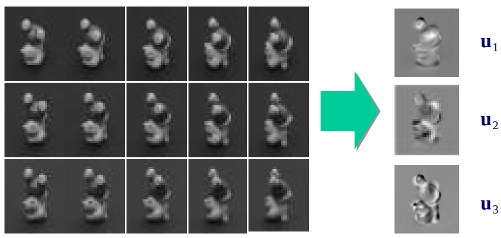


## Image presentation with PCA





## Image representation with PCA



## 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 :

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

- PCA minimizes reconstruction error
- PCA maximizes variance of projection
- Finds a more "natural" coordinate system for the sample data.

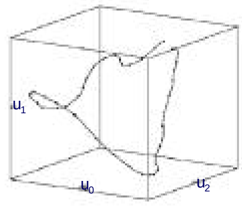


## PCA for visual recognition and pose estimation

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

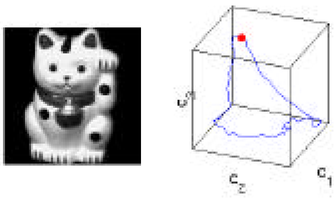
An example:

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






## Parametric eigenspace

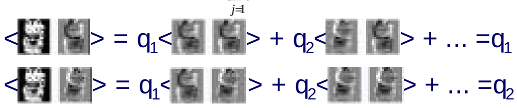





## Calculation of coefficients

To recover $a_i$ the image is projected onto the eigenspace

$$q_i(\mathbf{x}) = \langle \mathbf{x}, \mathbf{u}_i \rangle = \sum_{j=1}^{n-1} x_j u_{ij} \qquad 1 \le i \le k$$

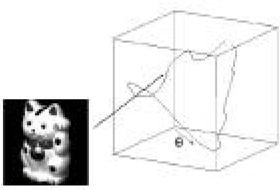



- Complete image **x** is required to calculate $a_i$.
- Corresponds to Least-Squares Solution

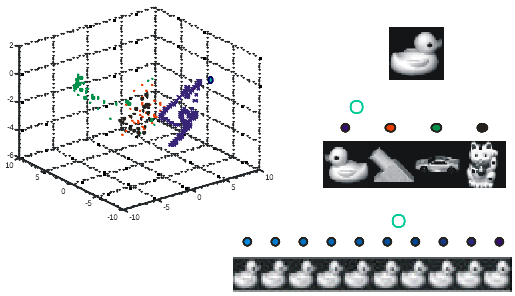


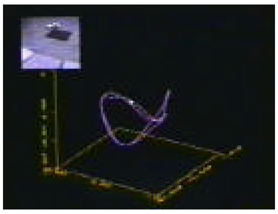
## PCA for visual recognition and pose estimation

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





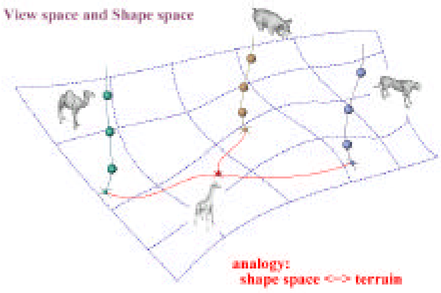








## View space and Shape space






## Principal Component Analysis (PCA)

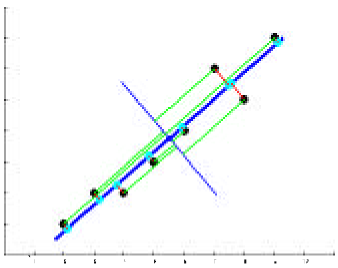
- **PCA is a linear transformation from a high-dimensional input space to a low-dimensional feature space, which**
  - maximizes variance of projected input vectors
  - minimizes reconstruction error
  - decorrelates input vectors.
- **PCA finds in a data-driven way a more "natural" coordinate frame for representing given data.**



## Principle of PCA

Rotate coordinate frame in order to:

- Maximize variance of projections.
- Minimize reconstruction error.



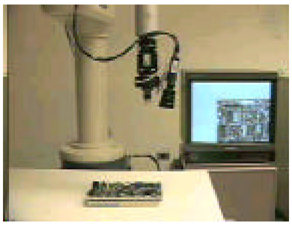


## Appearance-based approaches

A variety of successful applications:

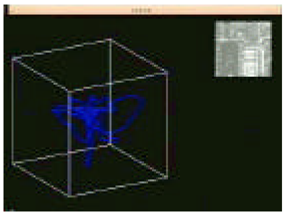
- Human face recognition e.g. [Beymer & Poggio, Turk & Pentland]
- Visual inspection e.g. [Yoshimura & Kanade]
- Visual positioning and tracking of robot manipulators, e.g. [Nayar & Murase]
- Tracking e.g., [Black & Jepson]
- Illumination planning e.g., [Murase & Nayar]
- Image spotting e.g., [Murase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

