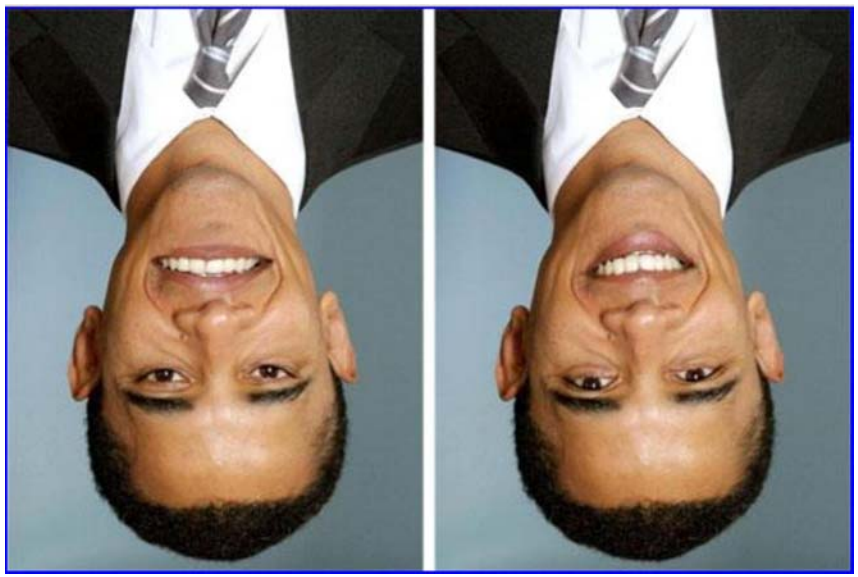
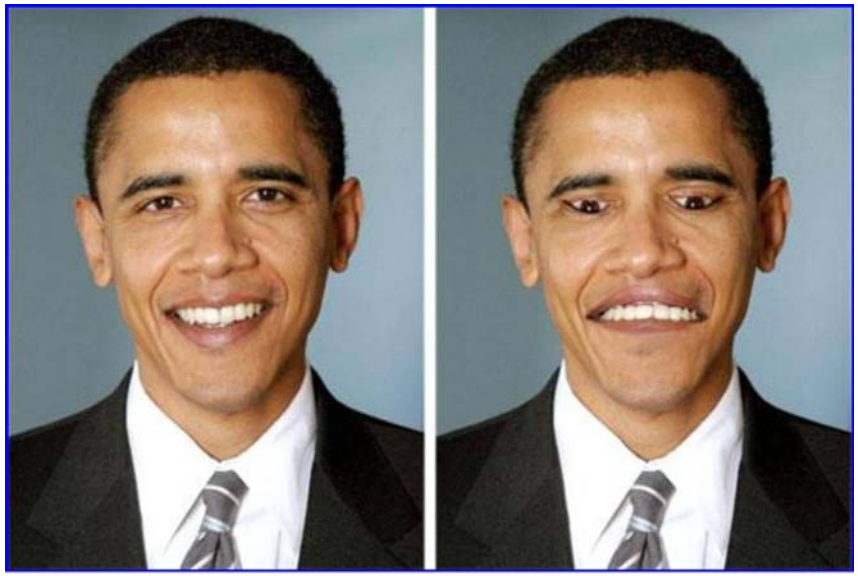
Lecture 14

Recognition



http://www.moillusions.com/



Human recognition system can be deceived by inverted faces as in previous slide!

Recognition problems

What versus Where

- Object Identification vs. Object Detection

What kind of object is it?

- Object class recognition (categorization)

Other recognition problems

- Activity recognition
- Texture recognition
- Shape recognition

Lots of applications! (can you think of some?)

Object Detection

Example: Face Detection (Lecture 9)

(Rowley, Baluja & Kanade, 1998)

Example: Skin Detection (Lecture 10)

(Jones & Rehg, 1999)

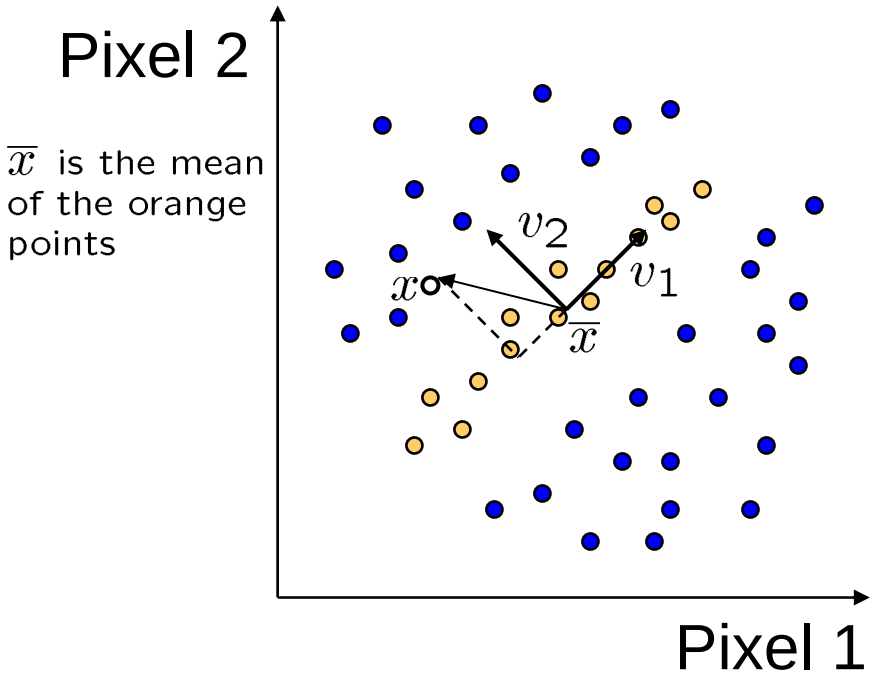
Object Identification

Whose face is it?

We will explore one approach, based on statistics of pixel values, called eigenfaces

Starting point: Treat N x M image as a vector in NM-dimensional space (form vector by collapsing rows from top to bottom into one long vector)

Linear subspaces



convert **x** into $\mathbf{v_1}$, $\mathbf{v_2}$ coordinates

$$\mathbf{x} \rightarrow ((\mathbf{x} - \overline{x}) \cdot \mathbf{v_1}, (\mathbf{x} - \overline{x}) \cdot \mathbf{v_2})$$

What does the $\mathbf{v_2}$ coordinate measure?

- distance to line
- use it for classification—near 0 for orange pts

What does the $\mathbf{v_1}$ coordinate measure?

- position along line
- use it to specify which orange point it is

Classification can be expensive

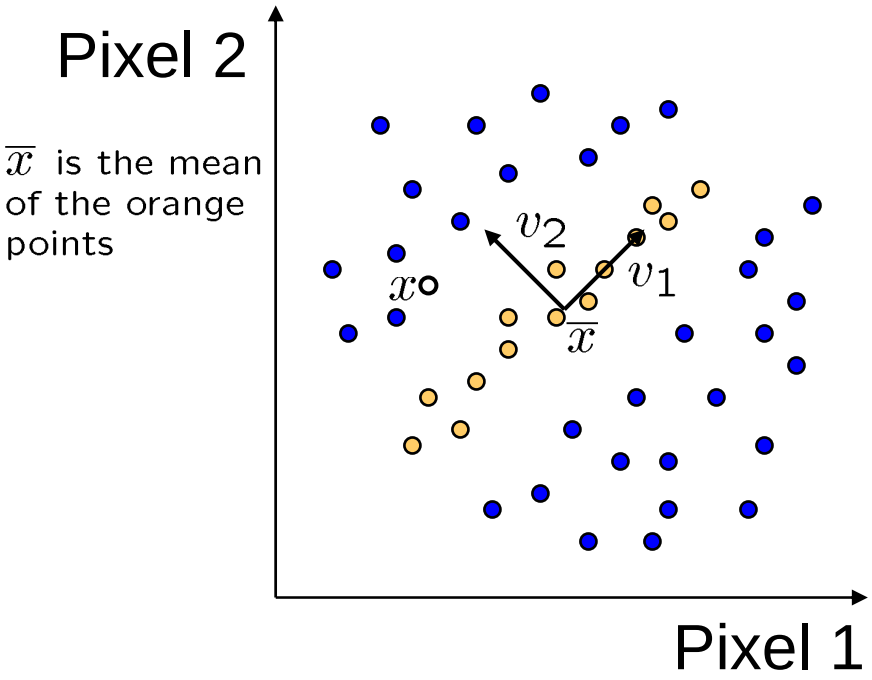
- Big search prob (e.g., nearest neighbors) or store large PDF's

Suppose the data points are arranged as above

- Idea—fit a line, classifier measures distance to line

Selected slides adapted from Steve Seitz, Linda Shapiro

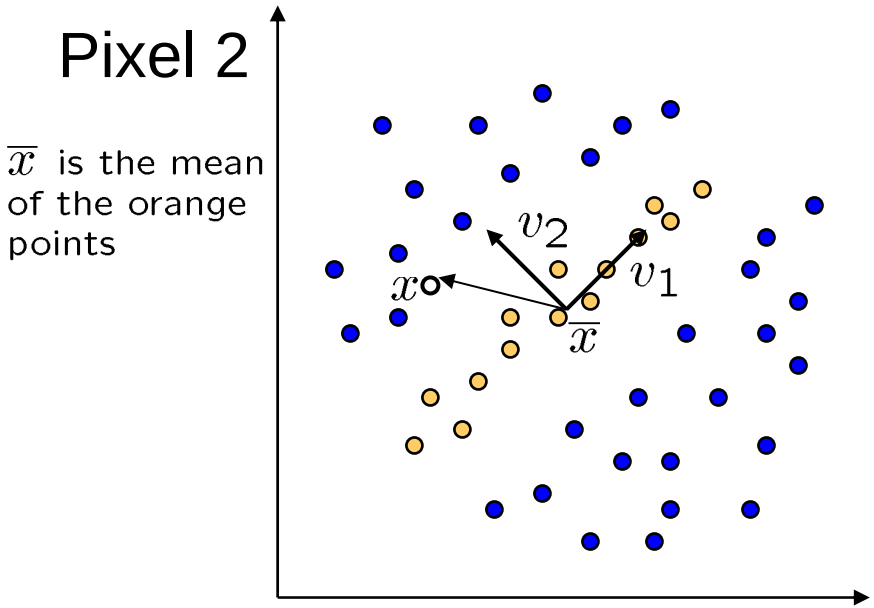
Dimensionality reduction



Dimensionality reduction

- We can represent the orange points with *only* their $\mathbf{v}_1$ coordinates
  - since $\mathbf{v}_2$ coordinates are all essentially 0
- This makes it much cheaper to store and compare points
- A bigger deal for higher dimensional problems (like images!)

