# Eigenfaces and Deformations

Oleg Makhnin
New Mexico Tech
Dept. of Mathematics

December 2011

Deformation technique

Results applied to "mean face"

Results applied to a subspace

Deformation technique

Results applied to ”mean face”

Results applied to a subspace

Deformation technique

Results applied to "mean face"

Results applied to a subspace

Deformation technique

Results applied to ”mean face”

Results applied to a subspace

(Smooth) transformation of one image into another: should take care of

- • incorrect centering

(Smooth) transformation of one image into another: should take care of

- • incorrect centering
- • incorrect zoom

(Smooth) transformation of one image into another: should take care of

- incorrect centering
- incorrect zoom
- tilts of head

(Smooth) transformation of one image into another:
should take care of

- incorrect centering
- incorrect zoom
- tilts of head
- facial expressions

(Smooth) transformation of one image into another:
should take care of

- incorrect centering
- incorrect zoom
- tilts of head
- facial expressions

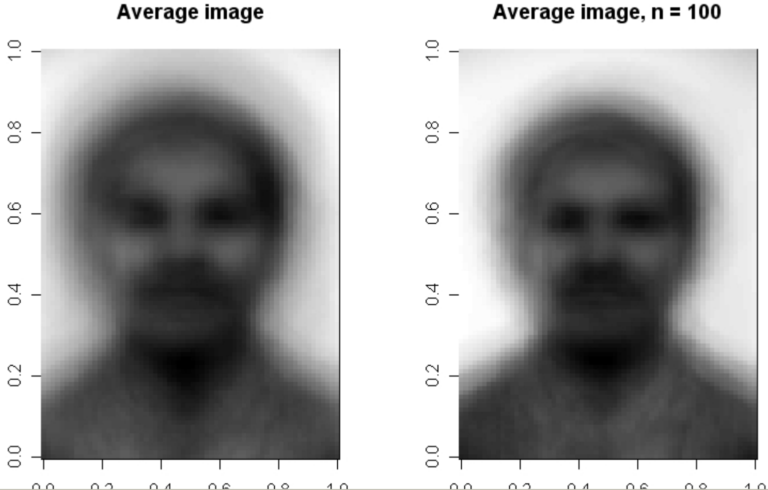
(Smooth) transformation of one image into another:
should take care of

- • incorrect centering
- • incorrect zoom
- • tilts of head
- • facial expressions

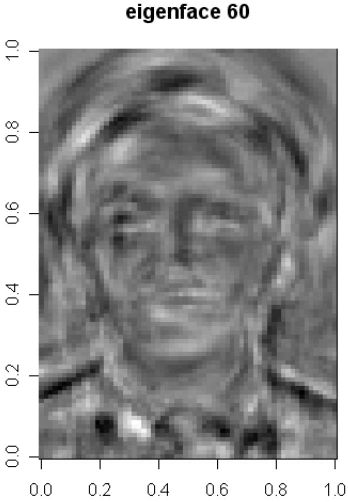
- • more generally, allows to develop common features of many faces, even though they vary in proportions

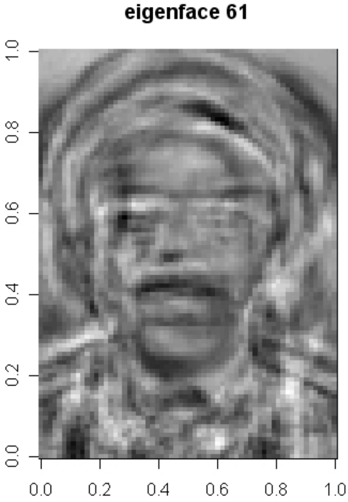
# Motivation: Mean face and some eigenfaces without deformations

– very blurry and/or busy



# Eigenfaces 60 and 61 (out of 161).



# How does the deformation work?

Deformations are not new, but usually they require extensive feature-matching work (10-20 reference points need to be marked by hand)

Vector field V; V is defined over a fairly coarse grid, then interpolated for the whole image grid.

To calculate the deformed image: start from a pixel of deformed image and trace back to the pixels of original image. Allow for partial overlap (take weighted averages of original pixels that match)


1
2
3
4
5
6

# Estimation methods

Suppose we try to match an image $I$ to some reference image $R$. Denote the image I deformed by vector field V as $I(V)$.

Treat this as an optimization problem, with objective function $F$ to minimize, where

$$F(V) = k_1 \times \|I(V) - R\|^2 + k_2 \times \texttt{roughness}(V)$$

That is, find "the closest match" to reference image R, with the restriction that the deformation should not be too wild.
$k_1$ and $k_2$ are tuning coefficients.

