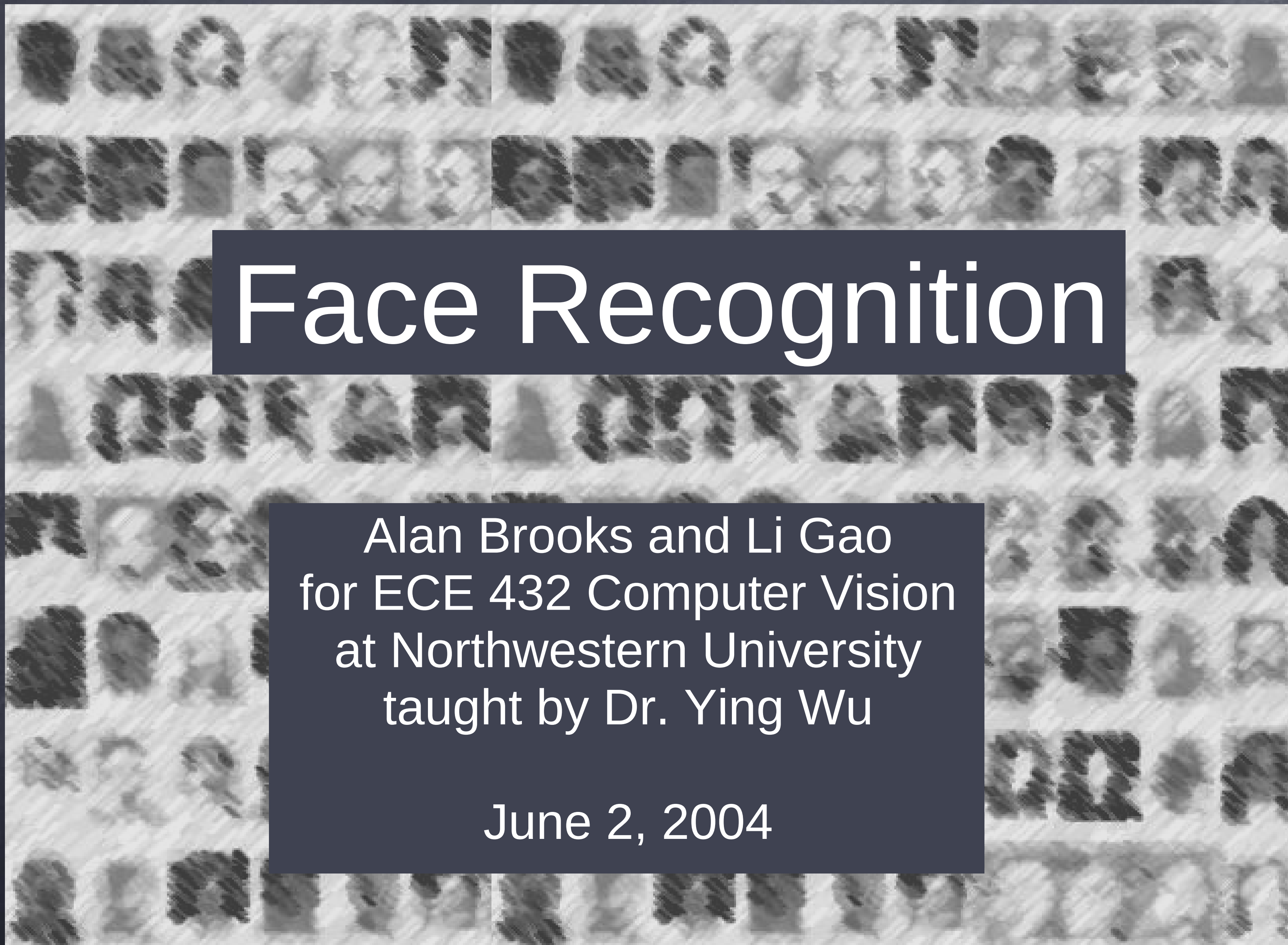


Face Recognition

Alan Brooks and Li Gao
for ECE 432 Computer Vision
at Northwestern University
taught by Dr. Ying Wu

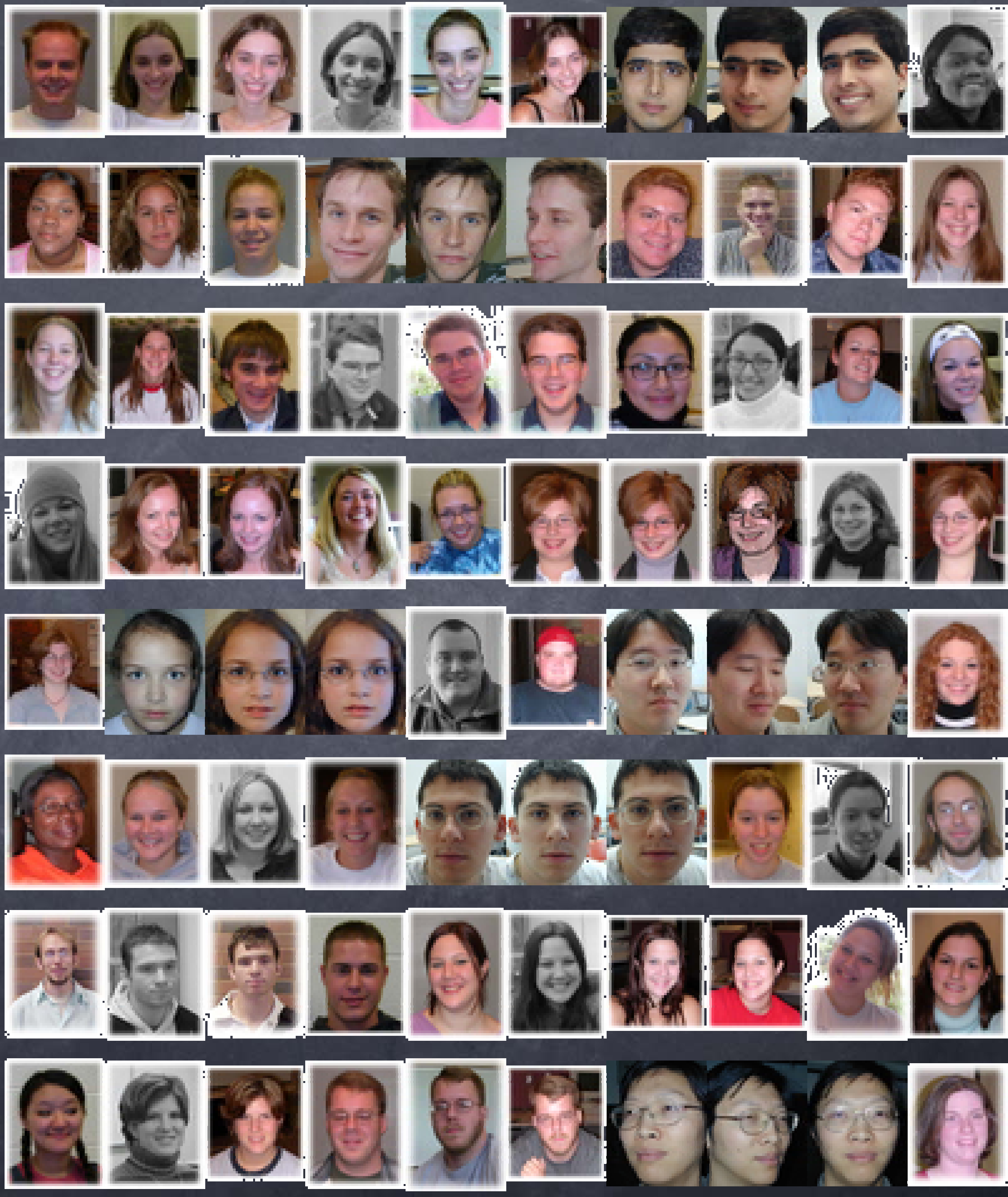
June 2, 2004

Outline

- Problem & History
- Mathematics
- Results
  - UMIST (controlled lighting)
  - ALAN (typical face snapshots)
- Ideas for Further Work

Problem Definition

Who is this?