Linear subspaces



Consider the variation along a direction **v** among all of the orange points:

$$var(\mathbf{v}) = \sum_{\text{orange point } \mathbf{x}} \|(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}} \cdot \mathbf{v}\|^2$$

What unit vector **v** minimizes *var*?

$$\mathbf{v}_2 = min_{\mathbf{v}} \{var(\mathbf{v})\}$$

What unit vector **v** maximizes *var*?

$$\mathbf{v}_1 = max_{\mathbf{v}} \{var(\mathbf{v})\}$$

$$\begin{aligned} var(\mathbf{v}) &= \sum_{\mathbf{x}} \|(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}} \cdot \mathbf{v}\|^2 \\ &= \sum_{\mathbf{x}} \mathbf{v}^{\mathbf{T}} (\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}} \mathbf{v} \\ &= \mathbf{v}^{\mathbf{T}} \left[\sum_{\mathbf{x}} (\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}} \right] \mathbf{v} \\ &= \mathbf{v}^{\mathbf{T}} \mathbf{A} \mathbf{v} \quad \text{where } \mathbf{A} = \sum_{\mathbf{x}} (\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^{\mathbf{T}} \end{aligned}$$

A = Covariance matrix of data points (if divided by no. of points)

Solution: $\mathbf{v}_1$ is eigenvector of **A** with *largest* eigenvalue
$\mathbf{v}_2$ is eigenvector of **A** with *smallest* eigenvalue

Principal component analysis

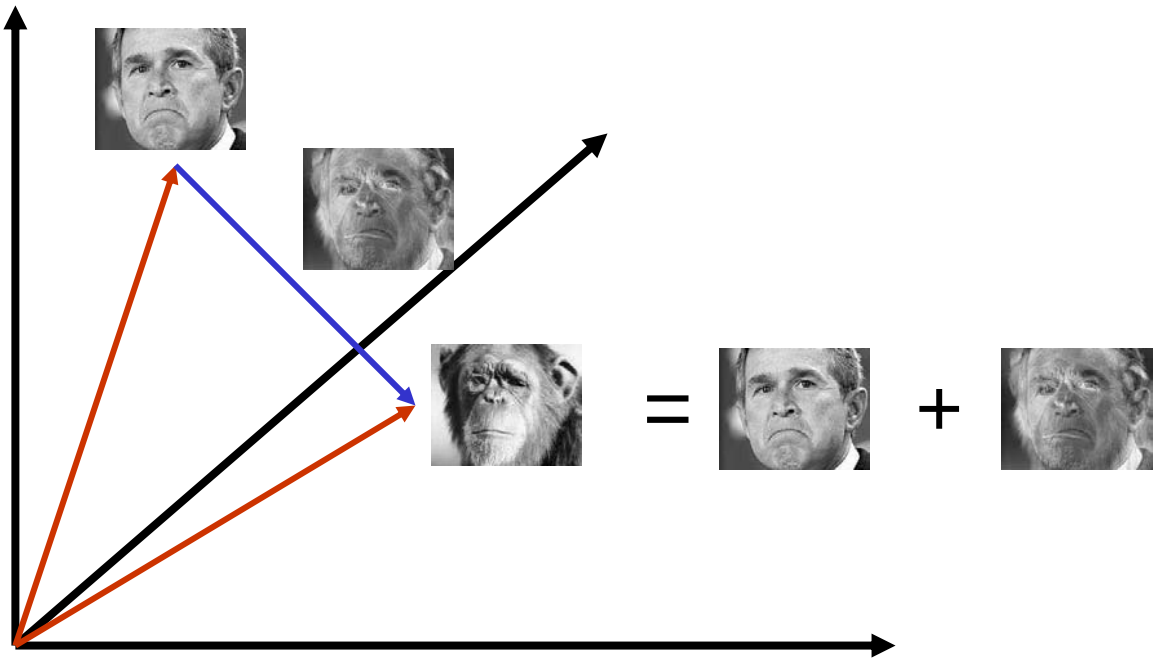
Suppose each data point is N-dimensional

- Same procedure applies:

$$\begin{aligned} var(\mathbf{v}) &= \sum_{\mathbf{x}} \|(\mathbf{x}-\overline{\mathbf{x}})^{\mathrm{T}} \cdot \mathbf{v}\| \\ &= \mathbf{v}^{\mathrm{T}}\mathbf{A}\mathbf{v} \quad \text{where } \mathbf{A} = \sum_{\mathbf{x}} (\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathrm{T}} \end{aligned}$$

- The eigenvectors of **A** define a new coordinate system
  - eigenvector with largest eigenvalue captures the most variation among training vectors **x**
  - eigenvector with smallest eigenvalue has least variation
- We can compress the data by only using the top few eigenvectors
  - corresponds to choosing a “linear subspace”
    - » represent points on a line, plane, or “hyper-plane”
  - these eigenvectors are known as ***principal component vectors***
  - Procedure is known as ***Principal Component Analysis (PCA)***

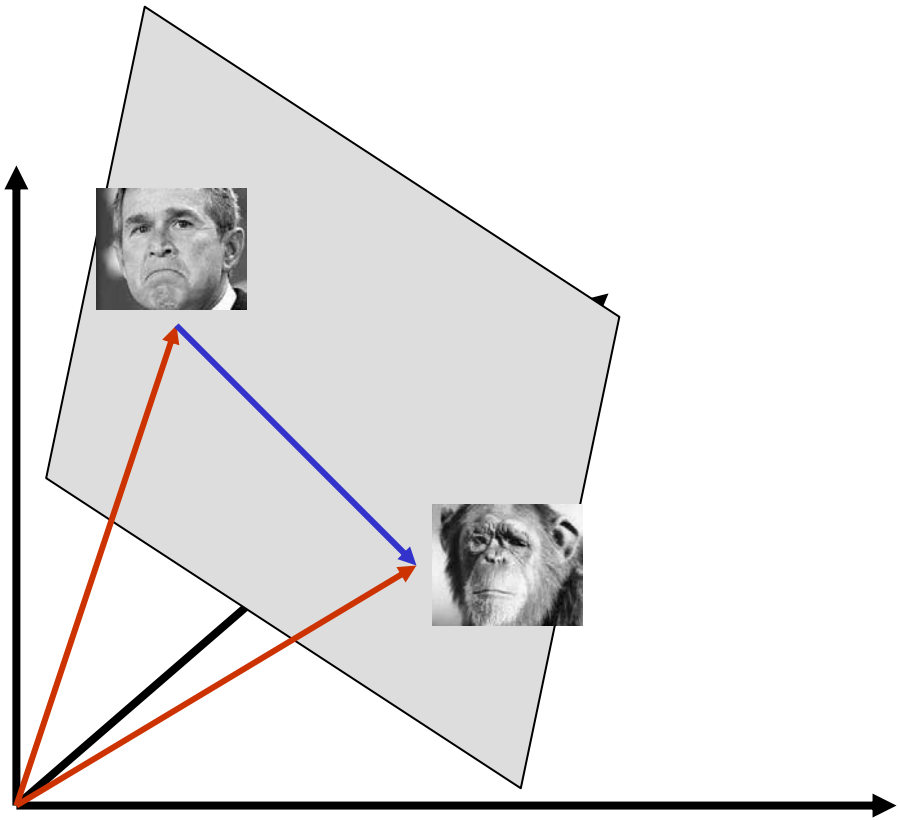
The space of faces



An image is a point in a high dimensional space

- An N x M image is a point in R^{NM}
- We can define vectors in this space as we did in the 2D case

Dimensionality reduction



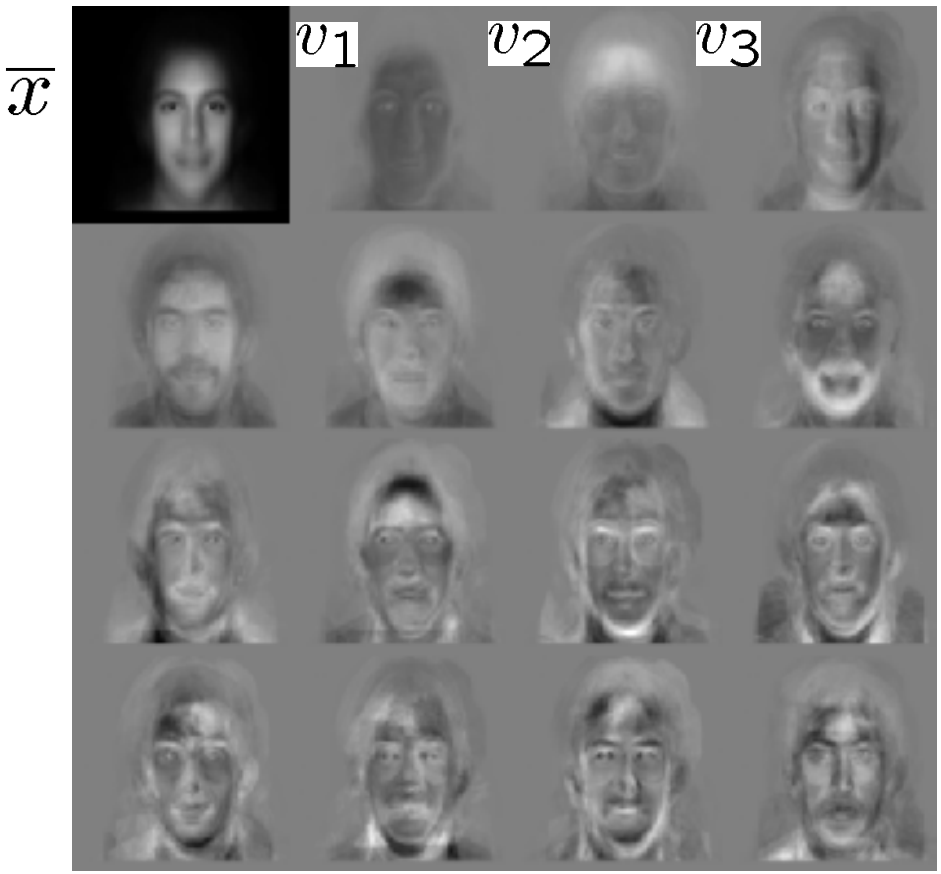
The space of all faces is a “subspace” of the space of all images

- Suppose it is K dimensional
- We can find the best subspace using PCA
- This is like fitting a “hyper-plane” to the set of faces
  - spanned by vectors $\mathbf{v_1}$, $\mathbf{v_2}$, ..., $\mathbf{v_K}$
  - any face $\mathbf{x} \approx \bar{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_k\mathbf{v_k}$

Eigenfaces

PCA extracts the eigenvectors of covariance matrix **A**

- Gives a set of vectors $\mathbf{v_1}$, $\mathbf{v_2}$, $\mathbf{v_3}$, ...
- Each one of these vectors is a direction in face space
  - what do these look like?