## Temporal inspection



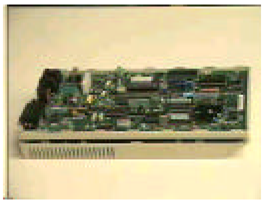


## Temporal inspection



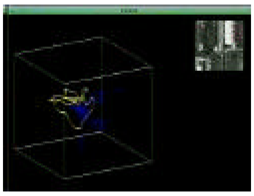


## Temporal inspection





## Temporal inspection



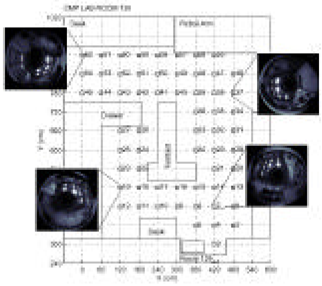


## Mobile Robot



## Panoramic image

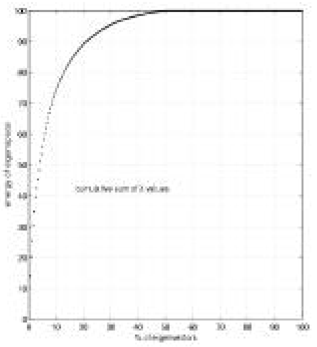
## Environment map



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

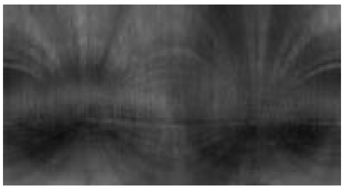


## Compression with PCA



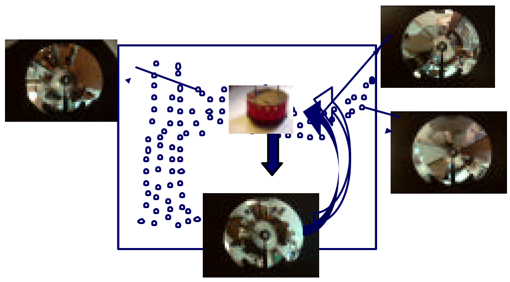


## Image representation with PCA



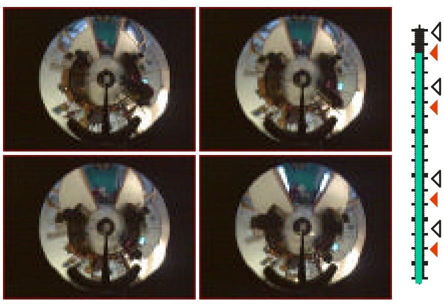


## Localisation





## Distance vs. similarity

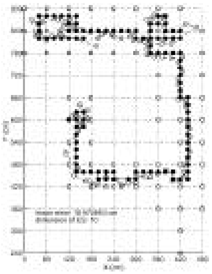




## Robot localisation

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

## Localisation





## Outline

- **Principal Components Analysis (PCA) – Extensions**
- **Robust recognition**
  - Robust PCA recognition
  - Scale invariant recognition using PCA
  - Illumination insensitive recognition
- **Representations**
  - Representations for panoramic images
  - Incremental building of eigenspaces
  - Multiple eigenspaces for efficient representation
  - Robust building of eigenspaces
- **Other subspace representations (LDA, CCA, ICA, NMF, Kernel)**
- **Research issues**

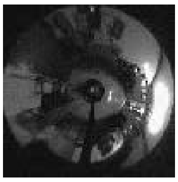


## Enhancing recognition and representations

- **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**



## Occlusions

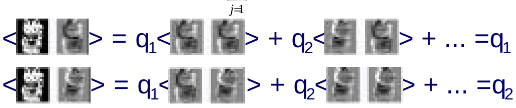




## Calculation of coefficients

To recover $q_i$ the image is projected onto the eigenspace

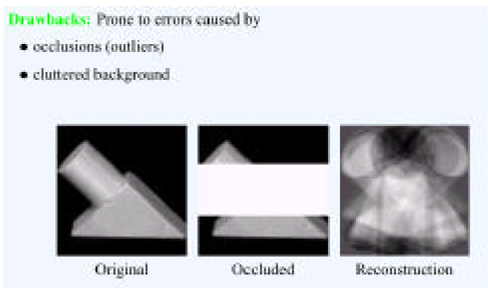
$$q_i(\mathbf{x}) = <\mathbf{x}, \mathbf{u}_i> = \sum_{j=1}^{n-1} x_j u_{ij} \qquad 1 \le i \le k$$




- Complete image $\mathbf{x}$ is required to calculate $q_i$.
- Corresponds to Least-Squares Solution

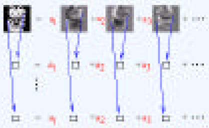


## Non-robustness



## Robust method

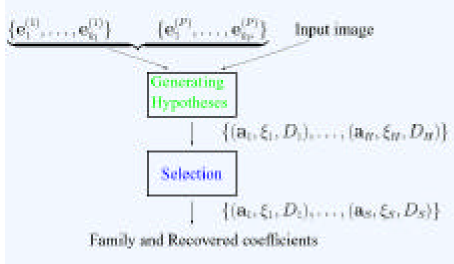
- **Major idea:** Instead of using the standard approach we:
  - **subset of data points** ⟶ linear system of equations
  - **Robust** solution of this system of equations
  - Perform **multiple hypotheses**



- 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 $i$

$$c_{ii} = K_1|D_i| - K_2||\vec{\xi}||_{D_i} - K_3 N_i$$

Off-diagonal terms handle overlapping hypotheses

$$c_{ij} = \frac{-K_1|D_i \cap D_j| + K_2\xi_{ij}}{2}$$



## 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 |

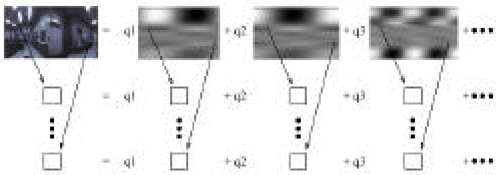
**Recognition (50 % salt & pepper noise):**

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

**Recognition (50 % occlusion):**

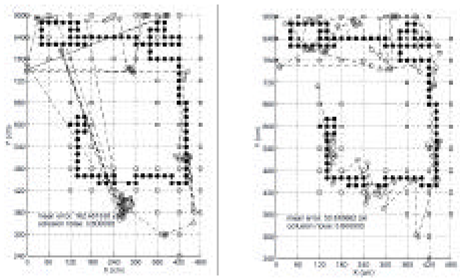
| Method | Recognition Rate | Mean absolute orientation error |
|---|---|---|
| Standard | 12 % | 57° |
| Robust | 66 % | 29° |

## Robust localisation under occlusions





## Robust localisation at 60% occlusion

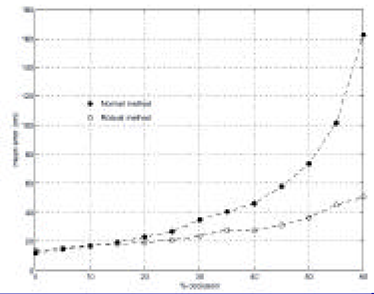


Standard approach Robust approach



## Mean error of localisation

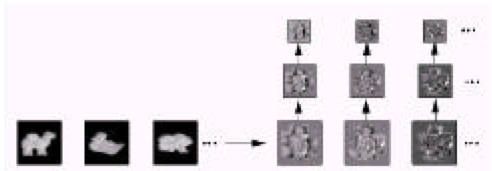
- Mean error of localisation with respect to % of occlusion





