Introduction to Computer Vision

Michael J. Black
Oct. 2009

Lecture 10:
Images as vectors.
Appearance-based models.

News

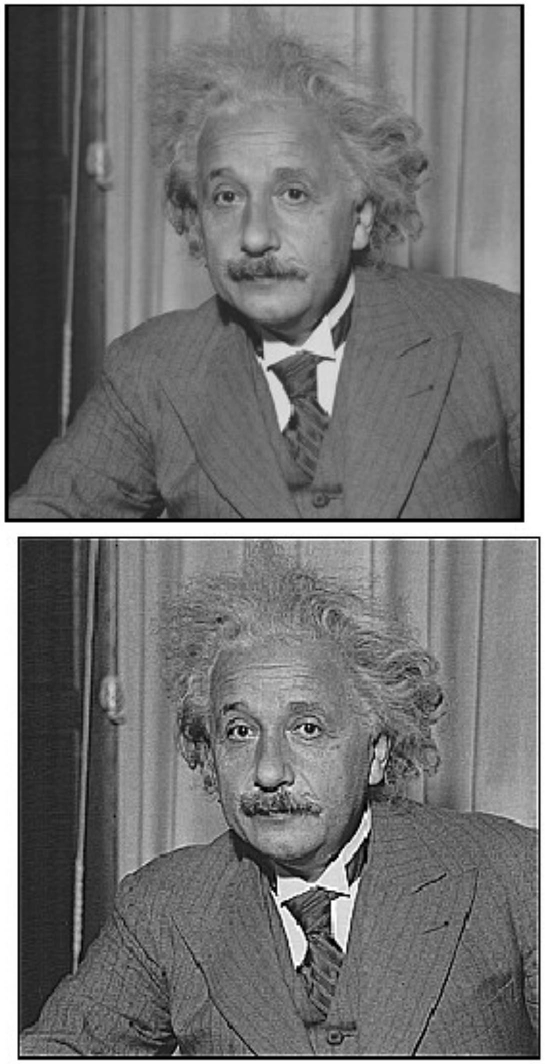
- Assignment 1 parts 3&4 – extension.
  - Due tomorrow, Tuesday, 10/6 at 11am.

Goals

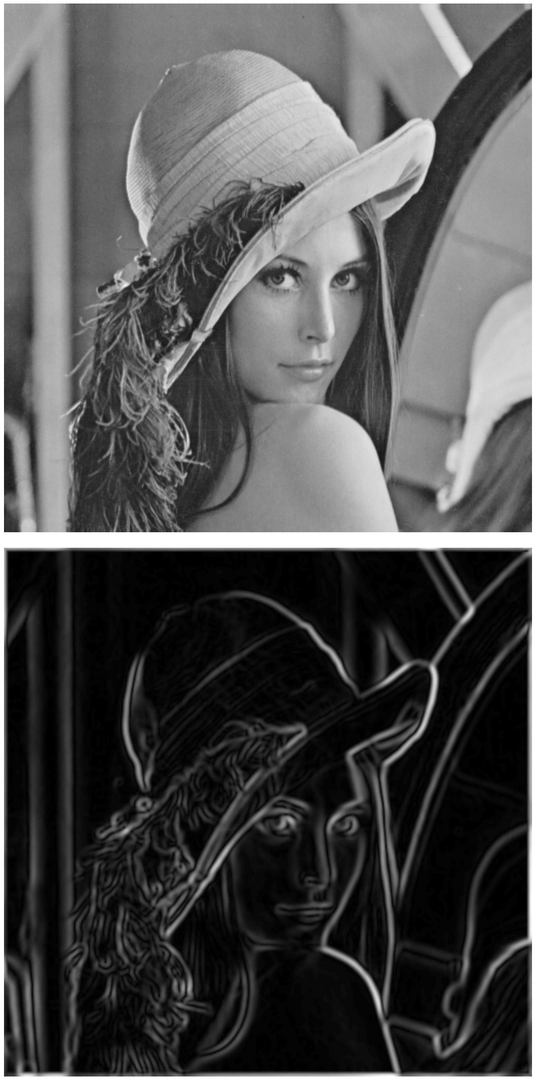
- Images as vectors in a high dimensional space

- Wed/Fri: Covariance, eigenvalues, features, principal component analysis.
  - Prep for next homework