Assumptions

- Face detection is done, so we know:
  - scale,
  - rotation,
  - and alignment

CMU's face detection online demo
http://www.vasc.ri.cmu.edu/cgi-bin/demos/findface.cgi
detected 77/80 faces with defaults. See results here:
http://vasc.ri.cmu.edu/demos/faceindex/05312004/users/4333.html.

History: Influential Contributions

- 1964, 1970, 1977: Facial feature-based recognition (Bledsoe, Kelly Kanade)
- 1984: WIZARD neural net approach (Stonham)
- 1991, 1994: Eigenface PCA (Pentland & Turk)
- 1997: Fisherface FLD + PCA (Belheumeur)
- 2000: FERET standard testing method & database
- 2002: Indep Component Analysis (ICA) captures higher-order statistics (Bartlett)
- 2003: Kernel, SVM, RBF, combos (Liu, Er)
- 2004: Plenoptic light-fields (Gross), 2D-PCA (Yang)

our focus

Chellappa's "Human and Machine Recognition of Faces: A Survey" gives a very good historical summary.

Mathematics: Eigenface

Given:

training images

#1

N  N

...

#M

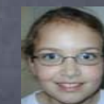
recognition images

#1

N  N

...

#c



Mp = desired # of principal components

Feature Extraction:

$X = [x_1\ x_2\ \dots\ x_m]$

column vector for each train face

```
me = mean(X,2)
```

average face 

```
A = X - [me me … me]
```

```
[U,E,V] = svd(A,0)
```

avoids $N^2 x N^2$ matrix computation of [V,D]=eig(A*A')
only computes M columns of U: A=U*E*V'

```
eigVals = diag(E)
lmda = eigVals(1:Mp)
P = U(:,1:Mp)
```

pick face-space principal components (eigenfaces)

```
train_wt = P'*A
```

store weights of training data projected into eigenspace

Nearest-Neighbor Classification:

```
recog_wt = P'*A2
```

A2 created from the recog data

```
euDis(i,j) = sqrt((recog_wt(:,j)-train_wt(:,i)).^2)
```

i^{th} recog face, j^{th} train face

Mathematics: Fisherface

Given:

training images

#1

N  N

...

#M

recognition images

#1

N  N

...

#c

P_1 = eigenface result (used by fisherface)

Feature Extraction:

```
A = X - [me me … me]
```

same as eigenface

```
for i=1:c
    Sb = Sb + clsMean_i*clsMean_i’
```

computes N^2xN^2 **between**-class scatter matrix

```
for i=1:c, j=1:c_i
    Sw = Sw + (X(j)-clsMean_i)*(X(j)-clsMean_i)’
```

computes N^2xN^2 **within**-class scatter matrix

```
Sbb = P_1’*Sb*P_1
Sww = P_1’*Sw*P_1
```

PCA project into (N-c) x (N-c) subspace

```
[V,D] = eig(Sbb,Sww)
```

generalized eigenvalue decomposition solves Sbb*V = Sww*V*D

```
eigVals = diag(D)
lmda = eigVals(1:Mp)
P = P1*V(:,1:Mp)
train_wt = P’*A
```

store training weights

Databases

UMIST

- By Daniel B Graham
- http://images.ee.umist.ac.uk/danny/database.html
- 20 people, 565 images
  (avg of 28 images per person)
- Constructed under controlled lighting & background conditions, with a purpose of being used for facial recognition algorithm testing.
- Varied pose uniformly from frontal to side view.

ALAN

- By Alan Brooks
- http://pubweb.northwestern.edu/~acb206/ece432/tmbwClassOval40x60.zip
- 14 people, 47 images
  (avg of 3 images per person)
- Gathered from collections of snapshots taken at different times. No controlled lighting or backgrounds.
- Pre-processed by hand to align, normalize, and remove background.