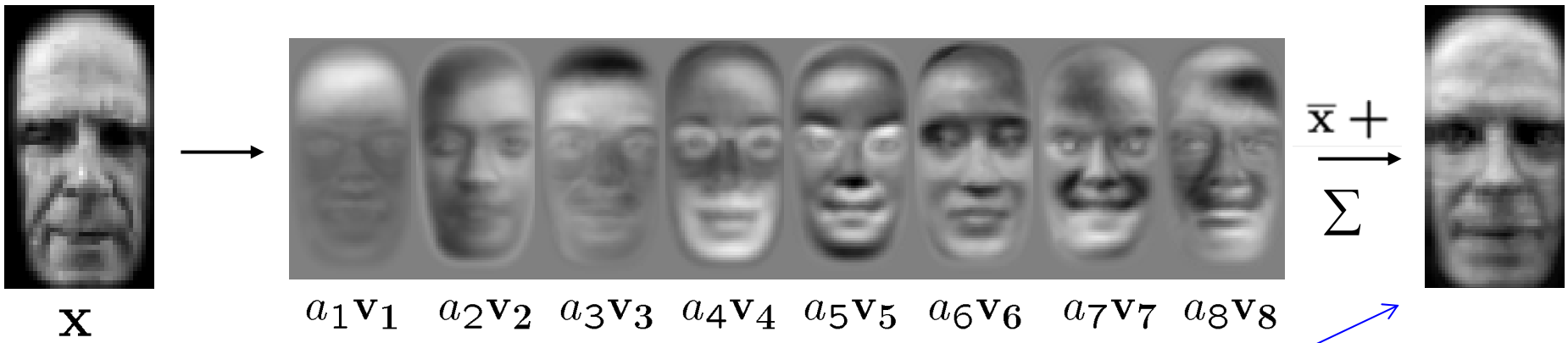
Projecting onto the eigenfaces

The eigenfaces $\mathbf{v_1}$, ..., $\mathbf{v_K}$ span the space of faces

- A face is converted to eigenface coordinates using dot products:

$$\mathbf{x} \rightarrow (\underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_1}}_{a_1}, \underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_2}}_{a_2}, \ldots, \underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_K}}_{a_K})$$

(Compressed representation of face,
K usually much smaller than NM)



Reconstructed face $\mathbf{x} \approx \overline{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_K\mathbf{v_K}$

Recognition with eigenfaces

Algorithm

1. Process the image database (set of images with labels)
   - Run PCA—compute eigenfaces
   - Calculate the K coefficients for each image
2. Given a new image (to be recognized) **x**, calculate K coefficients

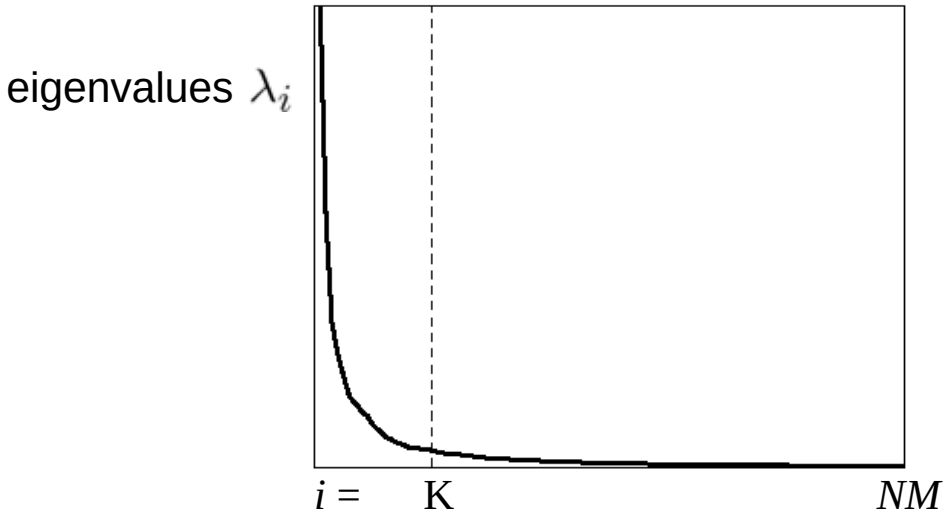
$$\mathbf{x} \rightarrow (a_1, a_2, \ldots, a_K)$$

3. Detect if x is a face

$$\|\mathbf{x} - (\overline{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_K\mathbf{v_K})\| < \text{threshold}$$

4. If it is a face, who is it?
   - Find closest labeled face in database
     - nearest-neighbor in K-dimensional space

Choosing the dimension K



How many eigenfaces to use?

Look at the decay of the eigenvalues

- the eigenvalue tells you the amount of variance “in the direction” of that eigenface
- ignore eigenfaces with low variance

You will get up close and personal with your own eigenfaces in Project 4!

Beyond Eigenfaces: Object Recognition by Parts

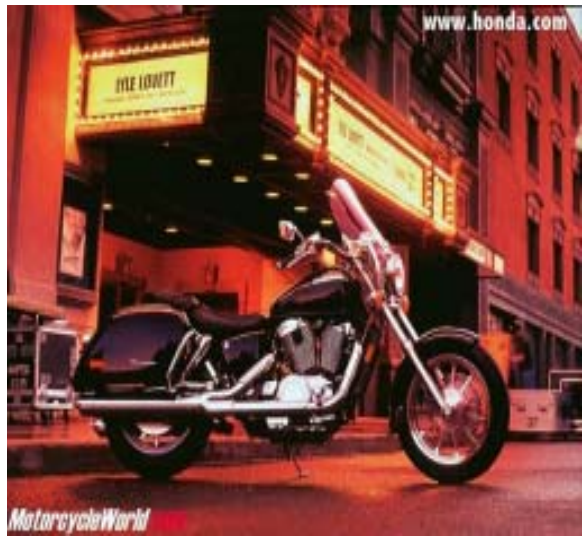
- Recall: Interest operators from Lecture 6 (Feature Detection)
  - E.g., Harris operator

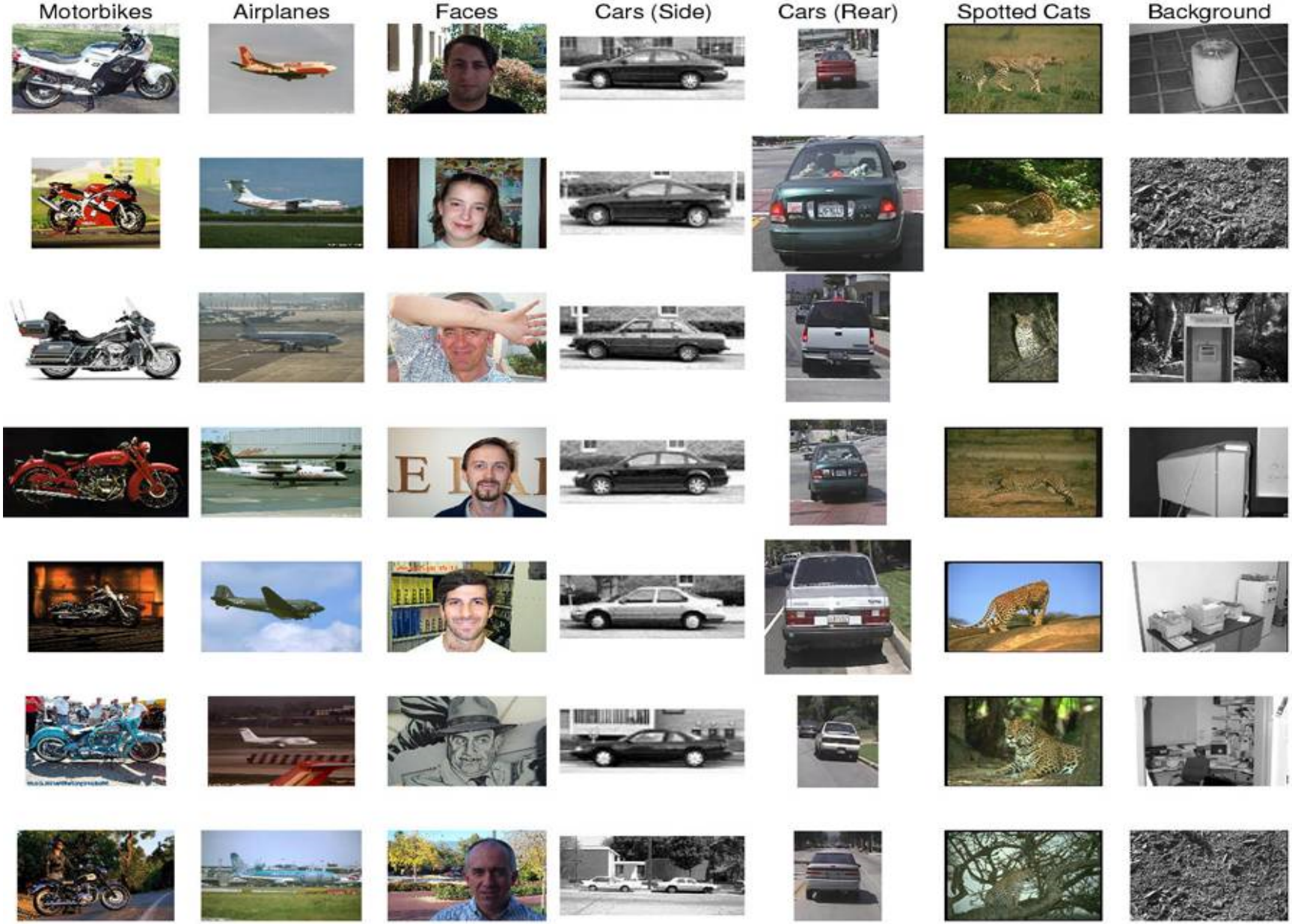
- Development of interest operators led to a new kind of recognition: Recognition by “parts”
  - allowed object class recognition/categorization

- We will look at one example: Object Class Recognition by Unsupervised Scale Invariant Learning by R. Fergus, P. Perona, and A. Zisserman, CVPR 2003.