Image Filtering



Smoothing and sharpening



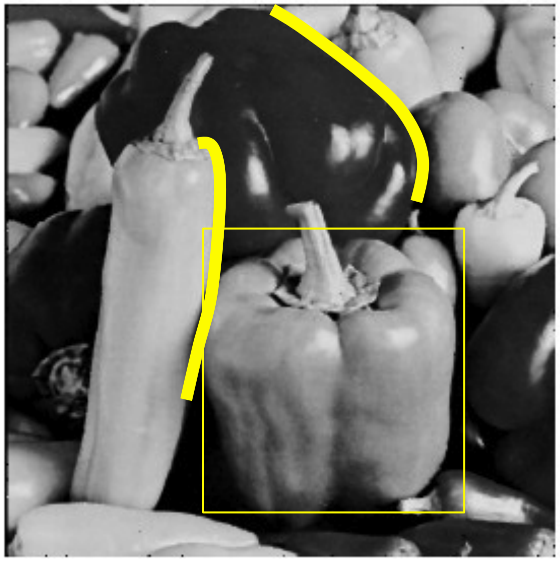
Edge detection

Feature detection/search

Source: T. Darrell

Filters for features

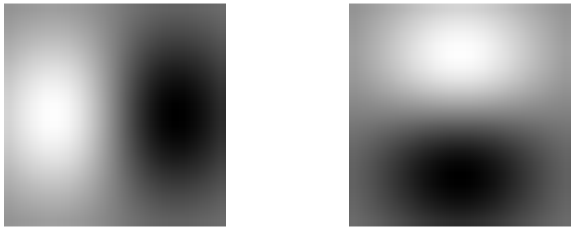
- Previously, thinking of filtering as a way to remove or reduce **noise**
- Now, consider how filters will allow us to abstract higher-level "**features**".
  - Map raw pixels to an intermediate representation that will be used for subsequent processing
  - Goal: reduce amount of data, discard redundancy, preserve what's useful



Source: T. Darrell

Filters as “templates”

- Note that filters look like the effects they are intended to find --- “matched filters”



- Use normalized cross-correlation score to find a given pattern (template) in the image.
  - Szeliski Eq. 8.11
- Normalization needed to control for relative brightnesses.

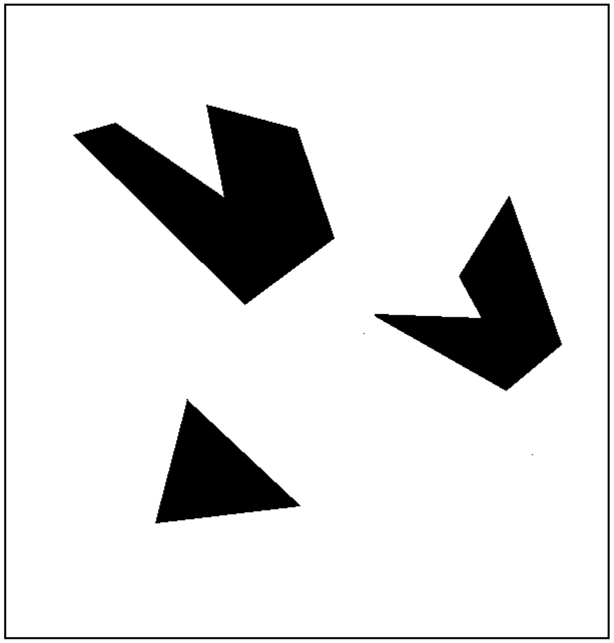
Source: T. Darrell

Normalized cross correlation

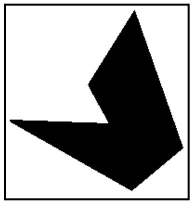
$$E_{\mathrm{NCC}}(\boldsymbol{u}) = \frac{\sum_i [I_0(\boldsymbol{x}_i) - \overline{I_0}]\,[I_1(\boldsymbol{x}_i + \boldsymbol{u}) - \overline{I_1}]}{\sqrt{\sum_i [I_0(\boldsymbol{x}_i) - \overline{I_0}]^2 [I_1(\boldsymbol{x}_i + \boldsymbol{u}) - \overline{I_1}]^2}},$$

$$\begin{aligned} \overline{I_0} &= \frac{1}{N} \sum_i I_0(\boldsymbol{x}_i) \quad \text{and} \\ \overline{I_1} &= \frac{1}{N} \sum_i I_1(\boldsymbol{x}_i + \boldsymbol{u}) \end{aligned}$$

Template matching



Scene

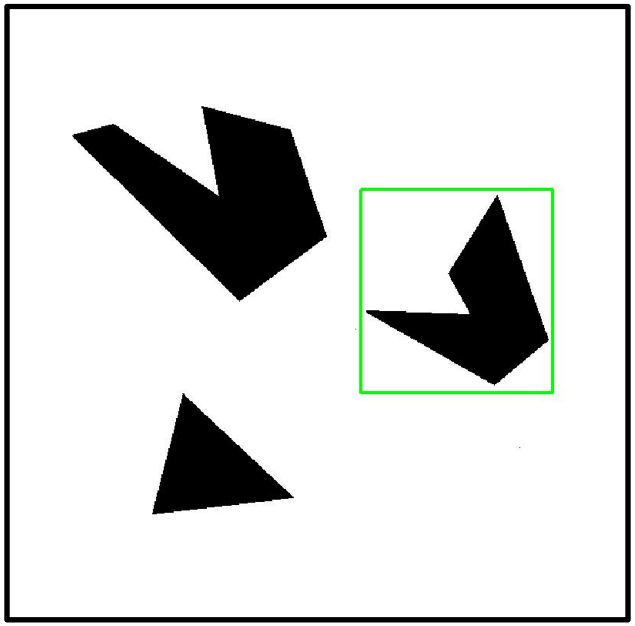


Template (mask)