UMIST Database

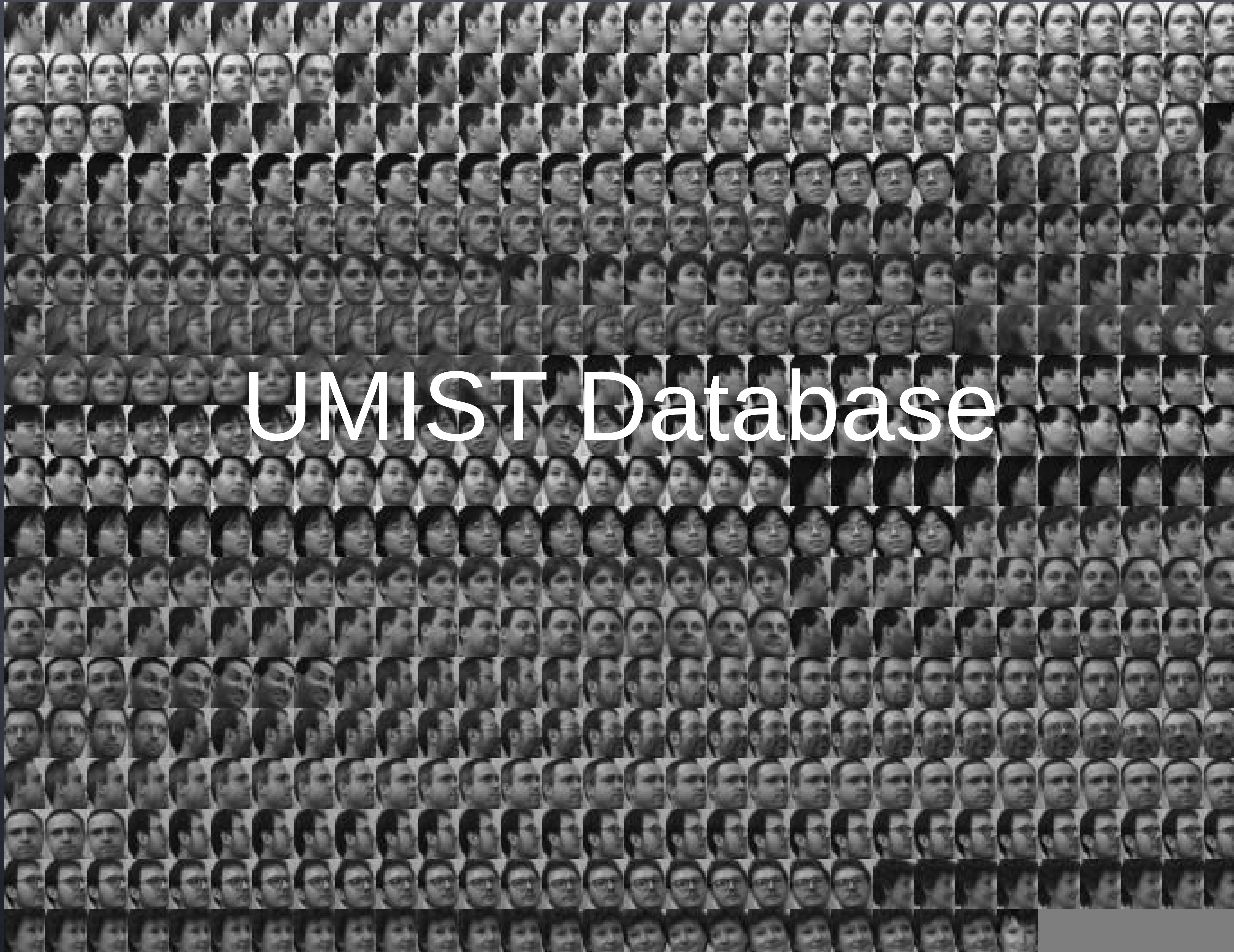


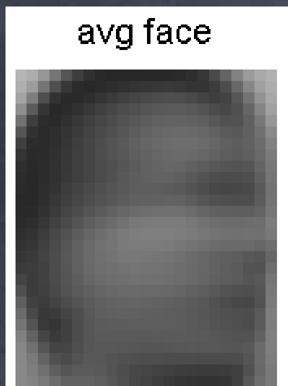
Image produced using ImageMagick’s “montage” command like this:

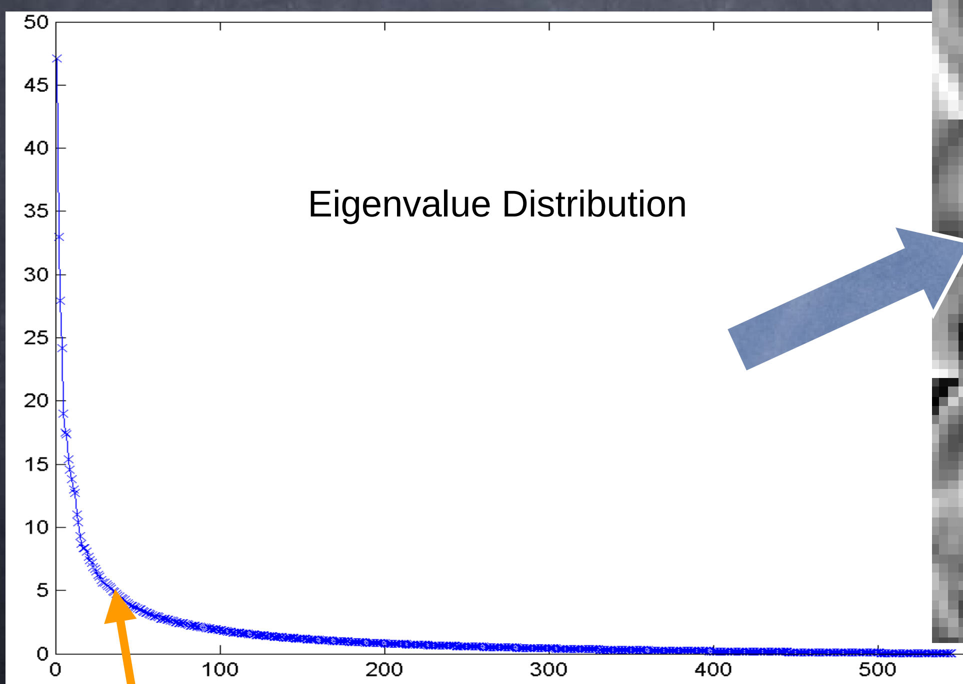
```
$ montage +frame +shadow +label -tile 30x20 -geometry 23x28+0+0 -background gray *.* joined.jpg
```

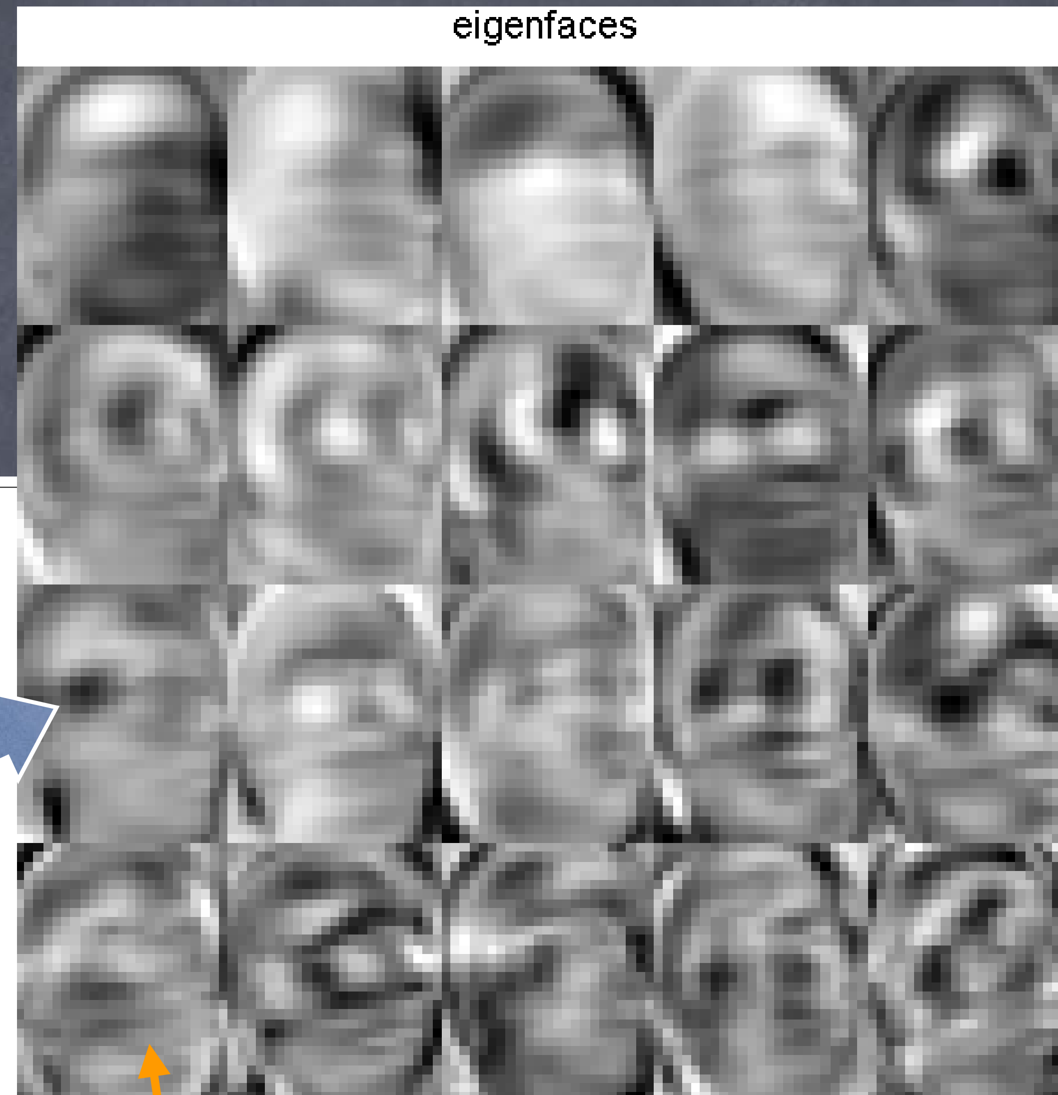
UMIST Results Using Eigenface Approach

- 545 training faces from 20 people
- 20 recognition faces (randomly picked from database)