## Multiresolution coefficient estimation

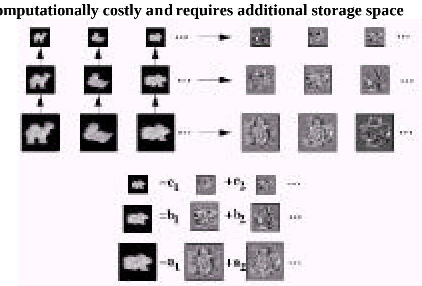
- **Multiresolution**
  - a well-known technique to reduce computational complexity
  - a search for the solution at the coarsest level and then a refinement through finer scales
- **Standard eigenspace method cannot be applied in an ordinary multiresolution way — it relies on the orthogonality of eigenimages.**





## Standard multiresolution coefficient estimation

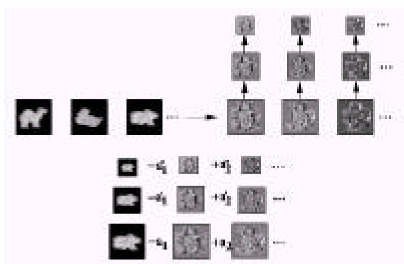
- **Eigenimages in each resolution layer are computed from a set of templates in that layer**
- **Computationally costly and requires additional storage space**





## 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:**

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

**Convolution:**

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

**Sub-sampling:**

$$\tilde{x}_1(\mathbf{r}_j) = \sum_{i=1}^{p} a_i e_{i1}(\mathbf{r}_j) \ .$$

**Convolution & Sub-sampling:**

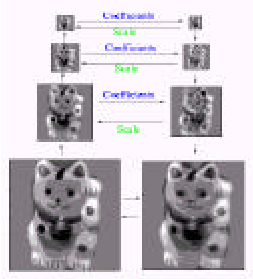
$$(f * \tilde{x})_1(\mathbf{r}_j) = \sum_{i=1}^{p} a_i (f * e_i)_1(\mathbf{r}_j) \ .$$

**Same coefficients on convolved and sub-sampled eigenimages**



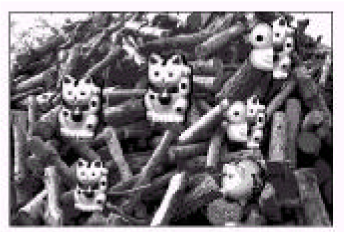
## Multiresolution approach

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



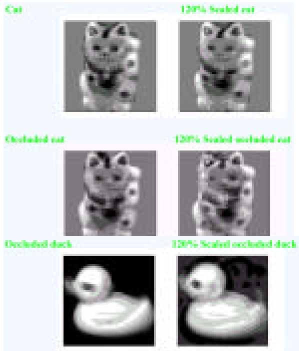


## Experimental results – test image





## Experimental results






## Illumination insensitive recognition

- Recognition of objects under varying illumination
  - global illumination changes
  - highlights
  - shadows
- Dramatic effects of illumination on objects appearance
- Training set under a single ambient illumination

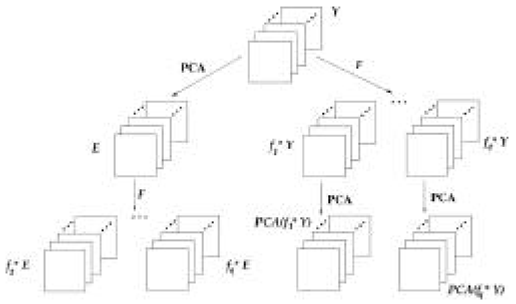


## Illumination insensitive recognition

**Our Approach**

- *Global* eigenspace representation
- *Local* gradient based filters
- Efficient combination of global and local representations
- Robust coefficient recovery in eigenspaces

## Eigenspaces and filtering






## Filtered eigenspaces

$$y_{r_i} = \sum_{j=1}^{n} q_j e_{j r_i} \quad 1 \le i \le k$$

$$(f * x)(r) = \sum_{i=1}^{p} q_i (f * e_i)(r)$$



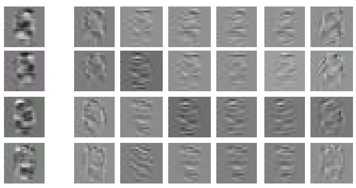
## Gradient-based filters

**Global illumination**

⇩

**Gradient-based filters**

Steerable filters [Simoncelli]





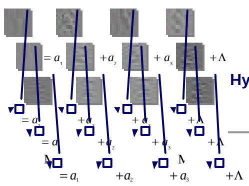
## Robust coefficient recovery

**Highlights and shadows**

⇩

**Robust coefficient recovery**

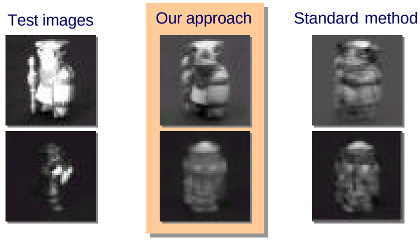
Robust solution of linear equations



**Hypothesize & Select**



## Experimental results




→ Demo



## Experimental results

Robust filtered method - all eigenvectors used

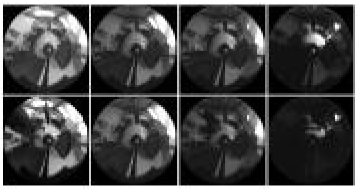
| obj. | 1 | 2 | 3 | 4 | 5 | % | ang. |
|---|---|---|---|---|---|---|---|
| 1 | 360 | 0 | 0 | 0 | 0 | 100.0 | 5.25 |
| 2 | 0 | 308 | 16 | 0 | 0 | 95.1 | 10.55 |
| 3 | 0 | 0 | 504 | 0 | 0 | 100.0 | 1.05 |
| 4 | 19 | 4 | 3 | 332 | 2 | 92.2 | 3.37 |
| 5 | 15 | 2 | 17 | 0 | 578 | 94.4 | 3.34 |
| avg. | | | | | | 96.4 | 4.19 |