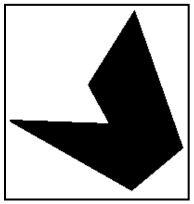
A toy example

Source: T. Darrell

Template matching



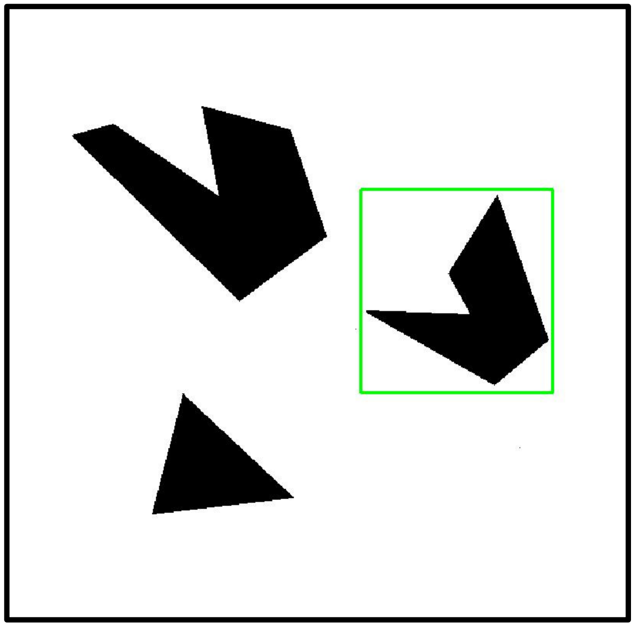
Detected template



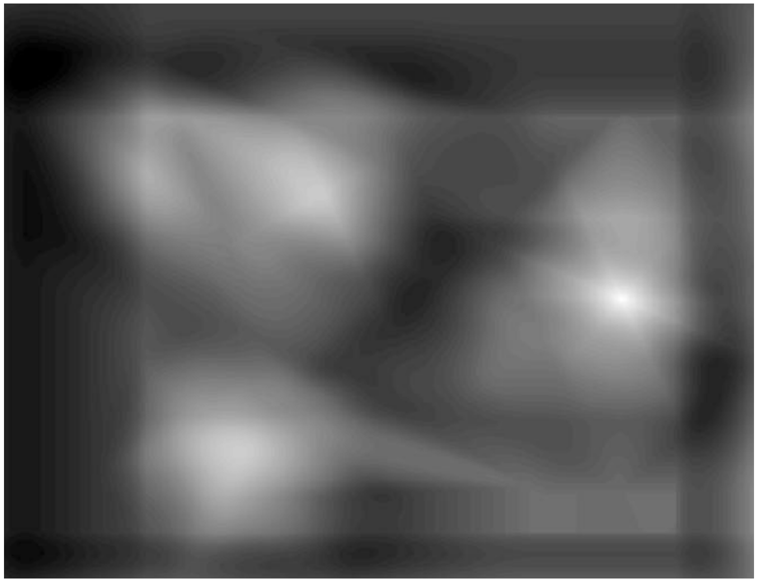
Template

Source: T. Darrell

Template matching



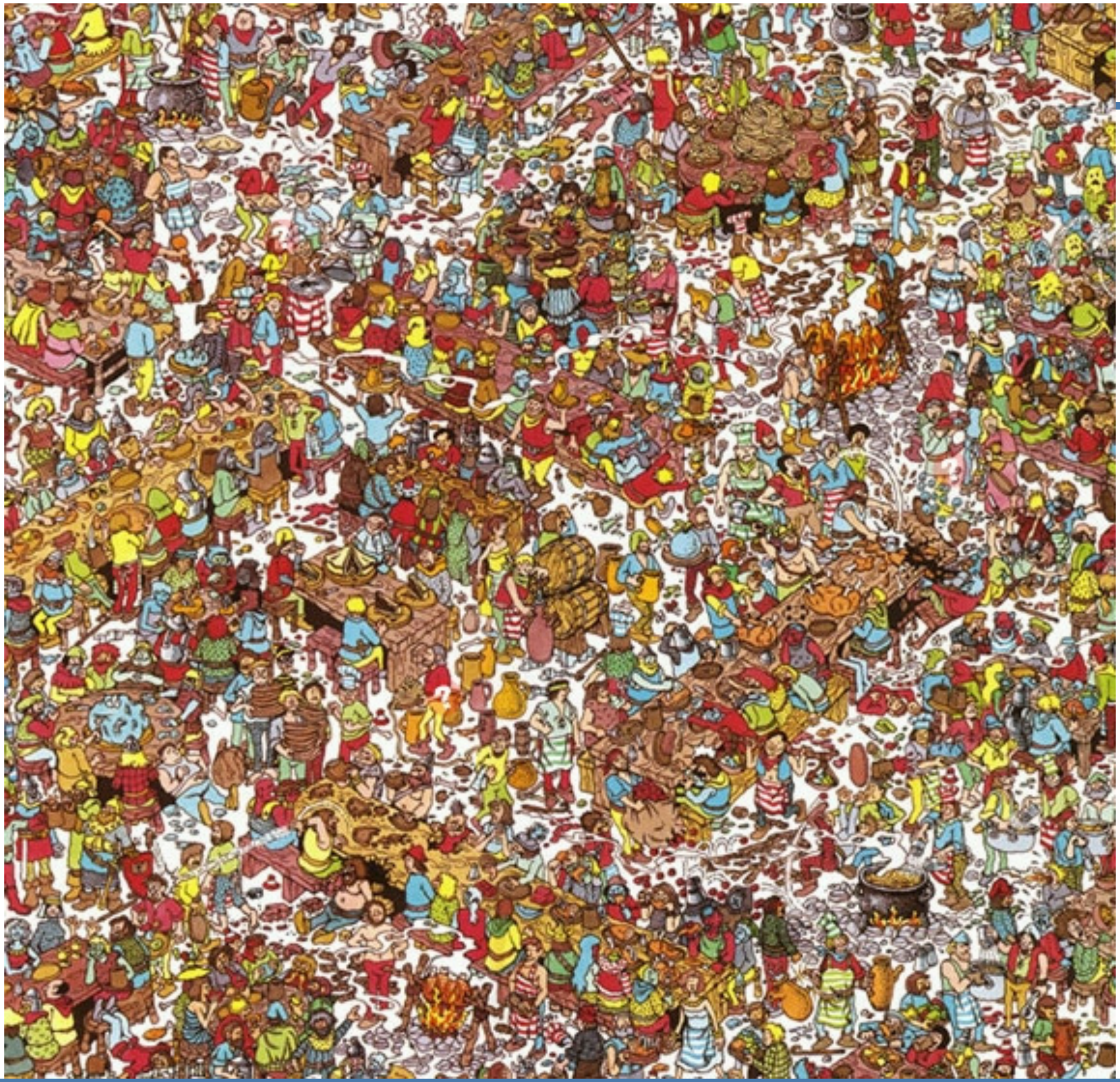
Detected template



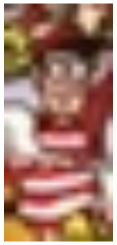
Correlation map

Source: T. Darrell

Where's Waldo?