Feature Extraction





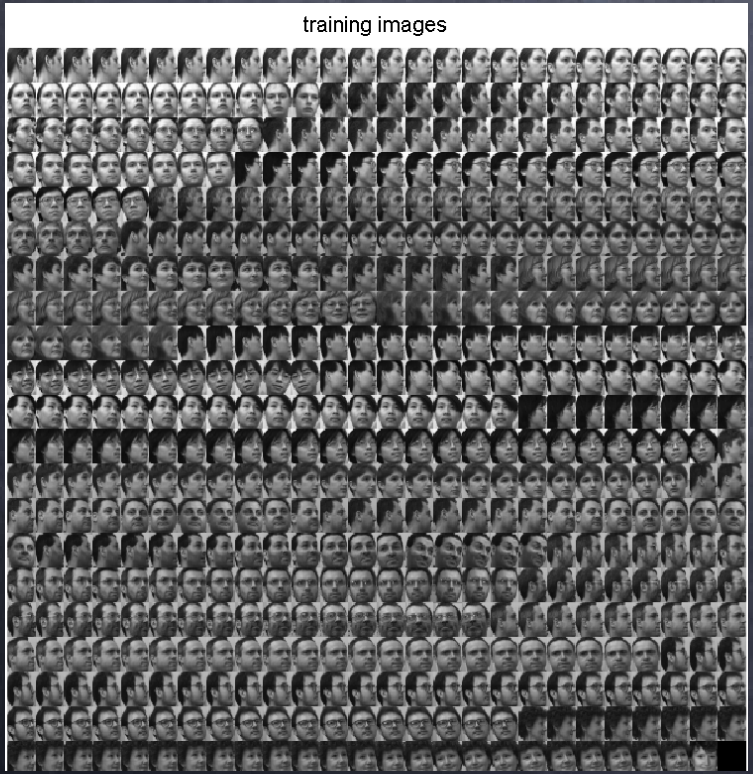


20 largest eigenvalues

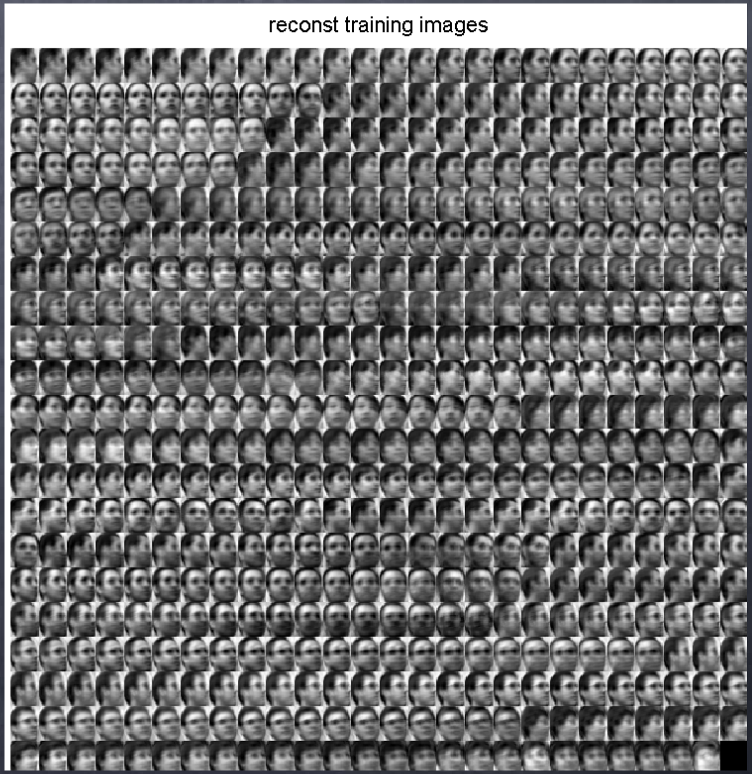
20 corresponding eigenfaces (eigenvectors)

Training

- Keeping Mp (20) principal components, project all training data into subspace and store projection weights.
- Weights can be used to reconstruct faces.

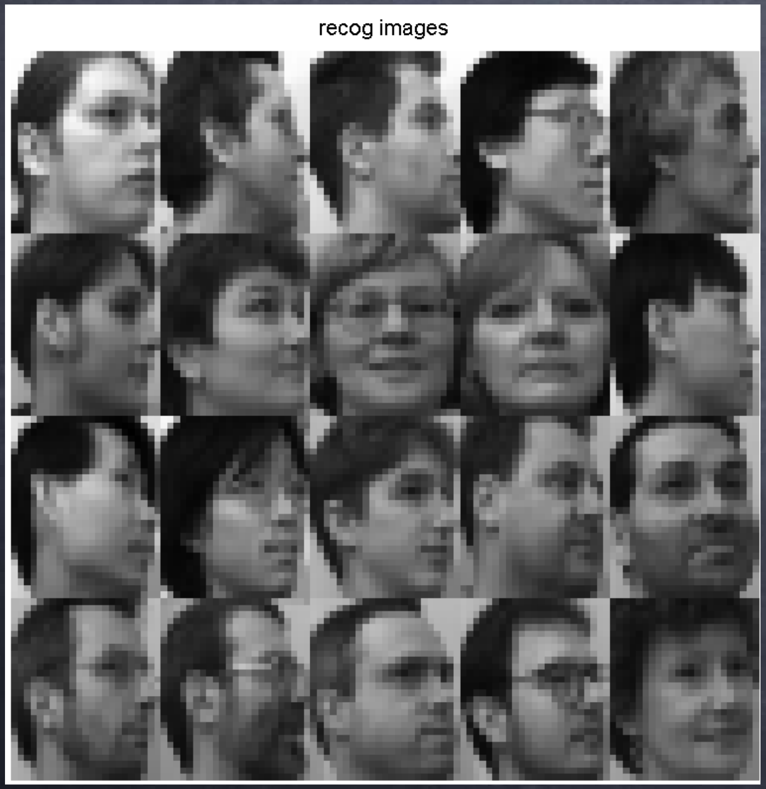


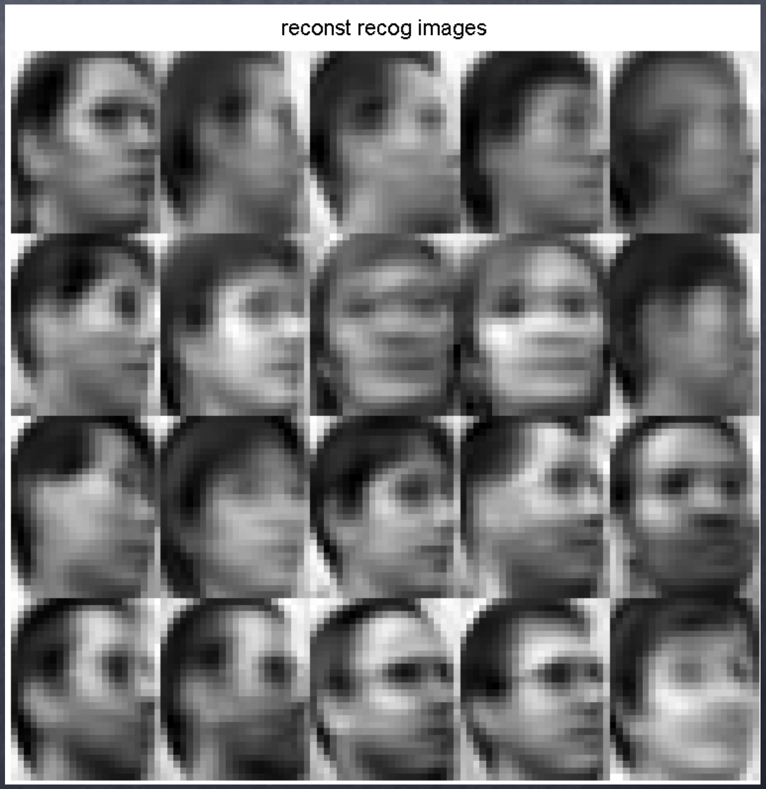
Store weights as training data



Projection

- Project new faces into eigenface-space.