Algorithm for optimization: a variant of stochastic hill-climbing, combined with genetic algorithms... (Very time-consuming!)

# An easier matching algorithm?

We could use correlation between images (or parts thereof) as a match criterion. Thus, the simplified match algorithm is as follows:

- Split the image into rectangular fragments

This may not be optimal, but at least may serve as a starting point for a more expensive algorithm.

# An easier matching algorithm?

We could use correlation between images (or parts thereof) as a match criterion. Thus, the simplified match algorithm is as follows:

- Split the image into rectangular fragments
- For each fragment, find the optimal shift $(\Delta x, \Delta y)$ that maximizes the correlation between that fragment and the corresponding part of the reference image

This may not be optimal, but at least may serve as a starting point for a more expensive algorithm.

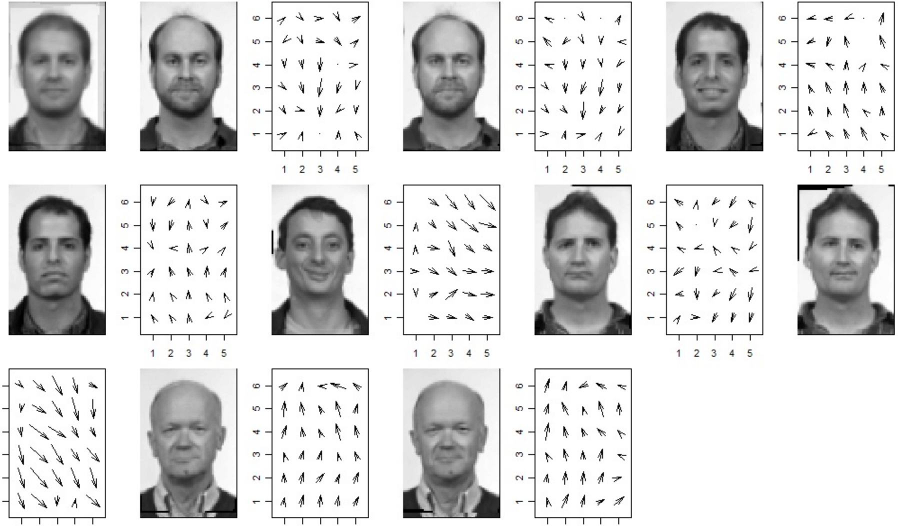
# An easier matching algorithm?

We could use correlation between images (or parts thereof) as a match criterion. Thus, the simplified match algorithm is as follows:

- Split the image into rectangular fragments
- For each fragment, find the optimal shift $(\Delta x, \Delta y)$ that maximizes the correlation between that fragment and the corresponding part of the reference image
- Smoothen out the results (to obtain a vector field $V$ that "makes sense")

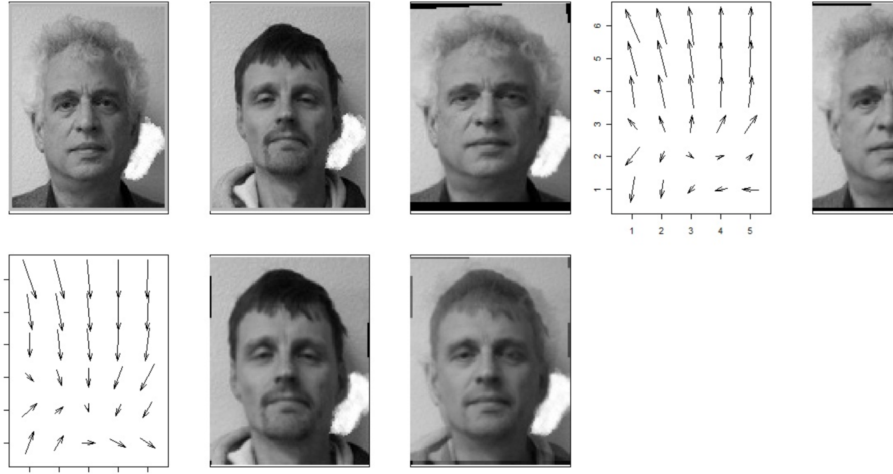
This may not be optimal, but at least may serve as a starting point for a more expensive algorithm.