Scene

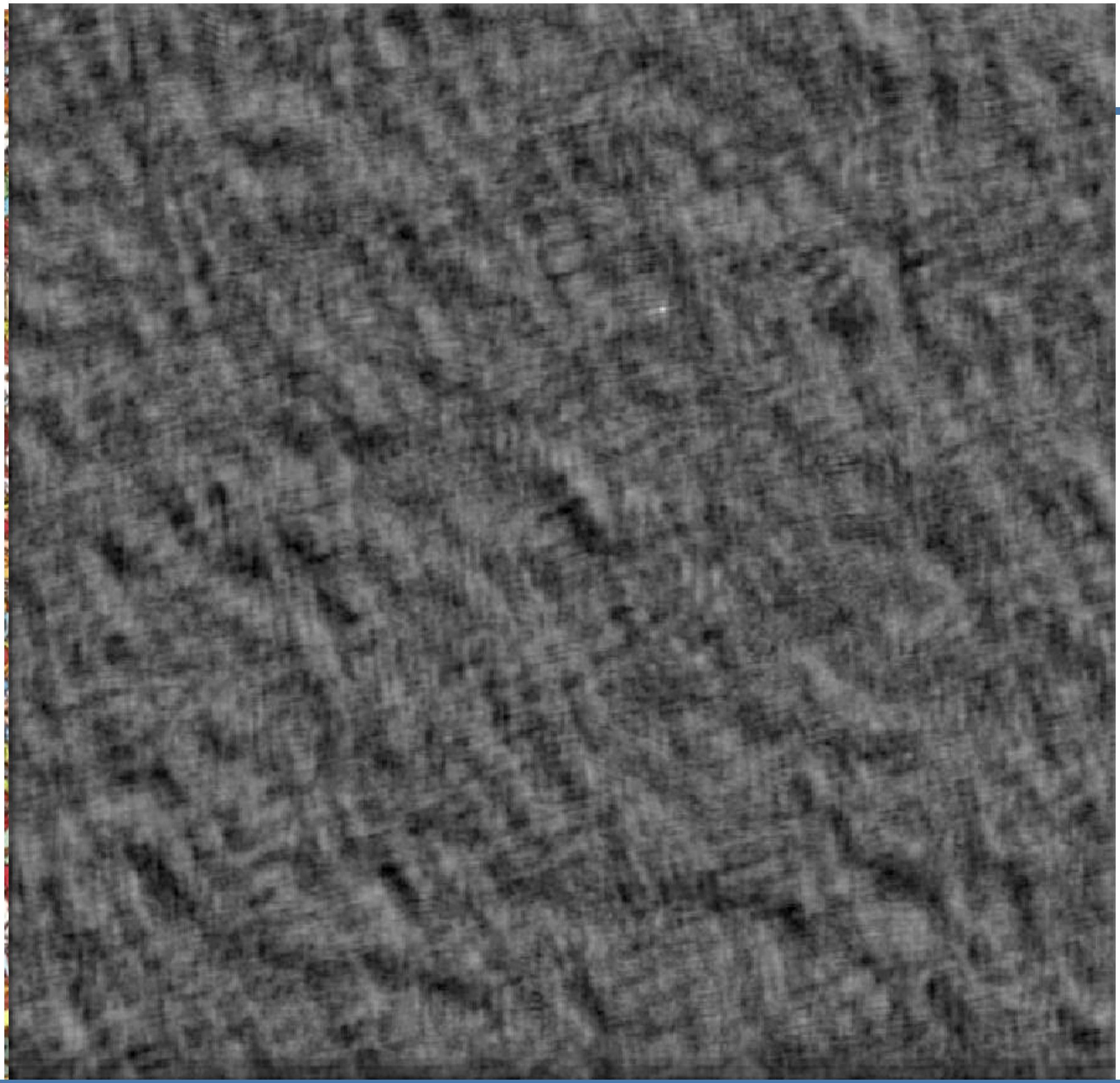


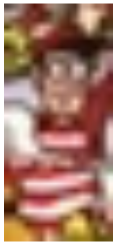
Template

Source: T. Darrell

©Michael J. Black

Where’s Waldo?