Goal: Object Class Recognition (Categorization)

Recognize the category of object in input image




Motorbikes
Airplanes
Faces
Cars (Side)
Cars (Rear)
Spotted Cats
Background

Approach

An object is a constellation of parts (from Burl, Weber and Perona, 1998).

Parts are detected by an interest operator (Kadir and Brady operator).

Parts can be recognized by appearance (PCA).

Objects may vary greatly in scale.

Constellation of parts for a given object is learned from training images using EM algorithm.

Components

Model

- Generative Probabilistic Model including Location, Scale, and Appearance of Parts

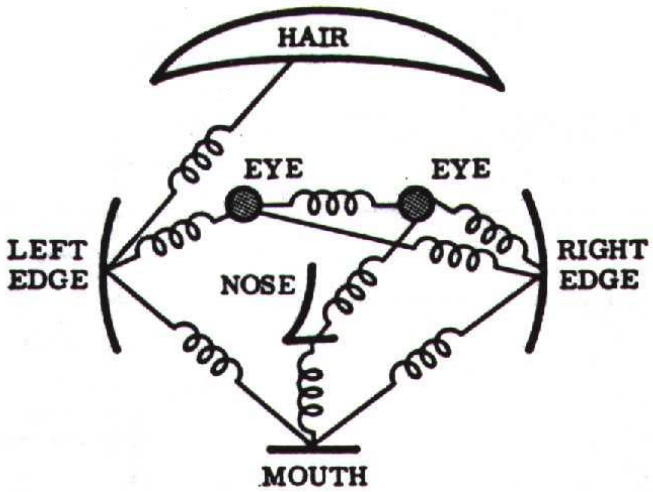
Learning

- Estimate Parameters Via EM Algorithm

Recognition

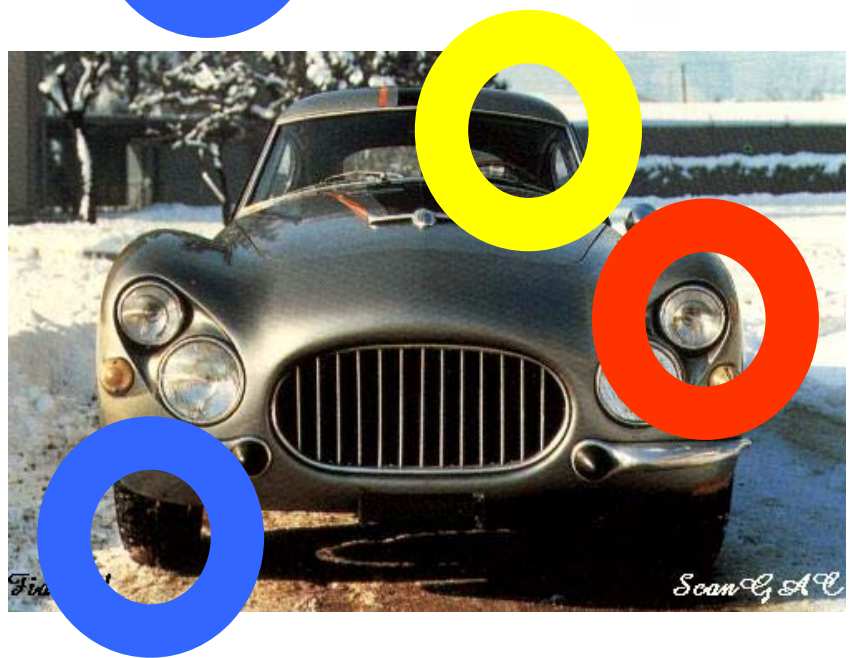
- Classify Image Using Learned Model and Threshold

Constellation Of Parts Model



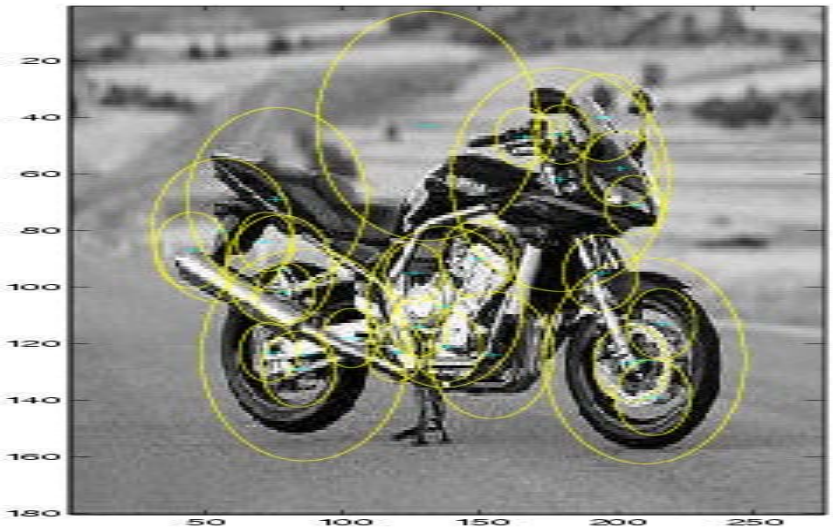
Fischler & Elschlager, 1973

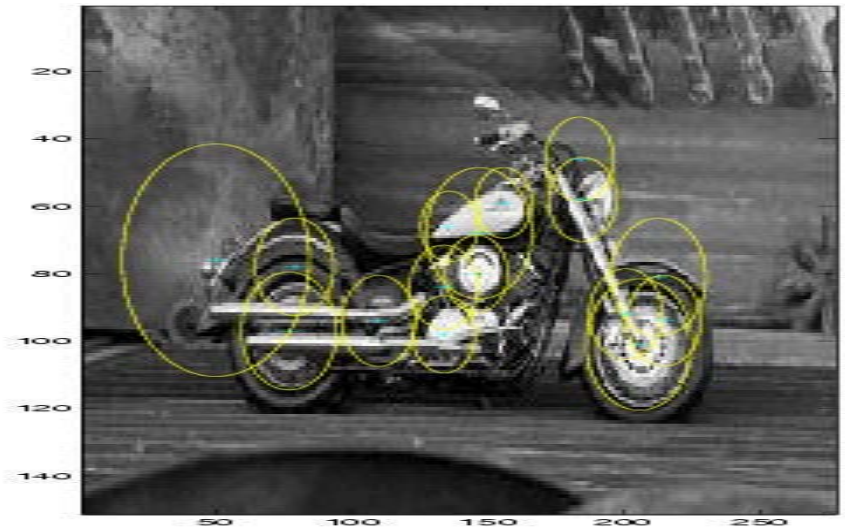
Yuille, 1991
Brunelli & Poggio, 1993
Lades, v.d. Malsburg et al. 1993
Cootes, Lanitis, Taylor et al. 1995
Amit & Geman, 1995, 1999
Perona et al. 1995, 1996, 1998, 1900



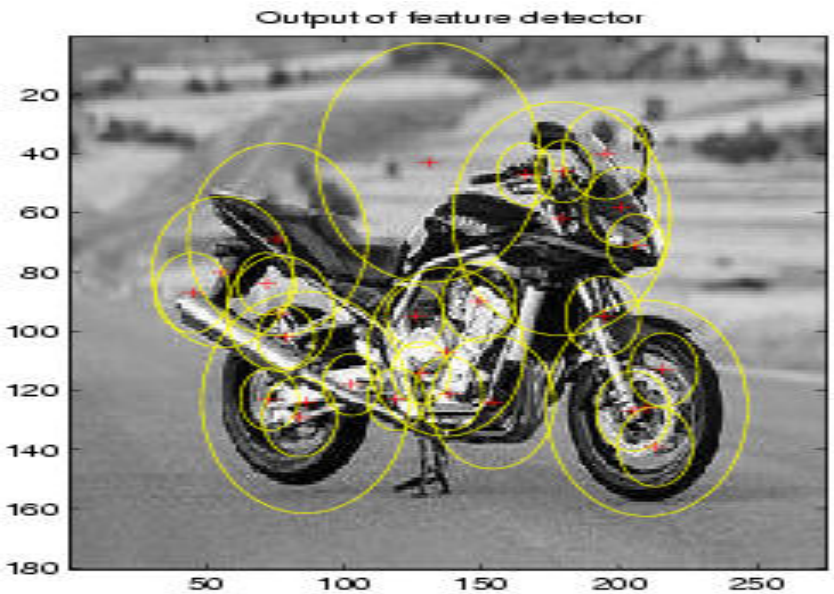
Parts Selected by Interest Operator

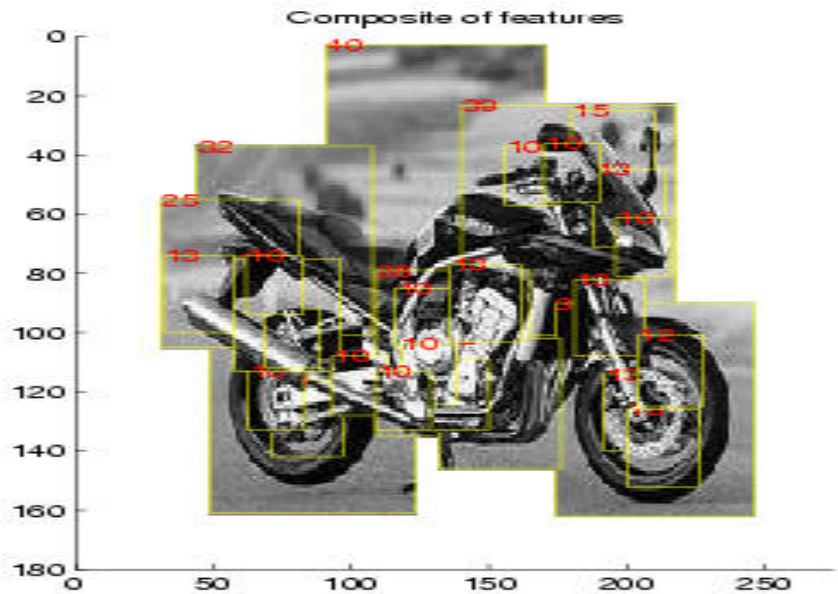
Kadir and Brady's Interest Operator: Finds Maxima in Entropy Over Scale and Location

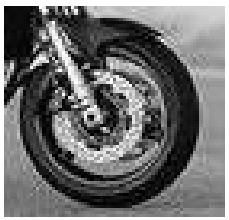




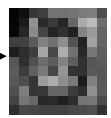
Representation of Appearance



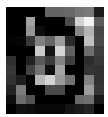




11x11 patch



Normalize



Projection onto PCA basis

c_1

c_2

c_{15}

121 dimensions too big, so use PCA to reduce to 10-15 eigenspace coefficients

Learning a Probabilistic Object Model

An object class is represented by a generative model with P parts (3-7 parts) and a set of parameters θ.