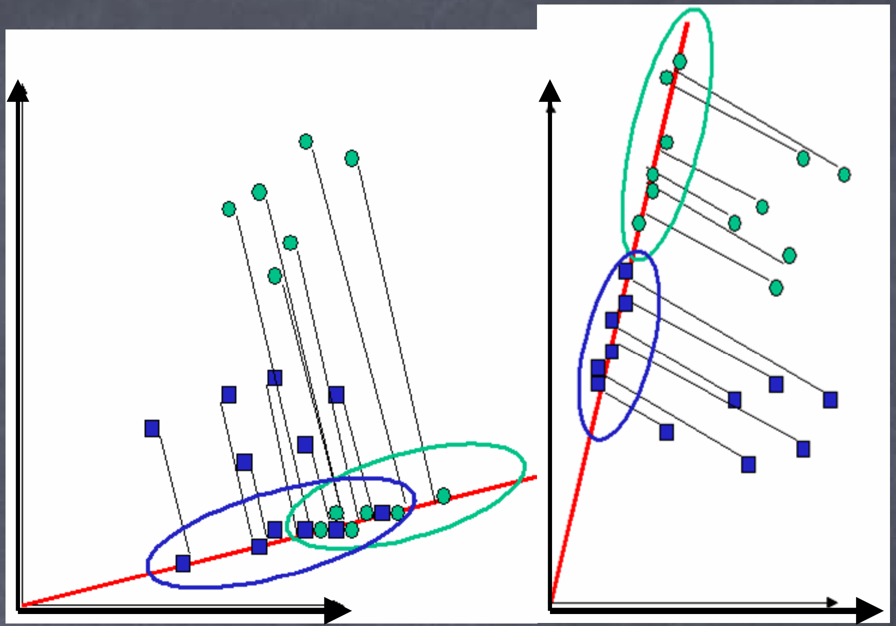
Classification

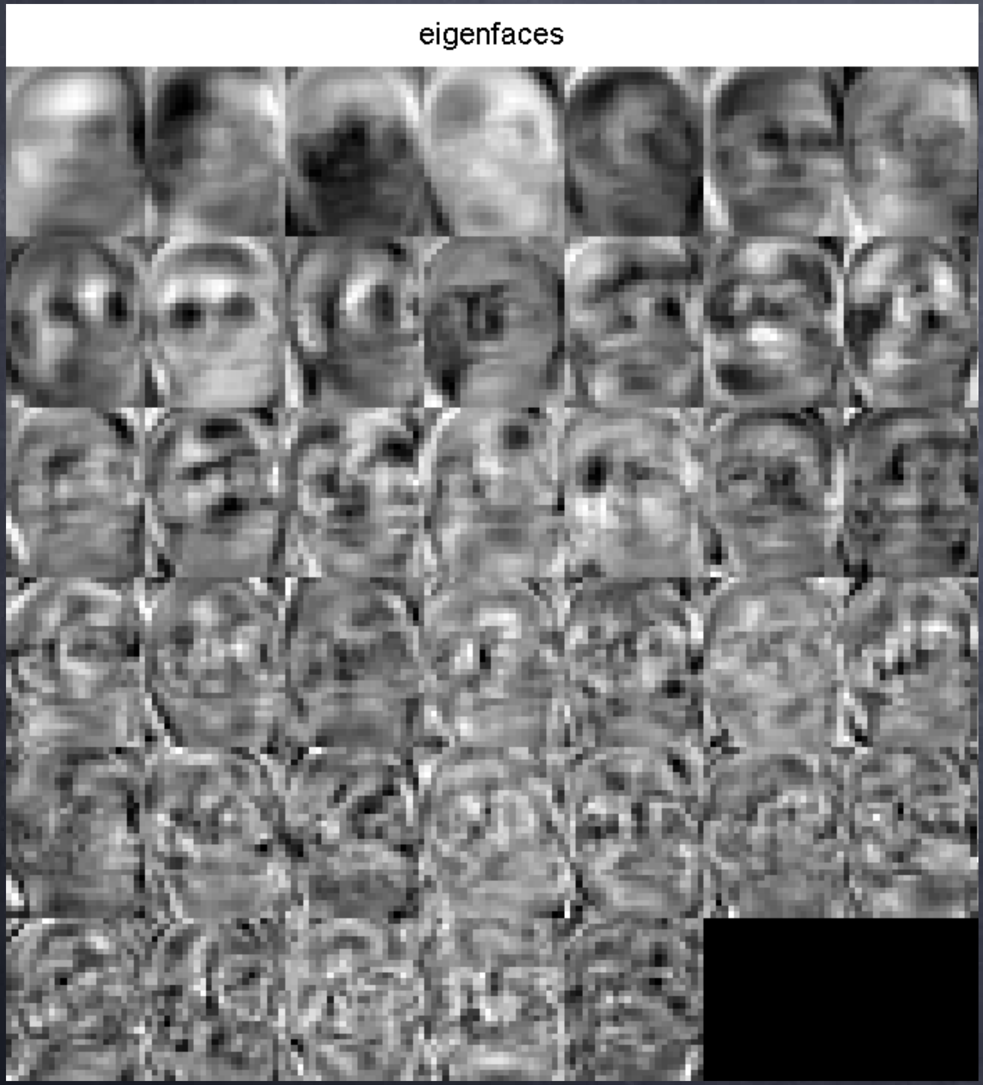
- Classify by comparing projection weights of new faces to known face weights.
- 20 of 20 (100%) faces correctly classified.