Standard method - all eigenvectors used

| obj. | 1 | 2 | 3 | 4 | 5 | % | ang. |
|---|---|---|---|---|---|---|---|
| 1 | 141 | 0 | 14 | 26 | 179 | 39.2 | 10.50 |
| 2 | 0 | 254 | 62 | 5 | 3 | 78.4 | 18.90 |
| 3 | 0 | 4 | 317 | 0 | 183 | 62.9 | 3.47 |
| 4 | 23 | 6 | 38 | 249 | 44 | 69.2 | 7.11 |
| 5 | 3 | 1 | 51 | 0 | 557 | 91.0 | 6.82 |
| avg. | | | | | | 70.3 | 8.53 |

## Illumination invariant localisation

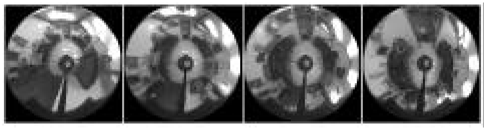
- Illumination variations and occlusions





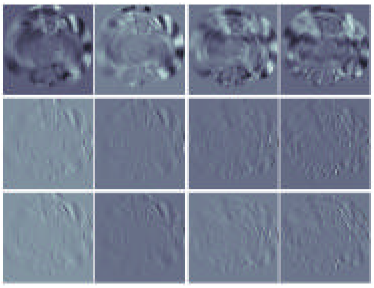
## Experimental results

- Training set: straight path, uniform illumination





## Filtered eigenvectors

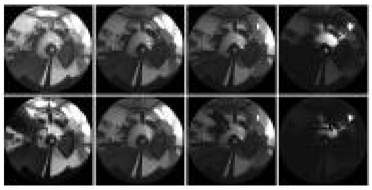




## Experimental results

Test sets T/1/2/3 without occlusion

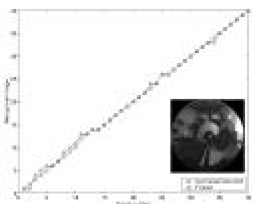
Test sets 4/5/6/7 with occlusion



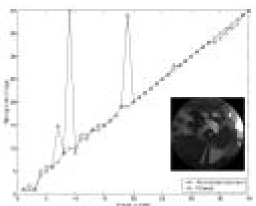


## Experimental results

- Comparison with standard method



Test set 2

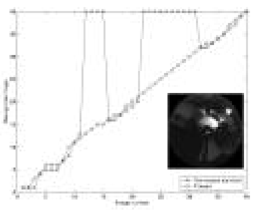


Test set 6

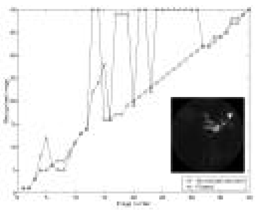


## Experimental results

- Comparison with standard method



Test set 3



Test set 7

## Experimental results

- **Average localisation error (in cm).**

| Set | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Standard | 7 | 48.7 | 73.8 | 2.5 | 13.5 | 57.8 | 108.0 |
| Normalized | 1.5 | 3.3 | 65.0 | 0.8 | 3.3 | 19.0 | 68.3 |
| Filtered | 0 | 1.3 | 4.0 | 0.5 | 1.8 | 2.3 | 14.0 |



## Building eigenspace representations

- **Rotated panoramic images**
  - Spinning eigenimages
- **Multiple eigenspaces**
- **Incremental building of eigenspaces**
- **Robust building of eigenspaces**



## Multiple Eigenspaces - Motivation

- **A single eigenspace**
  - high dimensionality
  - low-dimensional structure of data is ignored
  - poor generalisation
- **Ad-hoc partitioning of the image set is not efficient**



## Multiple eigenspaces – our goal

- **Systematically construct multiple low-dimensional eigenspaces from a set of training images**

$$\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_n | \mathbf{x}_i \in \mathbb{R}^n\}$$

- **Each image is described as a linear combination**

$$\mathbf{x}_i = \sum_{j=1}^{m} I_j^{(i)} \sum_{l=1}^{d_j} c_{jl}^{(i)} \mathbf{u}_{jl}$$

- **Design a numerically feasible and** robust **procedure**

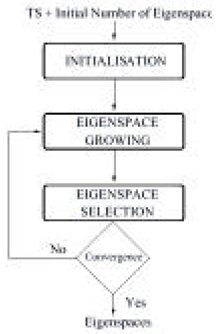


## Eigenspace growing and selection

- **A redundant set of eigenspaces by eigenspace-growing**
  - Balancing the number of images encompassed by an eigenspace
  - The dimension of the eigenspace
  - Its corresponding residual error
- **Eigenspace selection (MDL)**
- **Iterative combination of**
  - Eigenspace growing
  - Eigenspace selection



## Eigenspace growing and selection



## Eigenspace initialization

- Seed formation:
  - taking a small subset of images $\mathcal{G}_i^{(0)}$, $|\mathcal{G}_i^{(0)}| = k$, $(k << n)$ from the set $\mathcal{X}$,
  - calculating the eigenspace $\mathcal{E}_i^{(0)}(\mathcal{G}_i^{(0)})$,
  - and determining the effective dimension $p_i^{(0)}$.



## Eigenspace growing

- Search for images that are compatible with the current eigenspace
- Project, reconstruct, calculate the error, decide on inclusion.
- The eigenspace is updated $\mathcal{E}_i^{(t+1)}$ (incremental updating)
- New value of the effective dimension $p_i^{(t+1)}$ is determined
- Based on the error we accept or reject the new eigenspace
- Iterate until termination (provable).