## 10 images (FERET database), matching to a "mean face"

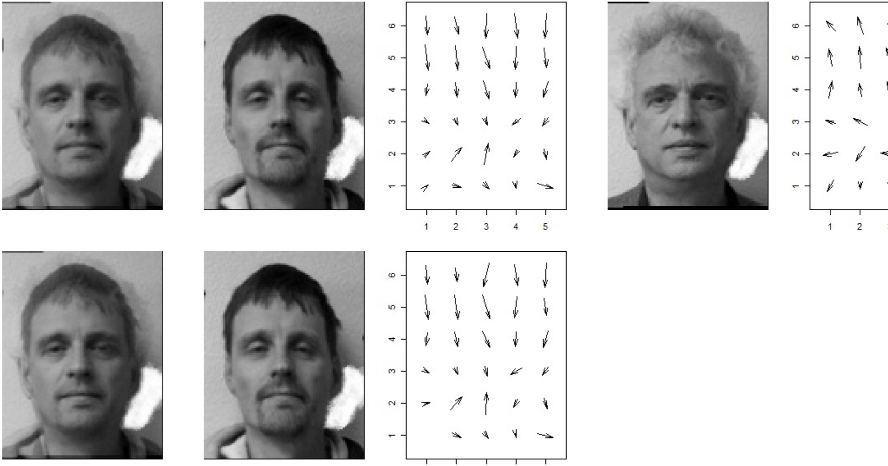


Evolution of the mean face:

## Another example: 2 people
## First, using the correlation-based technique



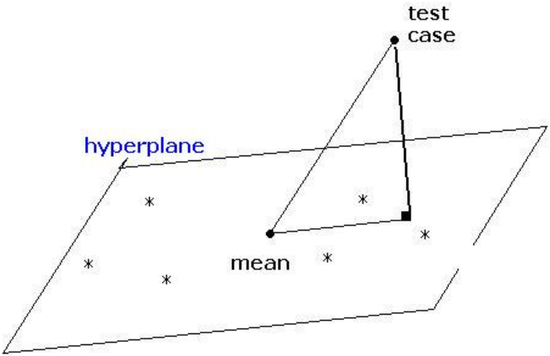
Then, using the full-scale optimization



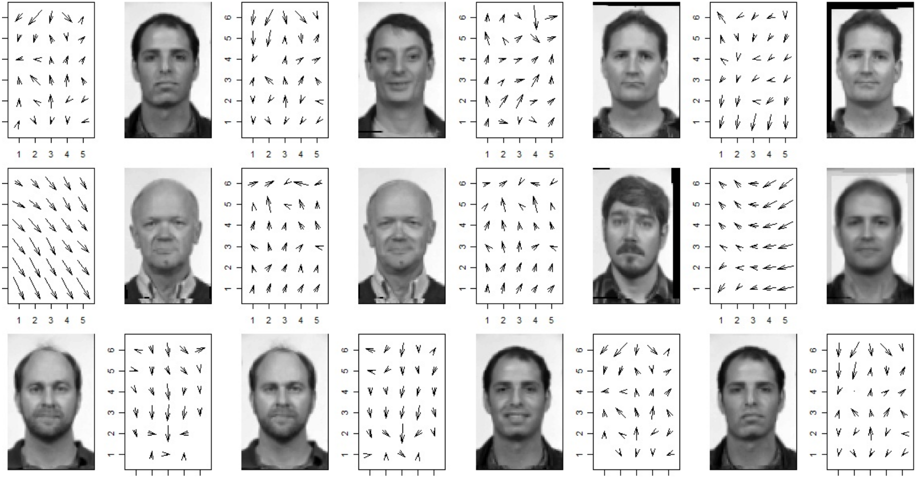
## Evolution of the mean face

# Improvement

Instead of matching the "mean face", match the low-dimensional subspace (defined by 1st few columns of U)



# Same faces using the subspace matching (dim = 3)



The basis of the subspace (dim = 3)

# To do

- • Acceleration: current matching algorithm is awfully slow

# To do

- Acceleration: current matching algorithm is awfully slow
- Might help: nested (coarse-to-fine) search

# To do

- • Acceleration: current matching algorithm is awfully slow
- • Might help: nested (coarse-to-fine) search
- • Ignoring features outside the face oval ("mask"). Possible auto-search for masks (e.g. occlusions)

# To do

- • Acceleration: current matching algorithm is awfully slow
- • Might help: nested (coarse-to-fine) search
- • Ignoring features outside the face oval ("mask"). Possible auto-search for masks (e.g. occlusions)
- • How to localize features?

# Conclusions

Image Processing is hard!!!

THANK YOU!

see

www.nmt.edu/~olegm/talks/EigDef.pdf