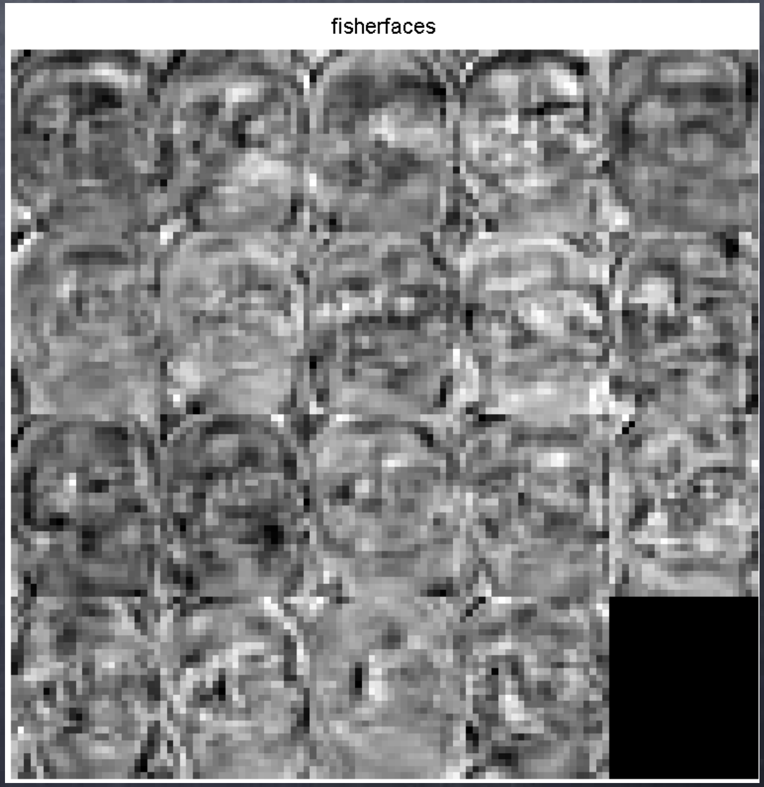
UMIST Results Using Fisherface Approach

- Subset of UMIST data
- 60 training faces from 20 people (front, angled, profile pose)
- 20 recognition faces (randomly picked from database)

Feature Extraction

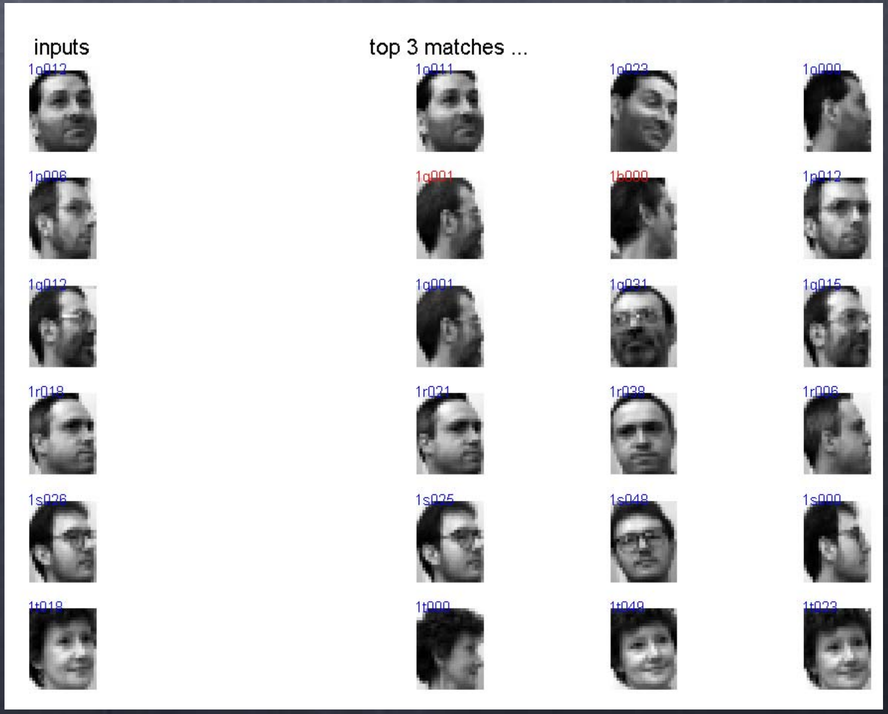






Classification

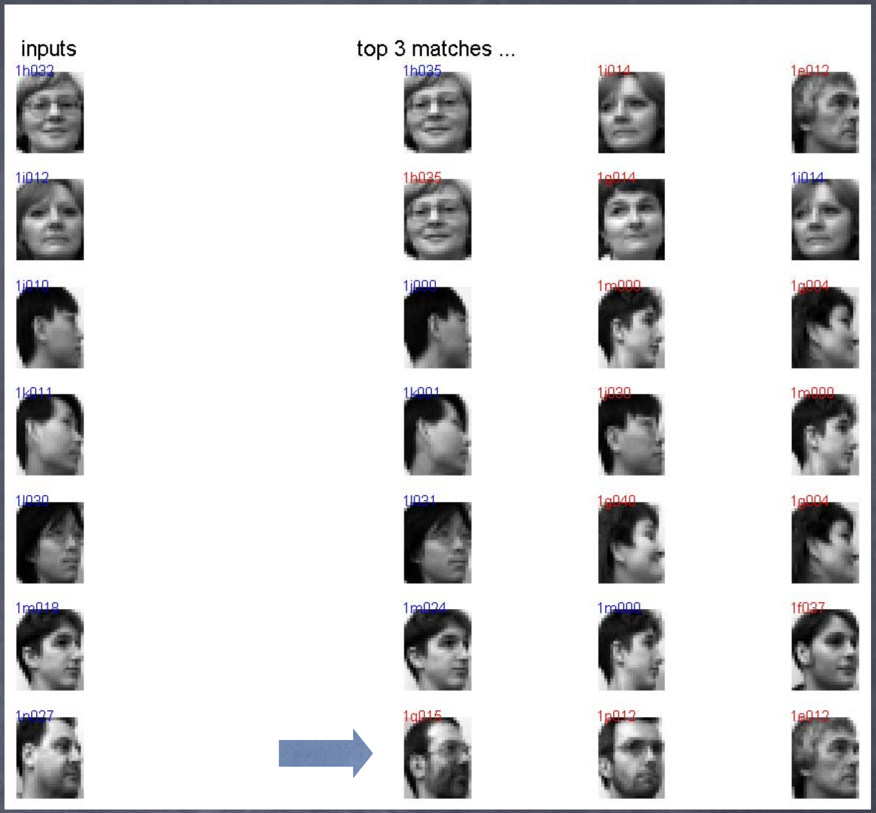
- Fisherface: 16 of 20 (80%) faces correctly classified.
- Pose invariant in top 3: often (13 times) picks all 3 correct database images.

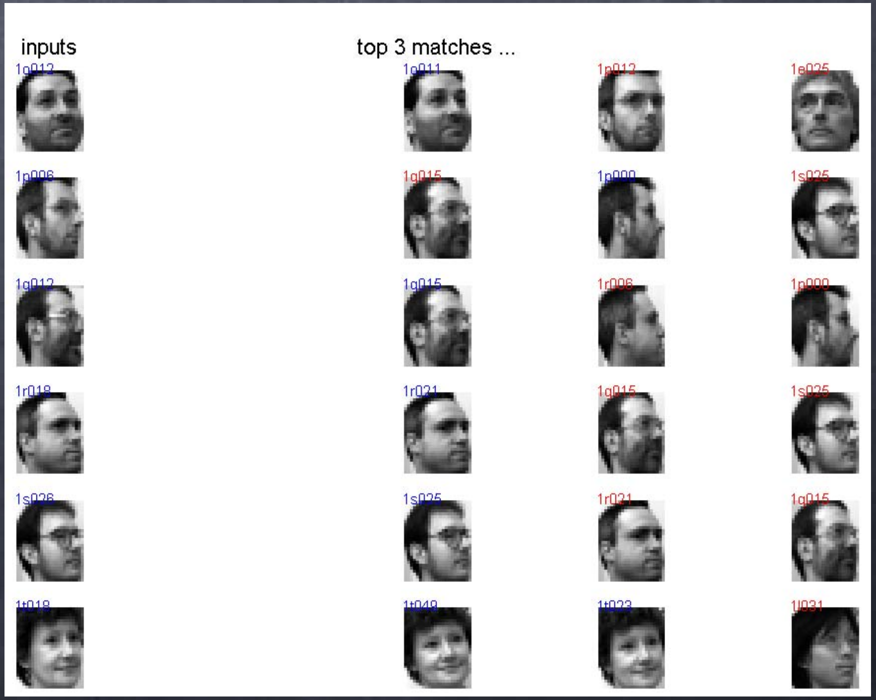


Classification

For comparison:

- Eigenface (with the same data): only 14 of 20 (70%) were correctly classified.
- Picks same pose, wrong people in top 3. Never gets all 3 correct poses.