## Redundant set of eigenspaces

- Start with a large number of seeds which are distributed in $\mathcal{X}$
- Set of eigenspaces $\mathcal{E}_i$, $(1 \leq i \leq r)$, each characterized by:
  - eigenimages of the i-th eigenspace
  - $|\mathcal{G}_i|$ . . . number of images in the i-th eigenspace
  - coefficients of the images in the i-th eigenspace,
  - $\chi_i$ . . . reconstruction error
- The redundant set of eigenspaces is a pool of hypotheses



## Eigenspace selection

MDL formulation

- $L(\mathcal{X}) = L(\mathcal{M}) + L(\mathcal{X}|\mathcal{M})$
  - $L(\mathcal{M})$: length of encoding of all eigenspaces $\mathcal{E}_i$, i.e., $p_i$ eigenimages
  - $L(\mathcal{M})$: length of encoding of coefficients for all images from $\mathcal{X}$
  - $L(\mathcal{X}|\mathcal{M})$: estimated reconstruction error, i.e., sum of eigenvalues corresponding to the truncated dimensions
- Set of eigenspaces $\mathcal{E}_i$, $(1 \leq i \leq r)$ and their individual savings:

$$S(\mathcal{E}_i(\mathcal{G}_i)) = K_0|\mathcal{G}_i| - (K_1 p_i + K_2|\mathcal{G}_i|p_i + K_3\chi_i)$$



## Eigenspace selection

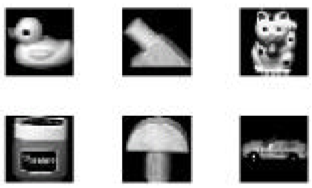
- Objective function:

$$F(\mathbf{h}) = \mathbf{h}^T\mathbf{C}\mathbf{h} = \mathbf{h}^T \begin{bmatrix} c_{11} & \cdots & c_{1R} \\ \vdots & & \vdots \\ c_{R1} & \cdots & c_{RR} \end{bmatrix} \mathbf{h}$$

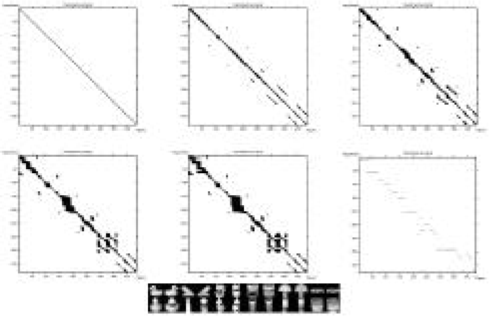
- $c_{ii} = S(\mathcal{E}_i(\mathcal{G}_i))$
- $c_{ij} = (-K_0|\mathcal{G}_i \cap \mathcal{G}_j| + K_2|\mathcal{G}_i \cap \mathcal{G}_j|\max(p_i, p_j) + K_3\chi_{ij})/2$
- Combinatorial optimization problem: (tabu search)



## Multiple eigenspaces - experiments



## Eigenspace growing and selection



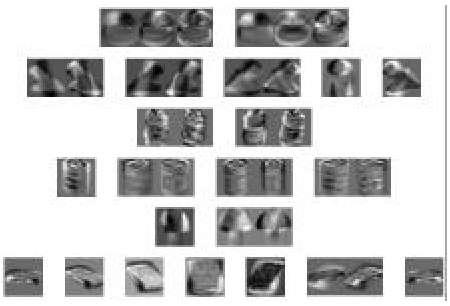


## Multiple eigenspaces

| EigenS | Dim | # of img | Mean err | EigenS | Dim | # of img | Mean err |
|---|---|---|---|---|---|---|---|
| Single ES | 11 | 432 | 0.0038 | Box-3 | 2 | 31 | 0.0039 |
| Duck-1 | 3 | 41 | 0.0031 | Box-4 | 2 | 17 | 0.0038 |
| Duck-2 | 3 | 50 | 0.0037 | Mush-1 | 1 | 20 | 0.0039 |
| Block-1 | 2 | 22 | 0.0035 | Mush-2 | 2 | 59 | 0.0039 |
| Block-2 | 2 | 24 | 0.0037 | Car-1 | 1 | 17 | 0.0039 |
| Block-3 | 2 | 19 | 0.0035 | Car-2 | 1 | 11 | 0.0038 |
| Block-4 | 1 | 7 | 0.0036 | Car-3 | 1 | 7 | 0.0039 |
| Block-5 | 1 | 10 | 0.0036 | Car-4 | 1 | 7 | 0.0039 |
| Cat-1 | 2 | 25 | 0.0039 | Car-5 | 1 | 6 | 0.0038 |
| Cat-2 | 2 | 50 | 0.0039 | Car-6 | 2 | 22 | 0.0039 |
| Box-1 | 1 | 15 | 0.0039 | Car-7 | 1 | 15 | 0.0037 |
| Box-2 | 2 | 23 | 0.0038 | | | | |