Suppose an image has N interesting features (up to 30) with locations X, scales S and appearances A.

Learn: *p*(Object | X, S, A) and p(No object | X, S, A)

Once models have been learned, use a likelihood ratio to determine if a new image contains an instance of each object class or not:

$$R = \frac{p(\text{Object}|\mathbf{X},\mathbf{S},\mathbf{A})}{p(\text{No object}|\mathbf{X},\mathbf{S},\mathbf{A})} > \text{Threshold ?}$$

Generative Probabilistic Model

$$
\begin{aligned}
R &= \frac{p(\text{Object}|\mathbf{X},\mathbf{S},\mathbf{A})}{p(\text{No object}|\mathbf{X},\mathbf{S},\mathbf{A})} \\
&= \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\text{Object})\,p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\text{No object})\,p(\text{No object})} \\
&\approx \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta)\,p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta_{bg})\,p(\text{No object})}
\end{aligned}
$$

$$
p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta) = \sum_{\mathbf{h}\in H} p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}|\,\theta) =
$$

$$
\sum_{\mathbf{h}\in H} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}_{Appearance}\underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}_{Shape}\underbrace{p(\mathbf{S}|\mathbf{h},\theta)}_{Rel.\ Scale}\underbrace{p(\mathbf{h}|\theta)}_{Other}
$$

R is the likelihood ratio.

θ is the maximum likelihood value of the parameters of the object and θ_{bg} of the background.

h is the *hypothesis* as to which P of the N features in the image are in the object, implemented as a vector of length P with values from 0 to N indicating which image feature corresponds to each object feature (0 = no part/occlusion)

H is the set of all hypotheses; Its size is $O(N^P)$.

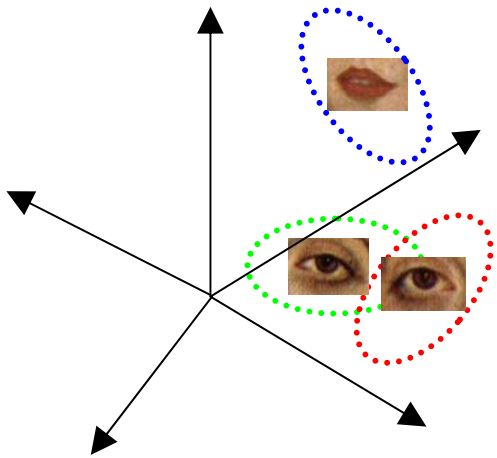
Appearance Model

$$p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta) = \sum_{\mathbf{h}\in H} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}_{Appearance}\underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}_{Shape}\underbrace{p(\mathbf{S}|\mathbf{h},\theta)}_{Rel.\ Scale}\underbrace{p(\mathbf{h}|\theta)}_{Other}$$

The appearance (A) of each part p has a Gaussian density with mean c_p and covariance V_P.

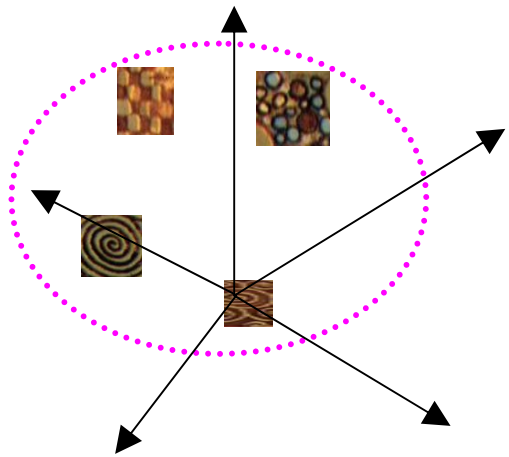
Background model has mean c_{bg} and covariance V_{bg}.

Gaussian Part Appearance PDF



Object

Guausian Appearance PDF



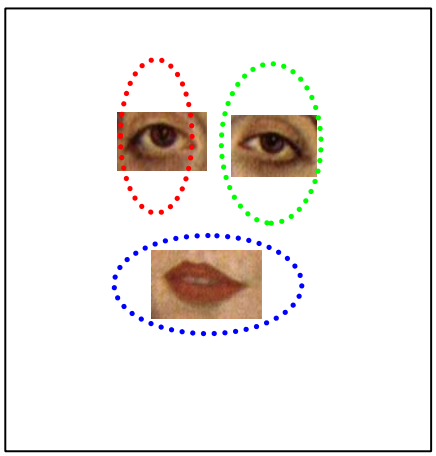
Background

Shape as Location

$$p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta) = \sum_{\mathbf{h}\in H} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}_{Appearance}\underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}_{Shape}\underbrace{p(\mathbf{S}|\mathbf{h},\theta)}_{Rel.\ Scale}\underbrace{p(\mathbf{h}|\theta)}_{Other}$$

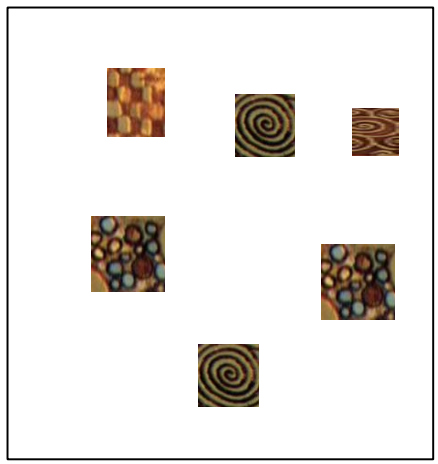
Object shape is represented by a joint Gaussian density of the locations (X) of features within a hypothesis transformed into a scale-invariant space.

Gaussian Shape PDF



Object

Uniform Shape PDF

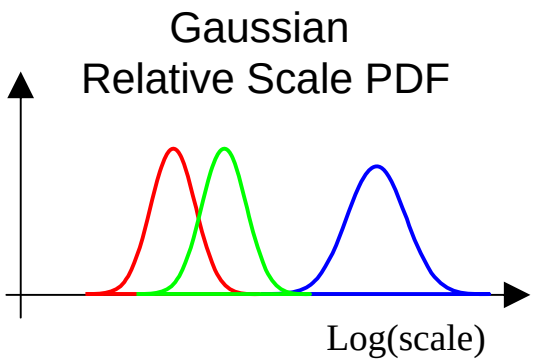


Background

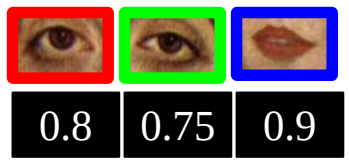
Scale Term

$$p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta) = \sum_{\mathbf{h}\in H} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}_{Appearance}\underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}_{Shape}\underbrace{p(\mathbf{S}|\mathbf{h},\theta)}_{Rel.\ Scale}\underbrace{p(\mathbf{h}|\theta)}_{Other}$$

The relative scale of each part is modeled by a Gaussian density with mean t_p and covariance U_p.



Prob. of detection



Other: Occlusion and Part Statistics

$$p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta) = \sum_{\mathbf{h}\in H} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}_{Appearance}\underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}_{Shape}\underbrace{p(\mathbf{S}|\mathbf{h},\theta)}_{Rel.\ Scale}\underbrace{p(\mathbf{h}|\theta)}_{Other}$$

Defined as product of 3 terms:

- First term: Poisson distribution (mean M) models the number of features detection.
- Second term: (constant) 1/(number of combinations of f_t features out of a total of N_t)
- Third term: probability table for all possible occlusion patterns.

Learning Parameters using EM Algorithm

Train All Model Parameters
Using EM:

- Optimize Parameters
- Optimize Assignments
- Repeat Until Convergence

$$\theta = \{\underbrace{\mu, \Sigma}_{\text{location}}, \underbrace{\mathbf{c}, V}_{\text{appearance}}, \underbrace{M, p(\mathbf{d}|\theta)}_{\text{occlusion}}, \underbrace{t, U}_{\text{scale}}\}$$

$$\hat{\theta}_{ML} = \arg\max_{\theta} \; p(\mathbf{X}, \mathbf{S}, \mathbf{A} | \theta)$$

Recognition

$$
\begin{aligned}
R &= \frac{p(\text{Object}|\mathbf{X},\mathbf{S},\mathbf{A})}{p(\text{No object}|\mathbf{X},\mathbf{S},\mathbf{A})} \\
&= \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\text{Object})\,p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\text{No object})\,p(\text{No object})} \\
&\approx \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta)\,p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta_{bg})\,p(\text{No object})} \quad > \text{Threshold ?}
\end{aligned}
$$

Results

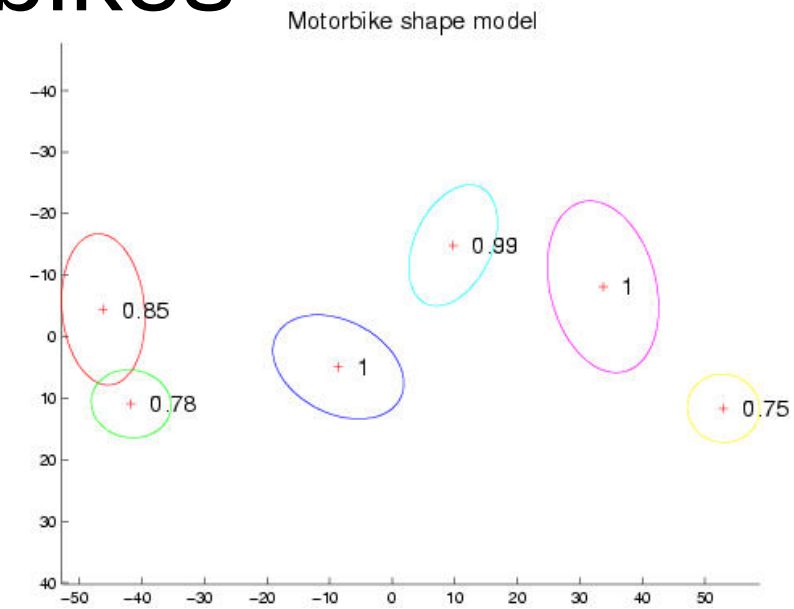
Initially tested on the Caltech-4 data set