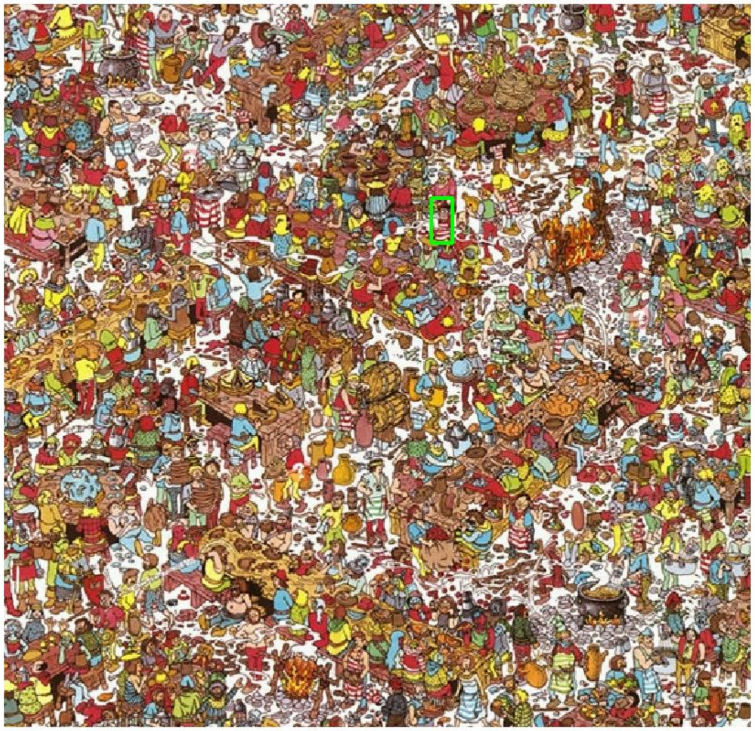




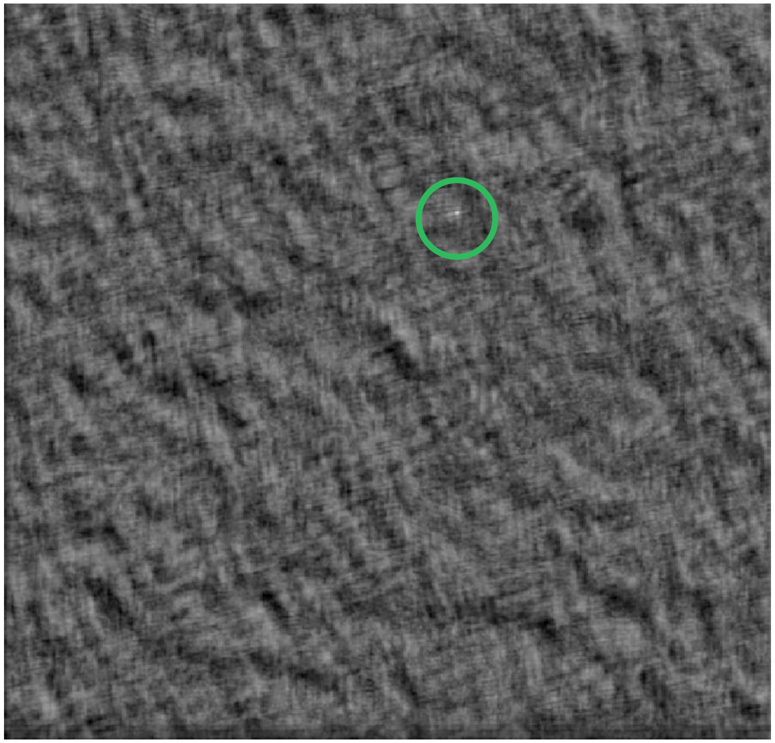
Template

Source: T. Darrell

Where's Waldo?



Detected template



Correlation map

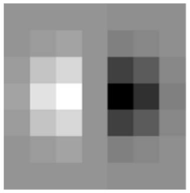
Source: T. Darrell

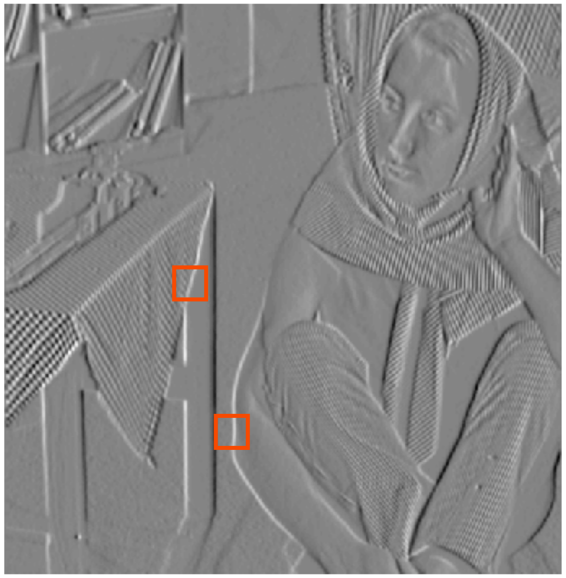
Let’s step back a moment…

Convolution

Correlation

Feature detection