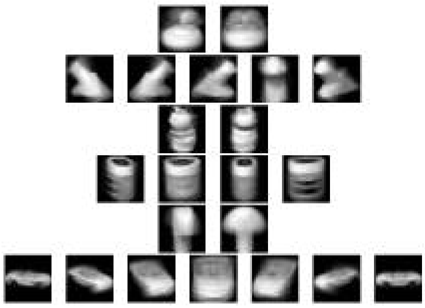


## Eigenimages of individual eigenspaces



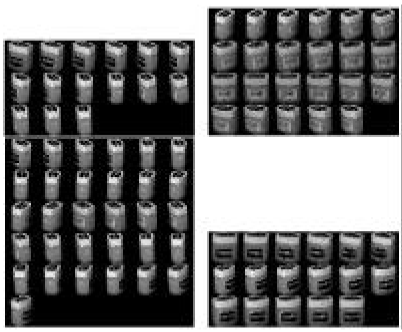


## Mean images of individual eigenspaces



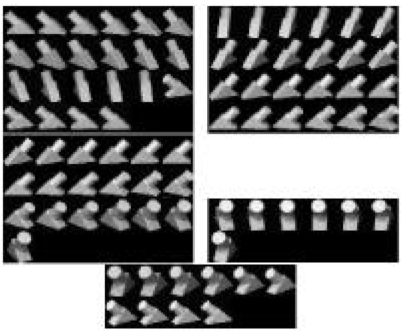


## "Box" images in four eigenspaces

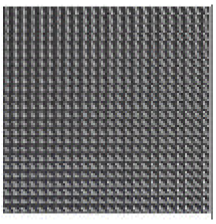


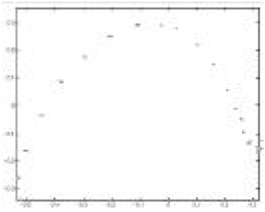


## "Block" images in five eigenspaces



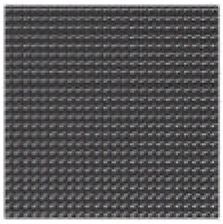
## Interpolated eigenspace

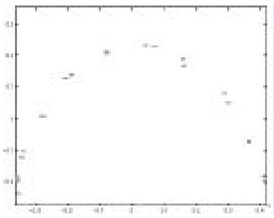






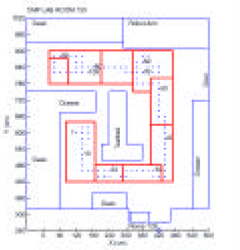
## Interpolated eigenspace





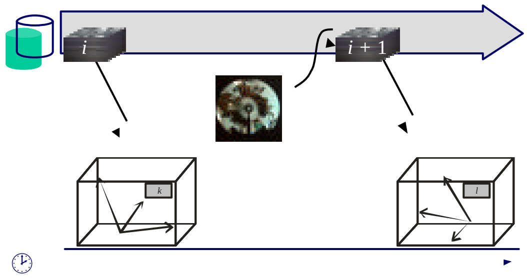


## Multiple eigenspaces



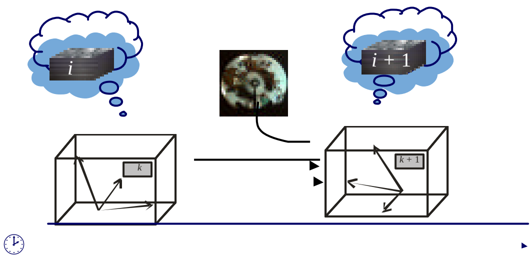


## Batch computation of PCA






## Incremental computation of PCA






## Batch method for PCA

- All input images are processed simultaneously.

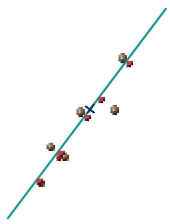
$$\mu = \frac{1}{N}\sum_{j=1}^{N} \mathbf{x}_j$$

$$\hat{\mathbf{X}} = \mathbf{X} - \mu \mathbf{1}_{1\times N}$$

$$\mathbf{C} = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^{\top}$$

$$\mathrm{SVD}(\mathbf{C}) \rightarrow \mathbf{U}, \Lambda$$

$$\mathbf{A} = \mathbf{U}^{\top}\hat{\mathbf{X}}$$



## Batch method – pros and cons

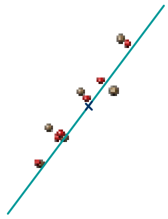
- all training images processed simultaneously

+ simple
+ optimal (in the sense of MSRE)
+ easy to choose subspace dimension
– all training images have to be given in advance
– computationally unfeasible for large sets of training images
– model built once forever



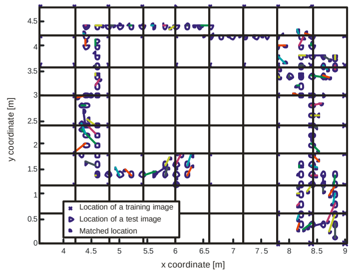
## Incremental approach

- Input images are being processed sequentialy.





## Localization with incremental method





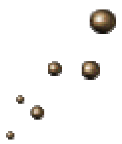

## Incremental method - Conclusions

- Incremental method does not significantly degrade the results of the batch method.
- If the training images are discarded after the update, the results are degraded, but still good.
- Better results are achieved, if heterogenuous images are present in the beggining of the training sequence.
- Discarding criterion can significantly influence the results.

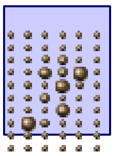


## Weighted influence

- Input images are treated selectively.
- Pixels within an image are treated selectively.



temporal weights ${}^{t}w$



spatial weights ${}^{s}w$



## Weighted PCA

- Weighted influence of pixels
- Minimizing weighted squared reconstruction error:

$$\mathcal{E} = \sum_{i=1}^{M} \sum_{j=1}^{N} w_{ij} \left( \hat{x}_{ij} - \sum_{l=1}^{k} u_{il} a_{lj} \right)^2$$

- Weighted mean: $\mu_i = \frac{\sum_{j=1}^{N} w_{ij} x_{ij}}{\sum_{j=1}^{N} w_{ij}}, \quad i = 1 \ldots M$

## Temporal weights

- Row vector ${}^t\mathbf{w}$ contains weights for all images.
- All the pixels in an image are equally treated
  - $\mathbf{W} = \mathbf{1}_{M\times 1}{}^t\mathbf{w}$
- Weighted squared reconstruction error:

$$\mathcal{E} = \sum_{i=1}^{M}\sum_{j=1}^{N} {}^tw_j \left( x_{ij} - \sum_{l=1}^{k} u_{il}a_{lj} \right)^2$$

- Weighted mean:

$$\boldsymbol{\mu} = \frac{1}{\sum_{j=1}^{N} {}^tw_j} \sum_{j=1}^{N} {}^tw_j \mathbf{x}_j$$

- Maximize weighted variance of projections.
- SVD of weighted covariance matrix.



## Algorithm for TWPCA

**Input**: data matrix $\mathbf{X}$ temporal weights ${}^t\mathbf{w}$.
**Output**: mean value $\boldsymbol{\mu}$, eigenvectors $\mathbf{U}$, eigenvalues $\boldsymbol{\lambda}$.