- motorbikes
- faces
- airplanes
- cars

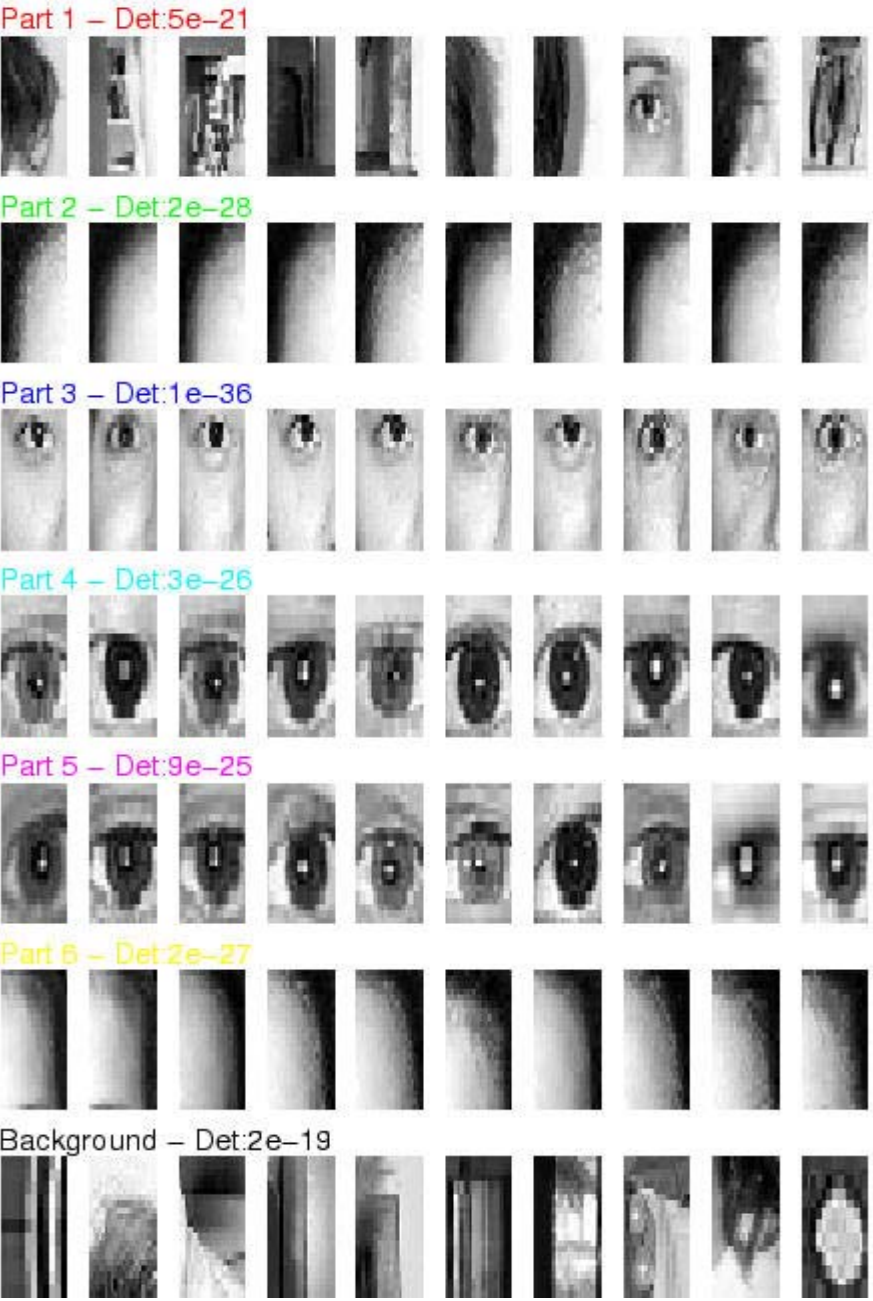
Now there is a much bigger data set: the Caltech-101
http://www.vision.caltech.edu/archive.html

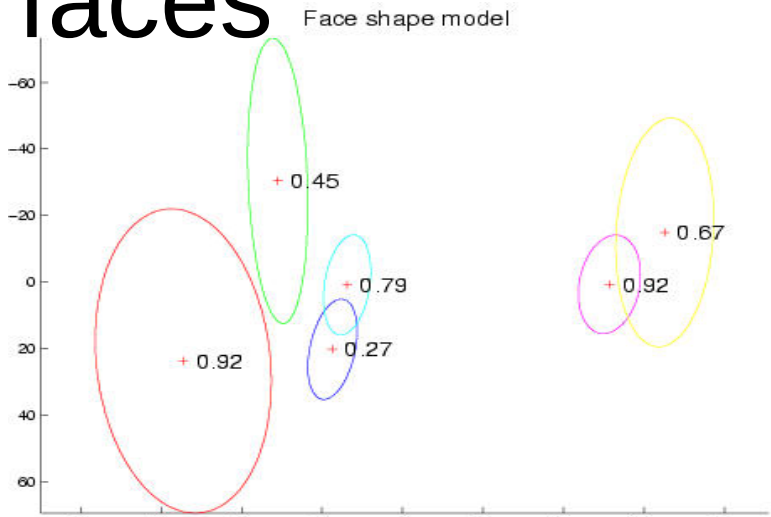
Motorbikes

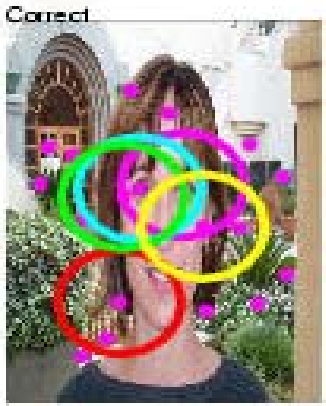

Part 1 – Det:5e–18
Part 2 – Det:8e–22
Part 3 – Det:6e–18
Part 4 – Det:1e–19
Part 5 – Det:3e–17
Part 6 – Det:4e–24
Background – Det:5e–19