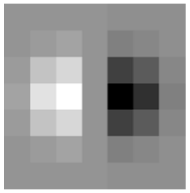
$$H[m,n] = f * I = \sum_{k,l} f[k,l]\, I[m+k, n+l]$$

Image as vectors

Convolution

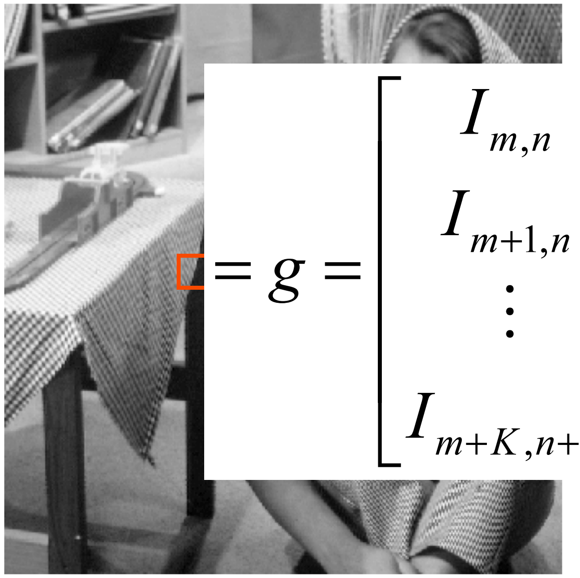
Correlation

Feature detection



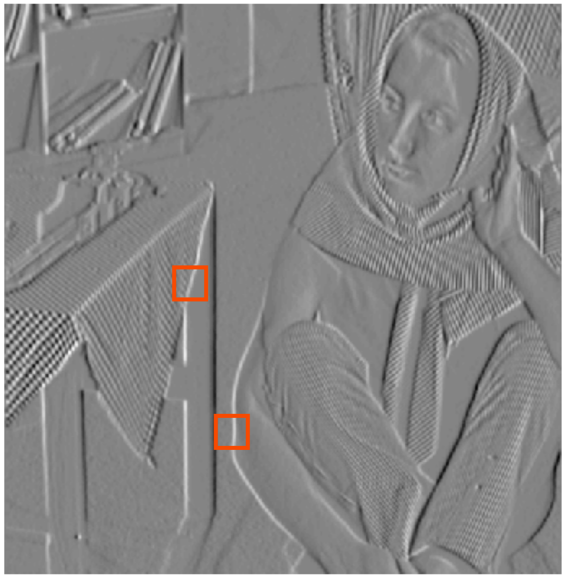
f

$$= \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{K\times L} \end{bmatrix}$$