1. Estimate the weighted mean vector: $\boldsymbol{\mu} = \frac{1}{\sum_{j=1}^{N} {}^tw_j} \sum_{j=1}^{N} {}^tw_j \mathbf{x}_j$
2. Scale the inputa data centered around the weighted mean: ${}^t\hat{\mathbf{x}}_j = \sqrt{{}^tw_j}(\mathbf{x}_j - \boldsymbol{\mu}),\ j = 1\ldots N$
3. **if** *M* < *N* **then**
4. Estimate the weighted covariance matrix: $\mathbf{C} = \frac{1}{\sum_{j=1}^{N} {}^tw_j} {}^t\hat{\mathbf{X}}{}^t\hat{\mathbf{X}}^\top$.
5. Perform SVD on $\mathbf{C}$. Obtain eigenvectors $\mathbf{U}$ and eigenvalues $\boldsymbol{\lambda}$.
6. **else**
7. Estimate weighted inner product matrix: $\mathbf{C}' = \frac{1}{\sum_{j=1}^{N} {}^tw_j} {}^t\hat{\mathbf{X}}^\top {}^t\hat{\mathbf{X}}$.
8. Perform SVD on $\mathbf{C}'$. Obtain eigenvectors $\mathbf{U}'$ and eigenvalues $\boldsymbol{\lambda}'$.
9. Determine eigenvectors $\mathbf{U}$ $\mathbf{u}_i = \frac{{}^t\hat{\mathbf{X}}\mathbf{u}'_i}{\sqrt{\sum_{j=1}^{N} {}^tw_j}\sqrt{\lambda'_i}},\ i = 1\ldots N$.
10. Determine eigenvalues $\boldsymbol{\lambda} = \boldsymbol{\lambda}'$.
11. **end if**



## EM algorithm for PCA

- Minimizing squared reconstruction error.

$$\mathcal{E} = \sum_{i=1}^{M}\sum_{j=1}^{N} \left( \hat{x}_{ij} - \sum_{l=1}^{k} u_{il}a_{lj} \right)^2$$

- EM algorithm for estimating principal subspace:

E-step: $\forall j: \ \hat{x}_{ij} = \sum_{p=1}^{k} u_{ip}a_{pj}\ ,\ i = 1\ldots M$

M-step: $\forall i: \ \hat{x}_{ij} = \sum_{p=1}^{k} u_{ip}a_{pj}\ ,\ j = 1\ldots N$



## Introducing general weights

- Minimizing **weighted** squared reconstruction error.

$$\mathcal{E} = \sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij} \left( \hat{x}_{ij} - \sum_{l=1}^{k} u_{il}a_{lj} \right)^2$$

- EM algorithm for estimating principal subspace:

E-step: $\forall j: \ \sqrt{w_{ij}}\,\hat{x}_{ij} = \sqrt{w_{ij}}\sum_{p=1}^{k} u_{ip}a_{pj}\ ,\ i = 1\ldots M$

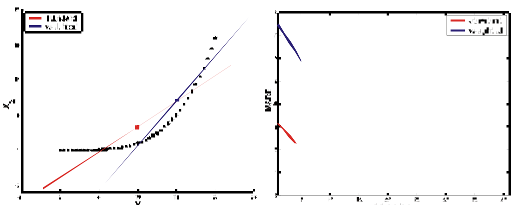
M-step: $\forall i: \ \sqrt{w_{ij}}\,\hat{x}_{ij} = \sqrt{w_{ij}}\sum_{p=1}^{k} u_{ip}a_{pj}\ ,\ j = 1\ldots N$

- Weighted mean: $\mu_i = \frac{\sum_{j=1}^{N} w_{ij}x_{ij}}{\sum_{j=1}^{N} w_{ij}}\ ,\ i = 1\ldots M$



## Simple example – temporal weights

- Temporal weights ${}^tw_j = j^2$.

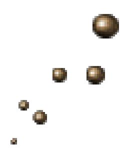


| | standard | weighted |
|---|---|---|
| WSRE | 1.78 | 0.97 |



## Introducing temporal weights in IPCA

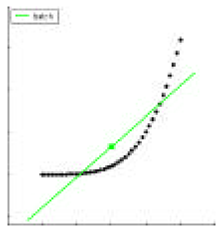
$\mathbf{a} = \mathbf{U}^{(n)\top}(\mathbf{x} - \boldsymbol{\mu}^{(n)})$
$\mathbf{y} = \mathbf{U}^{(n)}\mathbf{a} + \boldsymbol{\mu}^{(n)}$
$\mathbf{r} = \mathbf{x} - \mathbf{y}$
$\mathbf{U}' = \begin{bmatrix} \mathbf{U}^{(n)} & \frac{\mathbf{r}}{\|\mathbf{r}\|} \end{bmatrix}$
$\mathbf{A}' = \begin{bmatrix} \mathbf{A}^{(n)} & \mathbf{a} \\ \mathbf{0} & \|\mathbf{r}\| \end{bmatrix}$
$\mathrm{TWPCA}(\mathbf{A}', {}^t\mathbf{w}) \to \boldsymbol{\mu}'', \mathbf{U}'', \boldsymbol{\lambda}''$
$\mathbf{U}'' = [\mathbf{u}''_1, \ldots, \mathbf{u}''_{k(n+1)}]$
$\mathbf{A}^{(n+1)} = \mathbf{U}''^\top(\mathbf{A}' - \boldsymbol{\mu}''\mathbf{1}_{1\times n+1})$
$\mathbf{U}^{(n+1)} = \mathbf{U}'\mathbf{U}''$
$\boldsymbol{\mu}^{(n+1)} = \boldsymbol{\mu}^{(n)} + \mathbf{U}'\boldsymbol{\mu}''$
$\boldsymbol{\lambda}^{(n+1)} = [\lambda''_1, \ldots, \lambda''_{k(n+1)}]$



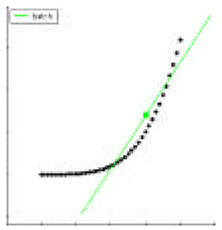
temporal weights ${}^t\mathbf{w}$

## Simple example – temporal weights

- Temporal weights $w_j = j^2$.