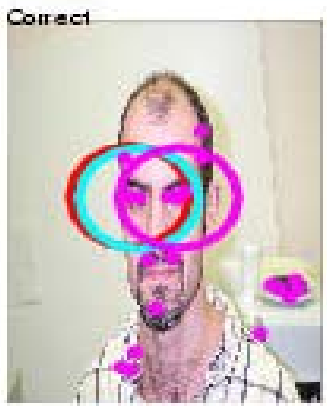

Motorbike shape model
0.85
0.78
0.99
1
1
0.75

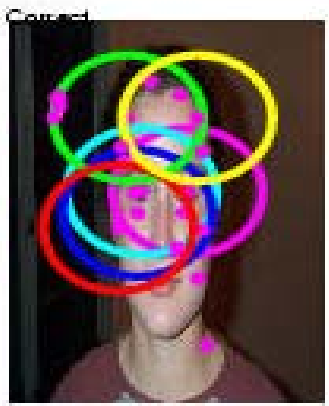
Frontal faces

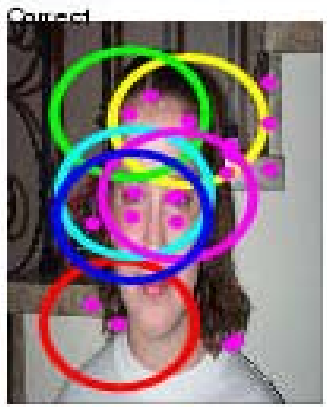


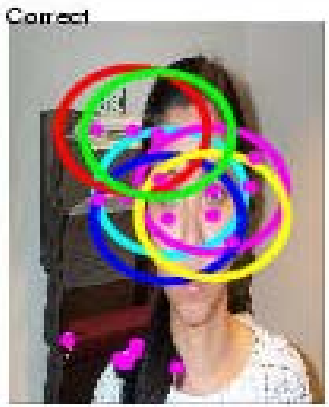




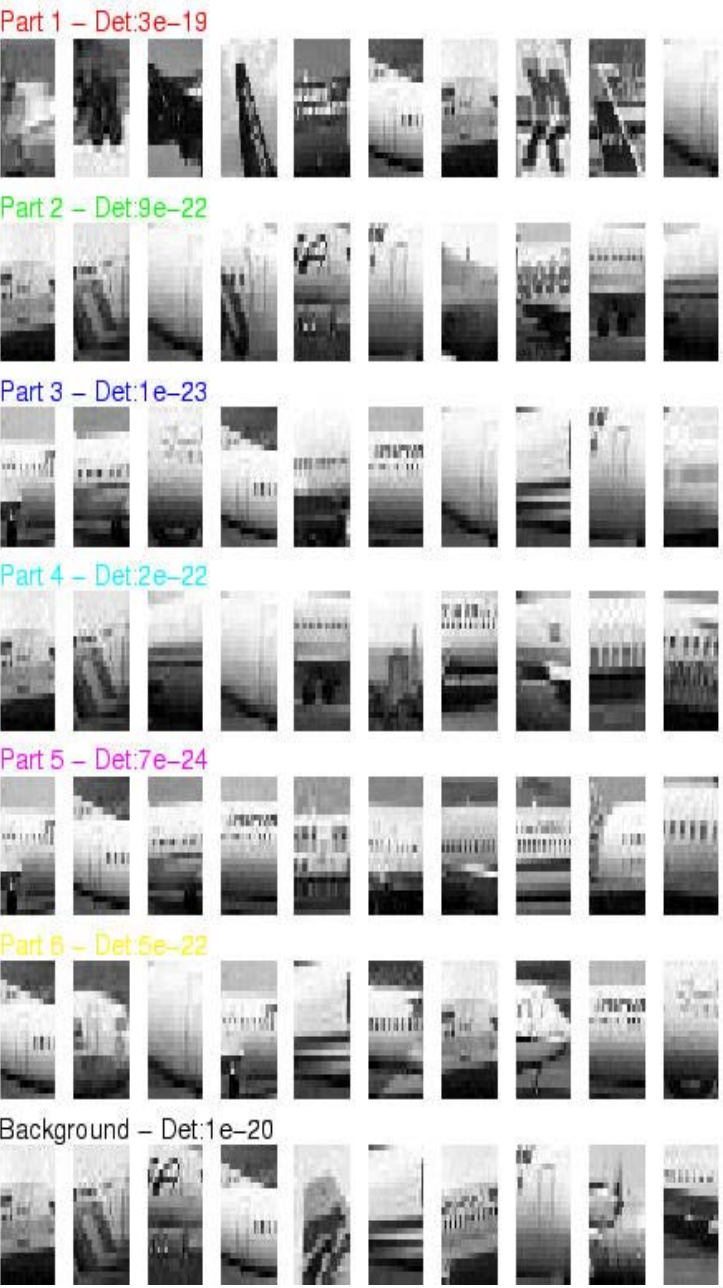


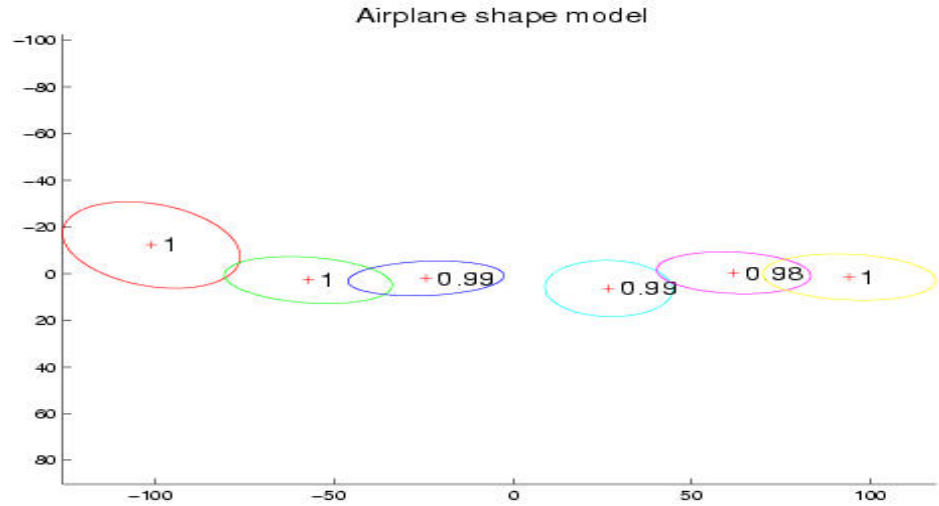


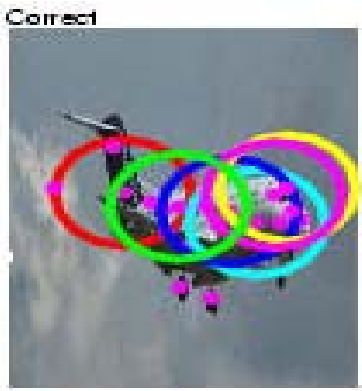


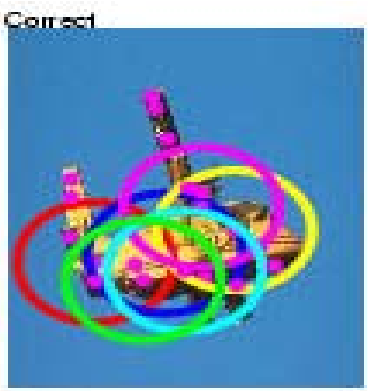


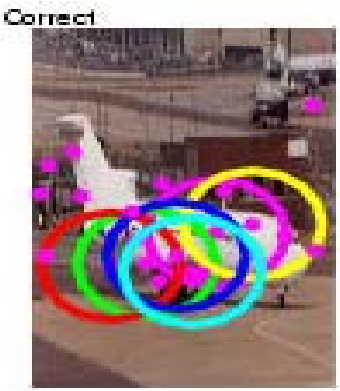
Airplanes


Part 1 – Det:3e–19
Part 2 – Det:9e–22
Part 3 – Det:1e–23
Part 4 – Det:2e–22
Part 5 – Det:7e–24
Part 6 – Det:5e–22
Background – Det:1e–20