I

$$= g = \begin{bmatrix} I_{m,n} \\ I_{m+1,n} \\ \vdots \\ I_{m+K,n+L} \end{bmatrix}$$



H

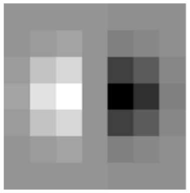
$$H[m,n] = f * I = \sum_{k,l} f[k,l]\, I[m+k,n+l]$$

Image as vectors

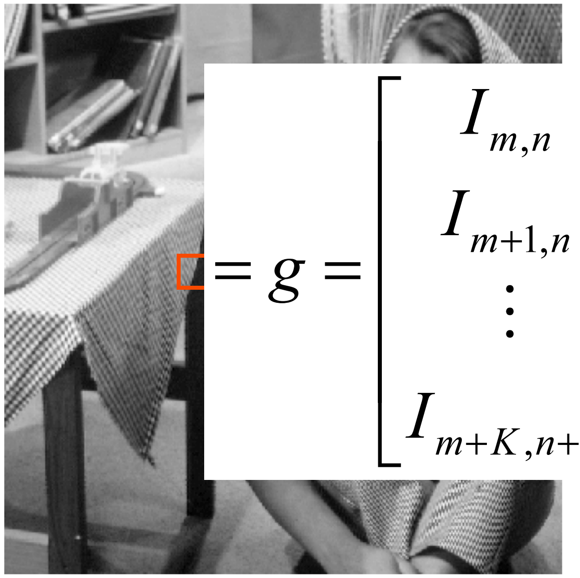
Convolution

Correlation

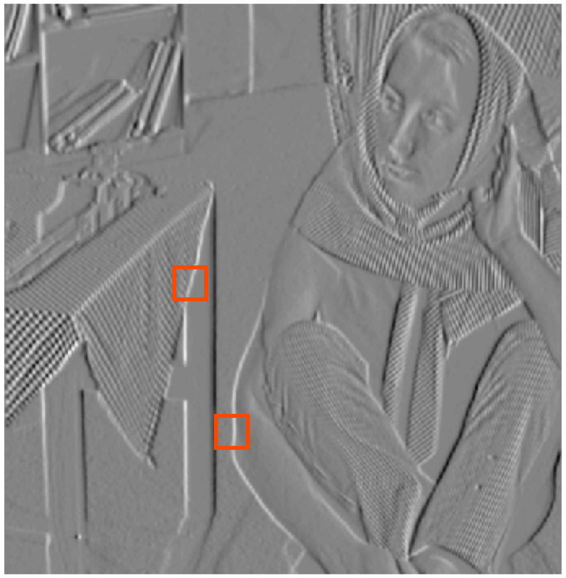
Feature detection



$$= \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{K\times L} \end{bmatrix}$$