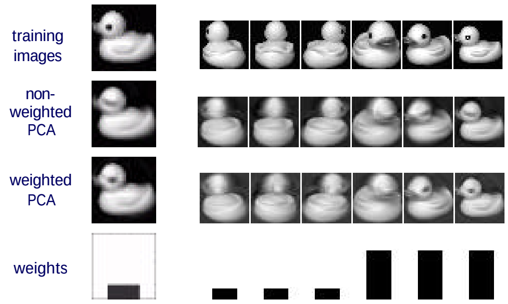
non-weighted



weighted

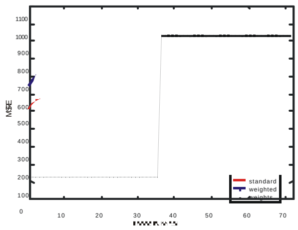


## Experimental results - WPCA






## Experimental results - WPCA

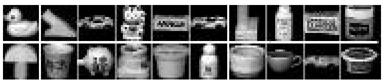



| | standard | weighted |
|---|---|---|
| WSRE | 660 | 605 |



## Experimental results - WIPCA

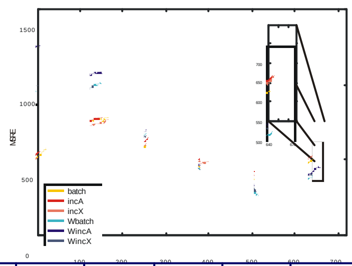
- Training images: 720 images of 20 objects from COIL20 database.



- Temporal weights proportional to the second power of the image index.
- Comparing weighted squared reconstruction error.



## Experimental results - WIPCA



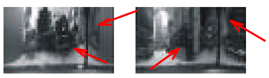

| | batch | incA | incX | Wbatch | WincA | WincX |
|---|---|---|---|---|---|---|
| WSRE | 617 | 658 | 648 | 554 | 583 | 565 |



## Experimental results – mobile robot

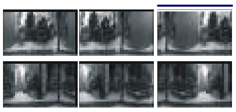
- **Navigation of a mobile robot**

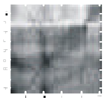
- Problem: self-occlusions



## Experimental results – mobile robot

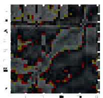
- **Spinning images**





non-weighted

- Weights

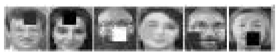


weighted

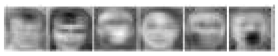


## Robust learning method for PCA

- **If the training images are occluded…**



... the occlusions are incorporated in the representation:

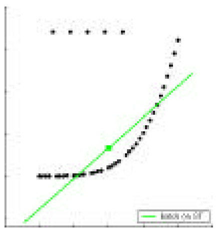


Robust learning algorithm: detect outliers and build the representation from inliers only (set weights of outliers to 0).

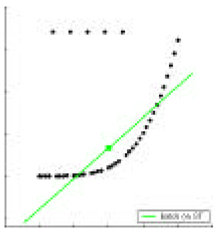


## Simple example – RIPCA

- **Simple 2D example**



non robust

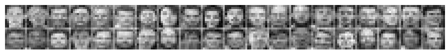


robust



## Experimental results - RIPCA

- ORL face database: 40 persons, 10 images per person.
- 40 non-occluded training images

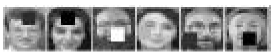
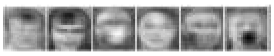
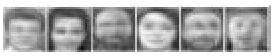


- 360 occluded training images

. . .



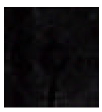
## Experimental results - RIPCA

| | | MSRE to ground thruth |
|---|---|---|
| training images |  | |
| non-robust |  | 915 |
| robust |  | 710 |
| optimal |  | 594 |

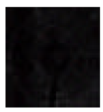


## Experimental results - RIPCA

- **Robust background modeling**
  - model illumination variation
  - discard outliers

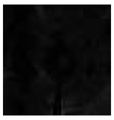


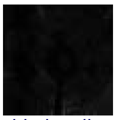
training images



reconstructed images

## Experimental results – synthetic data


ground truth


added outliers


standard PCA 2PC

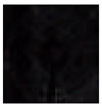

standard PCA 8PC

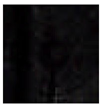

robust PCA 8PC

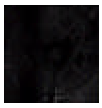


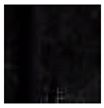
## Experimental results - RIPCA

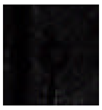
- Robust background modeling


training im.

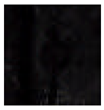

batchOnGT


batchStd


batchRob


incKnownOL


incPoorSeed

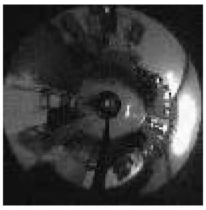

incNonDispSeed


incGoodSeed



## Robust Subspace Learning

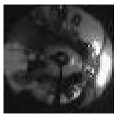
- Subspace learning from data containing outliers:
  - Detect outliers
  - Learn using only inliers.

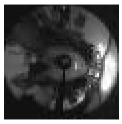


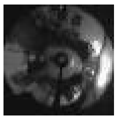
[D. Skocaj, A. Leonardis, H. Bischof: A robust PCA algorithm for building representations from panoramic images, ECCV 2002]

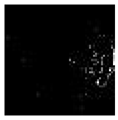


## Experimental results – real data


input


standard PCA


robust PCA


outliers



## Research issues

- Comparative studies (e.g., LDA versus PCA, PCA versus ICA)
- Robust learning of other representations (e.g. LDA, CCA)
- Integration of robust learning with modular eigenspaces
- Local versus Global subspace represenations
- Combination of subspace representations in a hierarchical framework



## Further readings

- Recognizing objects by their appearance using eigenimages (SOFSEM 2000, LNCS 1963)
- Robust recognition using eigenimages (CVIU 2000, Special Issue on Robust Methods in CV)
- Illumination insensitive eigenspaces (ICCV 2001)
- Mobile robot localization under varying illumination (ICPR 2002)
- Eigenspace of spinning images (OMNI 2000, ICPR 2000, ICAR 2001)
- Incremental building of eigenspaces (ICRA 2002, ICPR 2002)
- Multiple eigenspaces (Pattern Recognition, 2002)
- Robust building of eigenspaces (ECCV 2002)
- PhD Thesis, Danijel Skocaj (February, 2003)