Fisherface (FLD) vs. Eigenface (PCA)

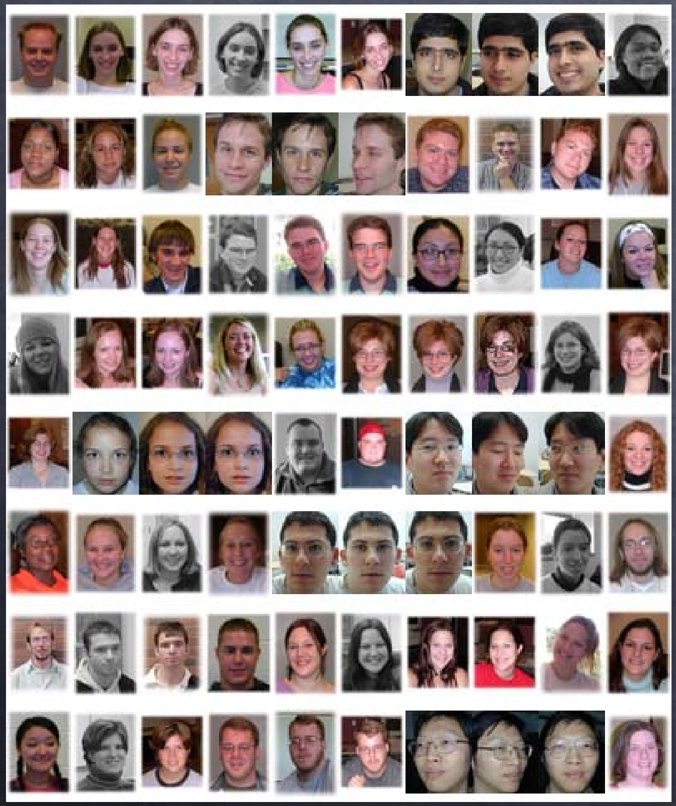
	Fisherface	**Eigenface**
Computational Complexity	slightly more complex	simple
Effectiveness across Pose	good, even with limited data	some, with enough data
Sensitivity to Lighting	little	very

ALAN Database Results

- 26 training faces from 11 people
- 11 recognition faces (randomly picked)