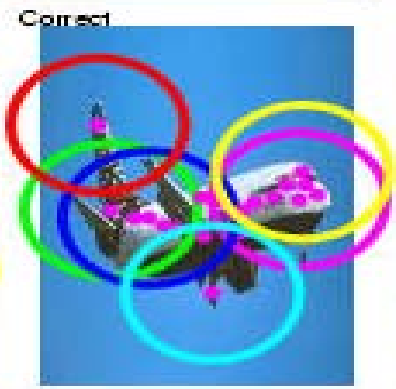

Airplane shape model
1
1
0.99
0.99
0.98
1

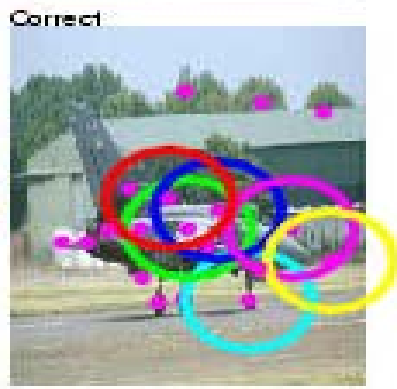

Correct


Correct

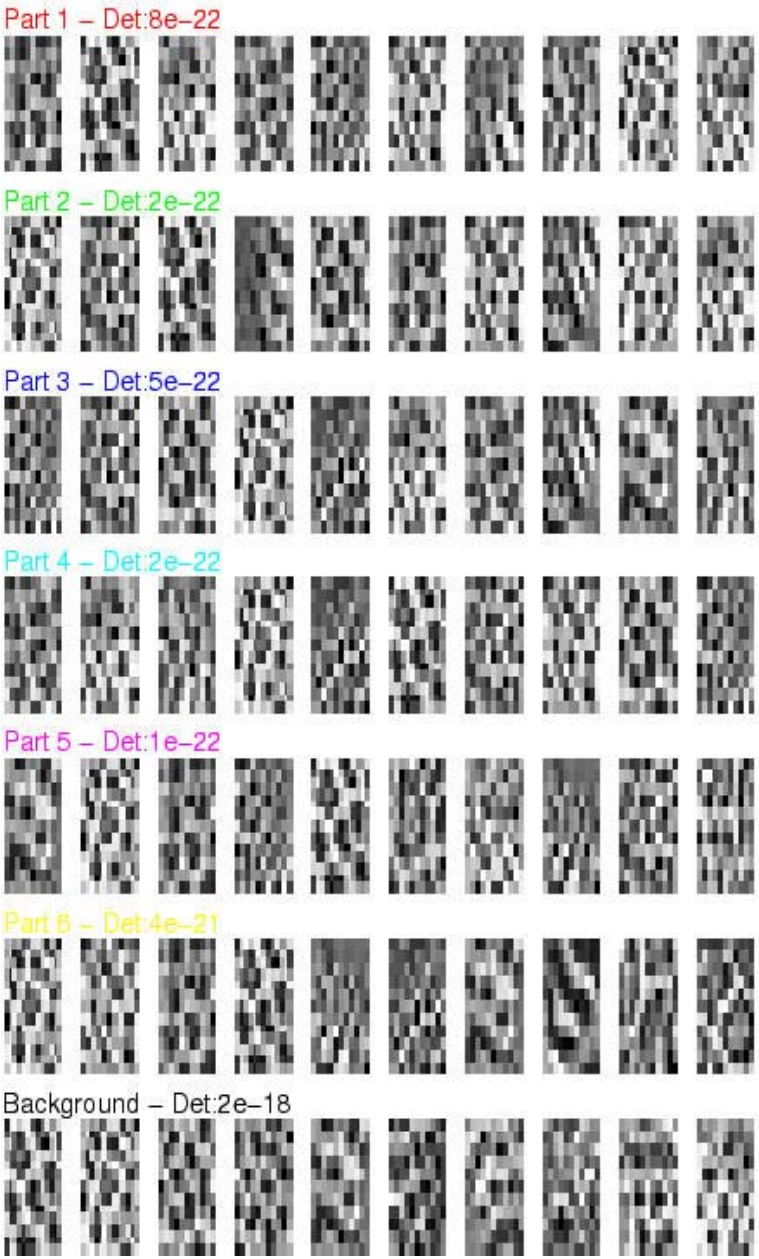

Correct

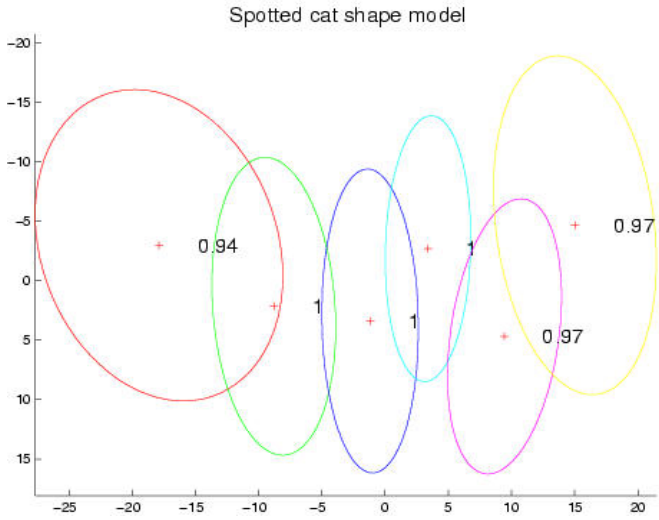

INCORRECT

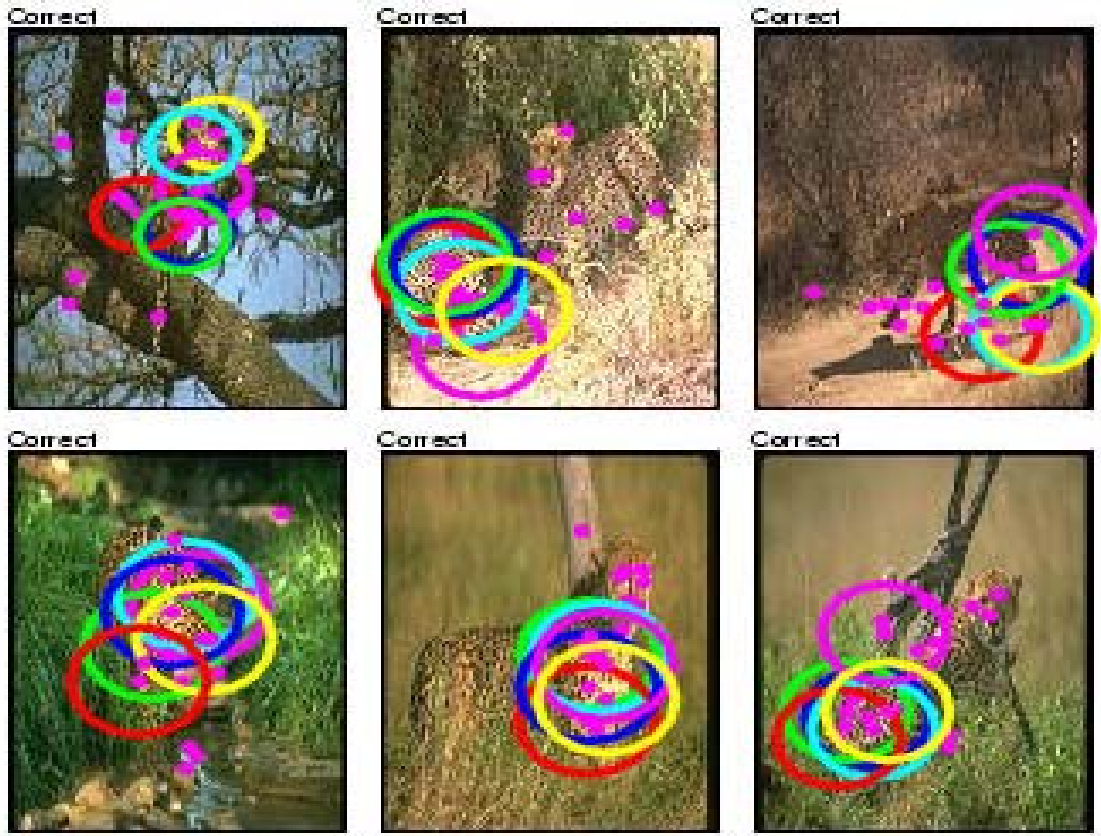

Correct


Correct

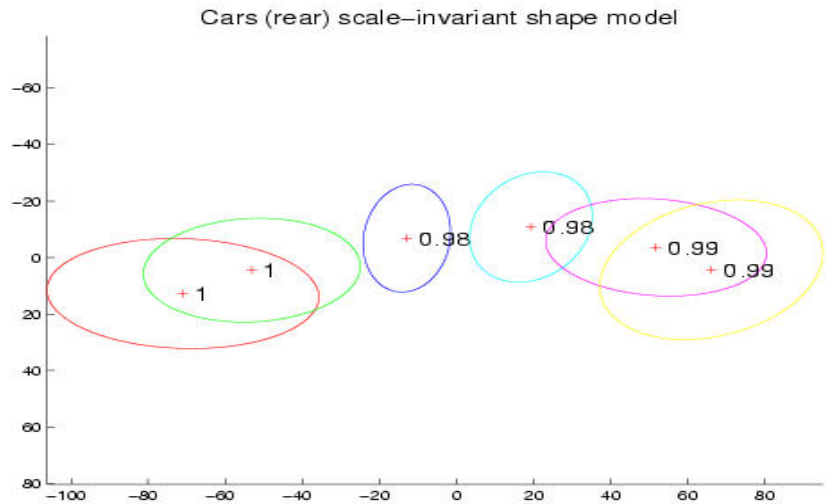
Spotted Cats

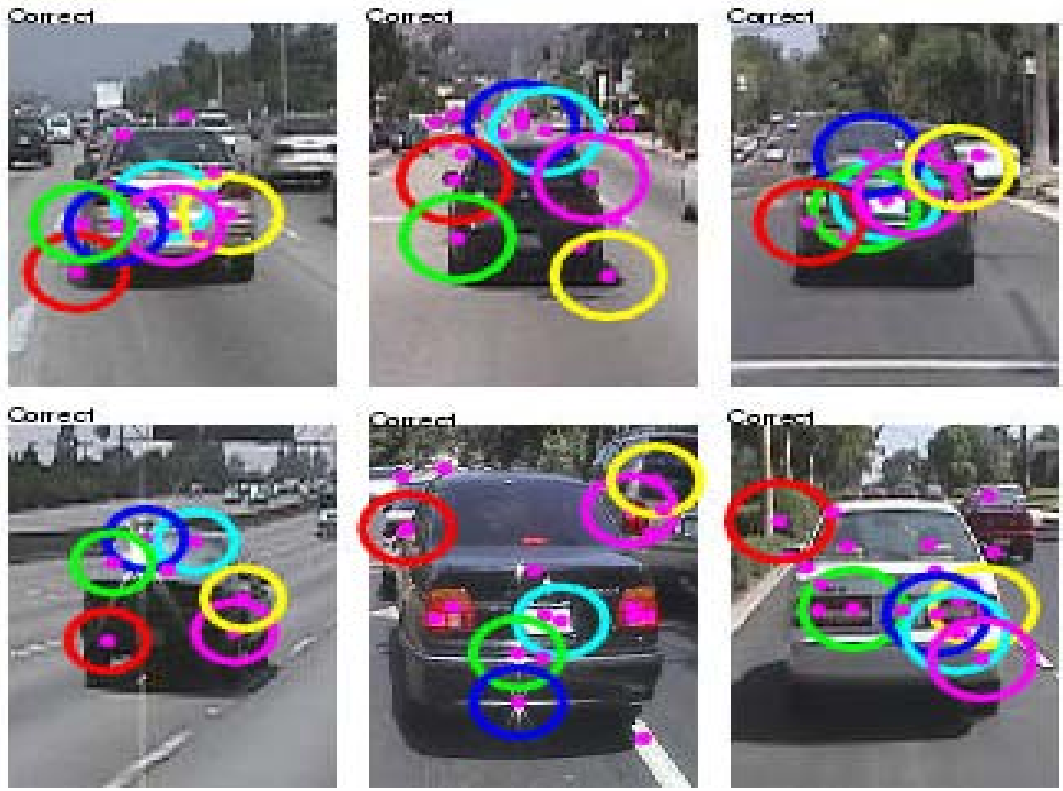




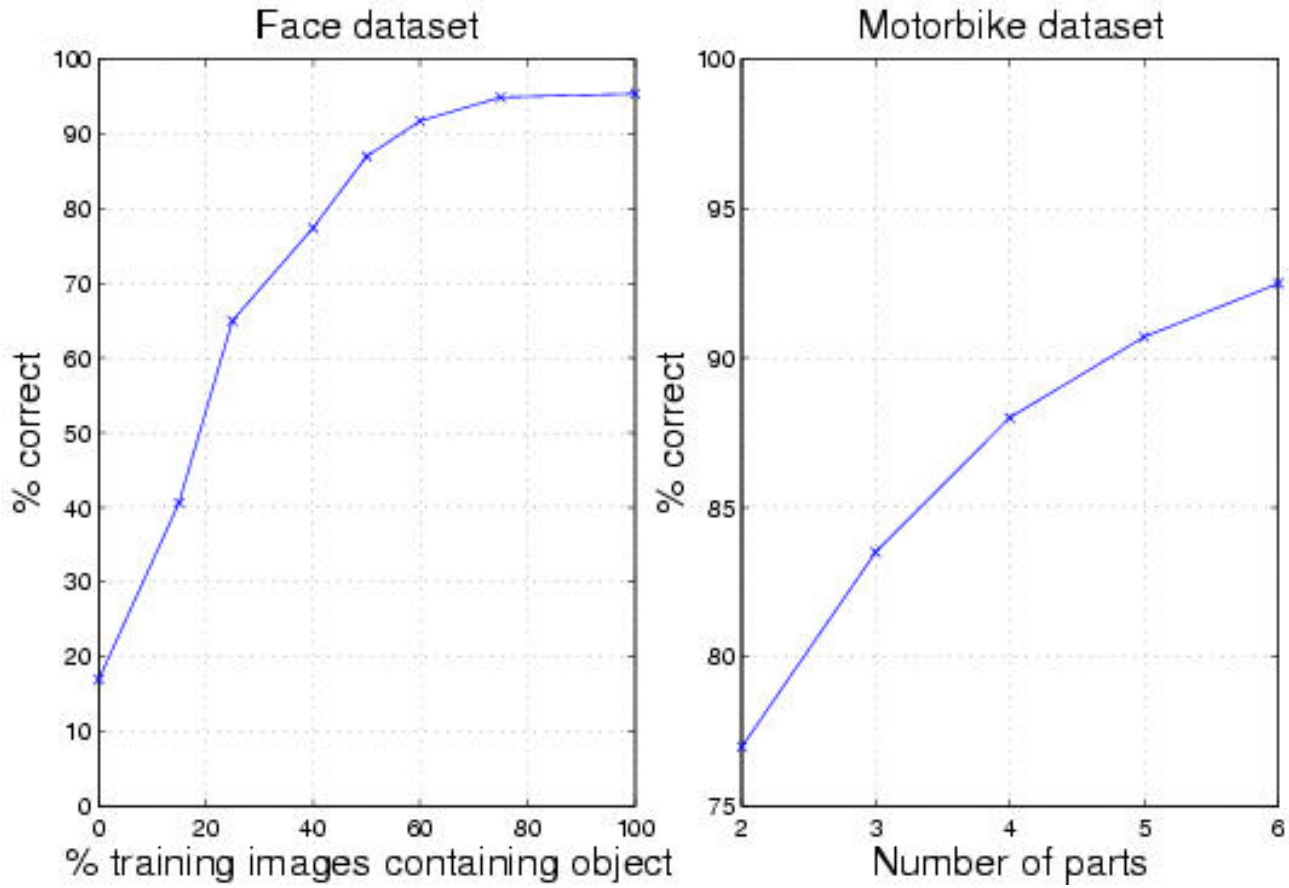


Scale-Invariant cars





Robustness of Algorithm



Accuracy

Initial Pre-Scaled Experiments

Dataset	Ours	Others	Ref.
Motorbikes	92.5	84	[17]
Faces	96.4	94	[19]
Airplanes	90.2	68	[17]
Cars(Side)	88.5	79	[1]

Scale-Invariant Learning and Recognition

Dataset	Total size of dataset	Object size range (pixels)	Pre-scaled performance	Unscaled performance
Motorbikes	800	200-480	95.0	93.3
Airplanes	800	200-500	94.0	93.0
Cars (Rear)	800	100-550	84.8	90.3

Object recognition

This is just the tip of the iceberg

- We've talked about using PCA-based features
- Many other features can be used:
  - edges
  - color
  - motion
  - object size
  - SIFT etc.

Classical object recognition techniques use line segments and recover 3D information as well

- Given an image and a database of CAD 3D models, determine which model(s) appears in that image
- Often recover 3D pose of the object as well
- Works well in industrial applications, not so well in unstructured environments

Recognition is a very active research area right now

Next Time: Motion

Things to do:

- Project 3 due tomorrow!
- Project 4 assigned today
- Read Chap. 9