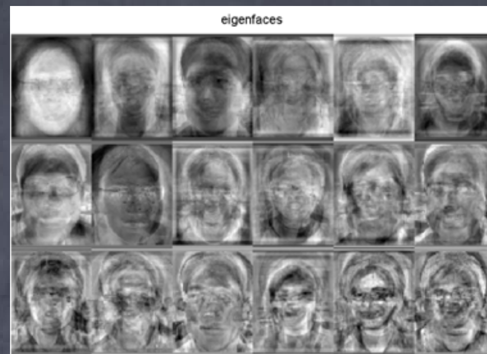
Preprocessing Important

- Lighting, scale, alignment, background

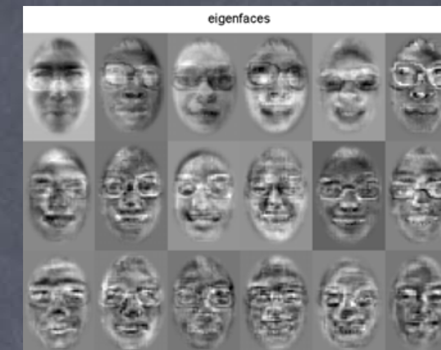


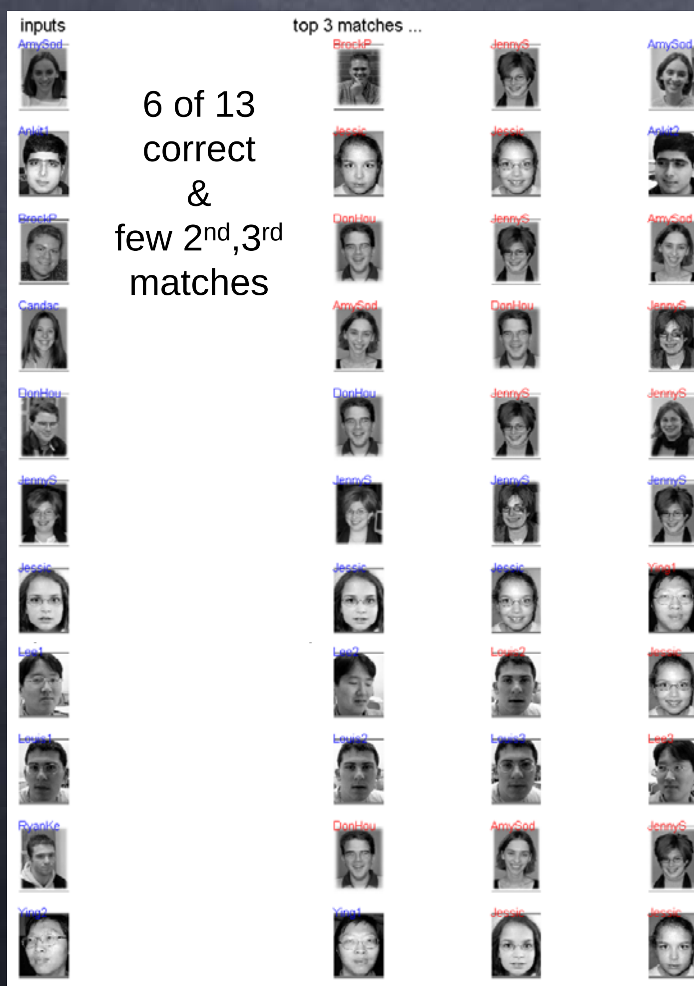
Performance Improvement

Before Pre-Processing

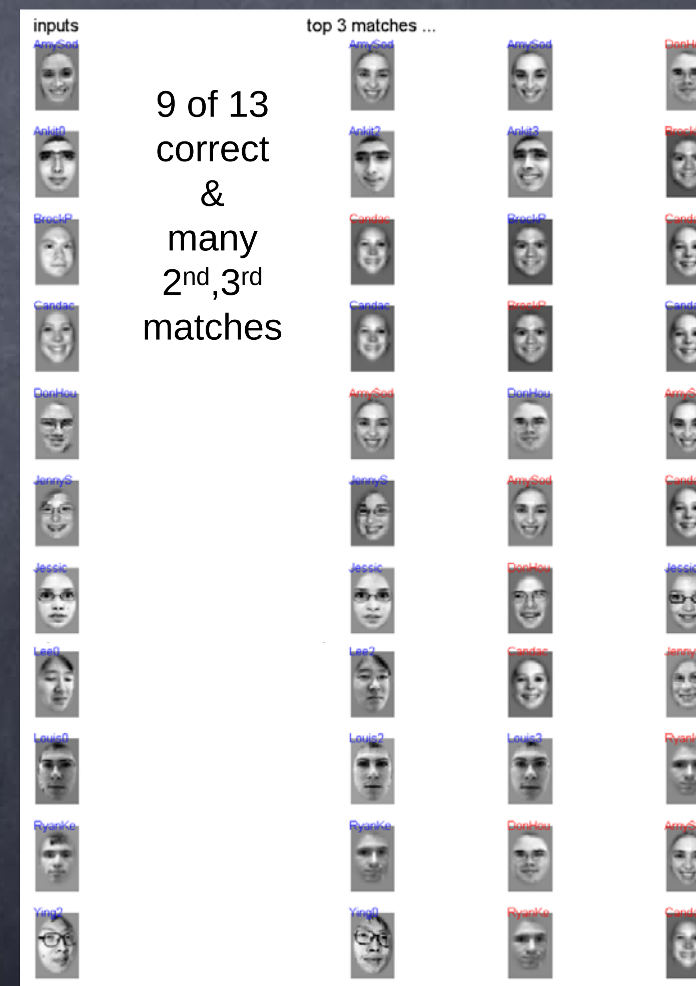


After Pre-Processing





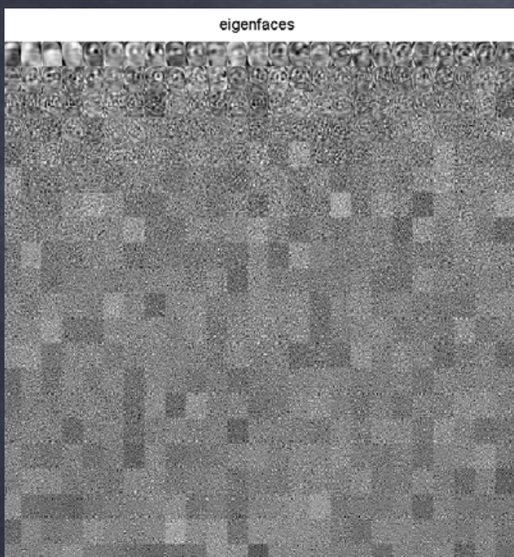
6 of 13 correct & few 2nd,3rd matches



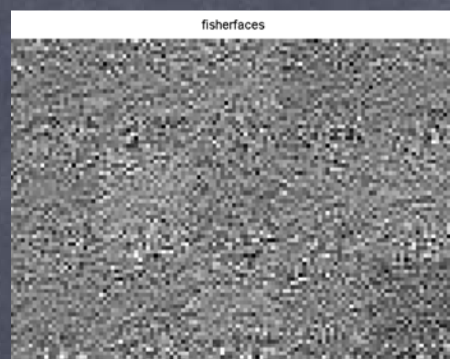
9 of 13 correct & many 2nd,3rd matches

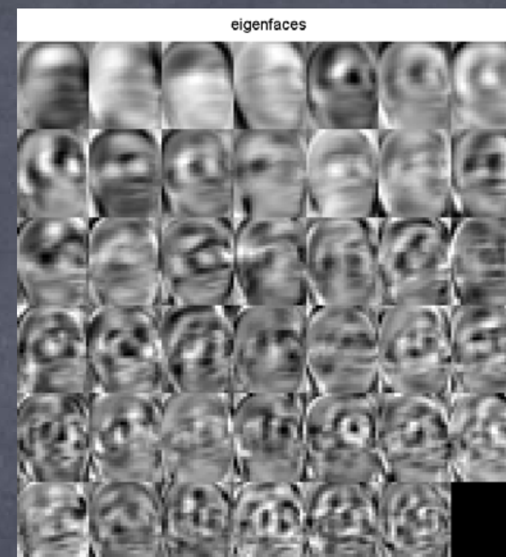
Fisherface Modification

- Impose a maximum on the number of PCA components to use in dimension reduction. Rationale: “face-space” doesn’t need more than Mp eigenvectors to be well-represented.

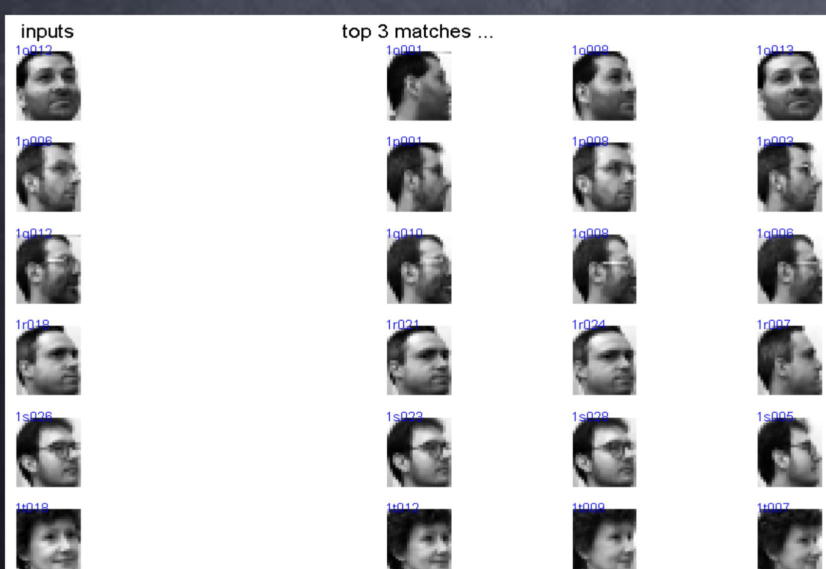


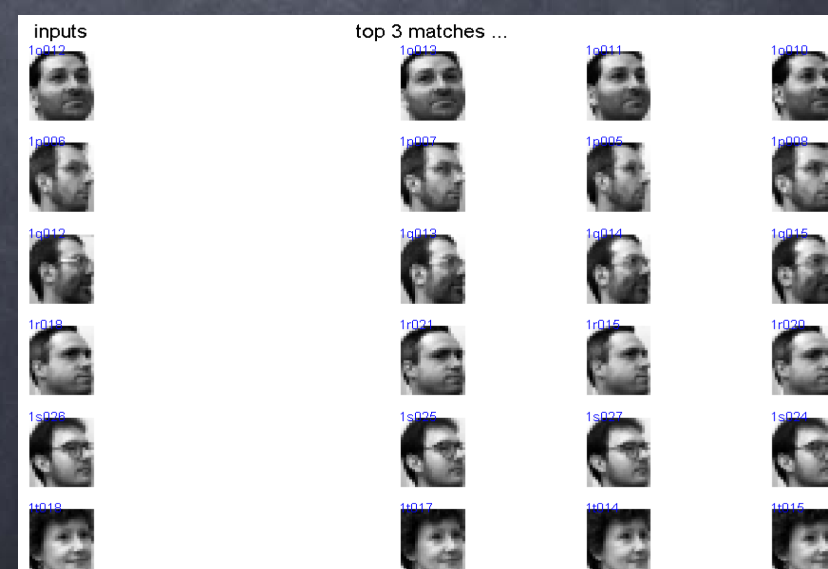
eigenface dimension
= n-c





eigenface dimension
= 41





Further Work

- Integrate with face detector.
- Incorporate time info in classifier.
- Try SVM, kernel, ICA, wavelet, plenoptic (light-field) approaches.
- Acquire & use CMU PIE and FERET databases and formal evaluation techniques.

References

- M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, 1991.
- M. Turk and A. Pentland, "Face recognition using eigenfaces," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1991, pp. 586-591.
- P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, July 1997.