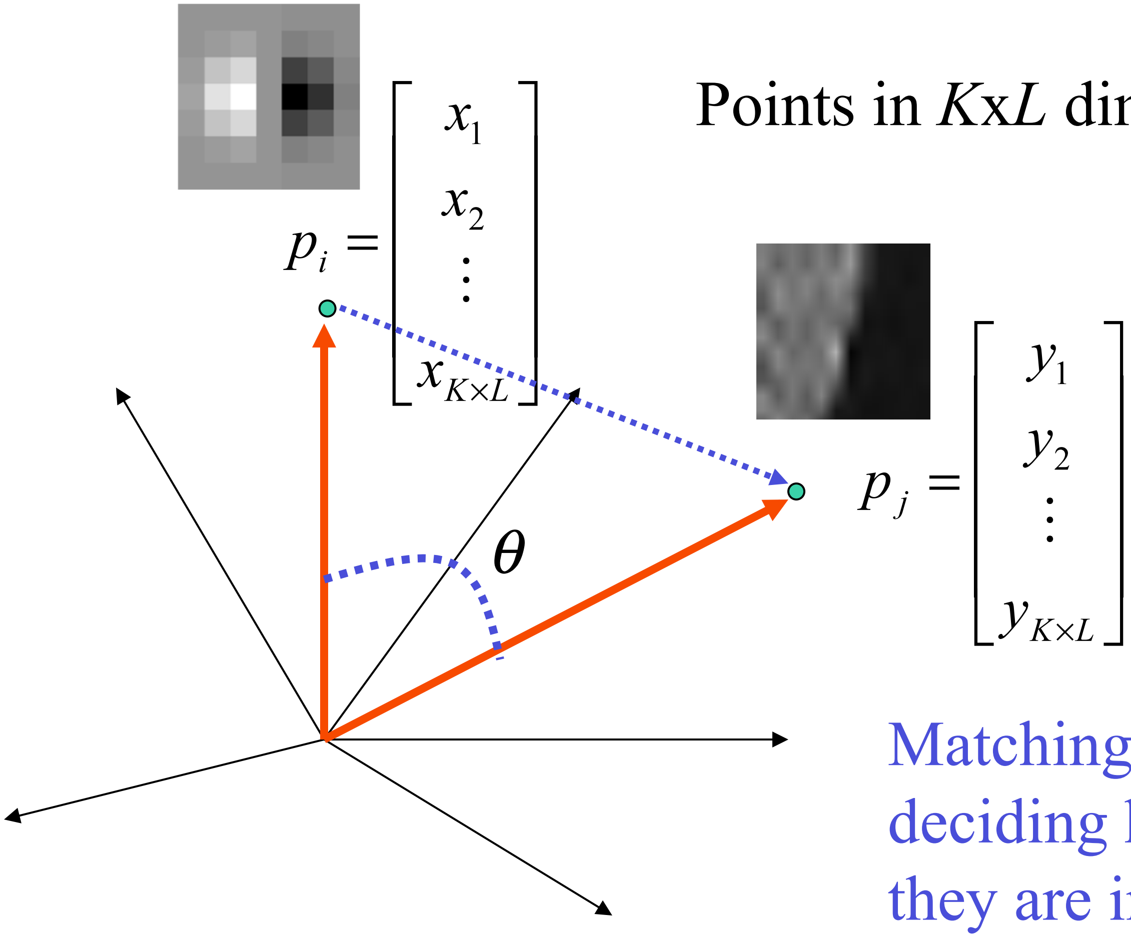


$$= g = \begin{bmatrix} I_{m,n} \\ I_{m+1,n} \\ \vdots \\ I_{m+K,n+L} \end{bmatrix}$$



$$H[m,n] = f \cdot g = f^T g = \sum_{i=1}^{KL} f[i]\,g[i]$$

Images as Points (Feature detection)



$$p_i = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{K\times L} \end{bmatrix}$$

$$p_j = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{K\times L} \end{bmatrix}$$

Points in *K*x*L* dimensional space

Matching involves deciding how far apart they are in this space.

Template Matching

$$p_i \cdot p_j = | p_i || p_j | \cos\theta$$

Angle between the vectors

Image patch as a vector

Template or filter as a vector

$$| p | = \sqrt{p_1^2 + p_1^2 + \dots p_n^2}$$

Template Matching

Correlation (sum of product of “signals”)

$$\boxed{p_i \cdot p_j} = |p_i||p_j| \cos\theta$$

Angle between the vectors

Image patch as a vector

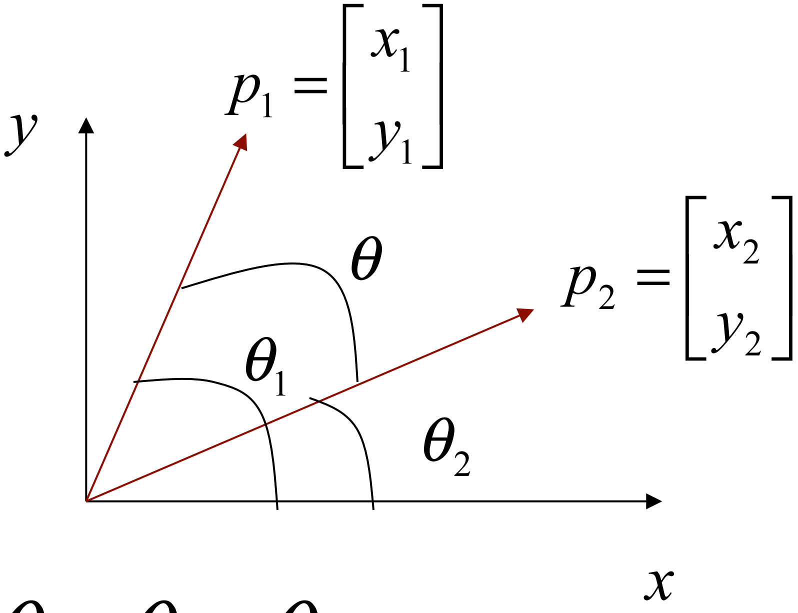
Template or filter as a vector

Normalized correlation:

$$\frac{p_i \cdot p_j}{|p_i||p_j|} = \cos\theta$$

Dot Product

Consider the 2D case. Want to know θ



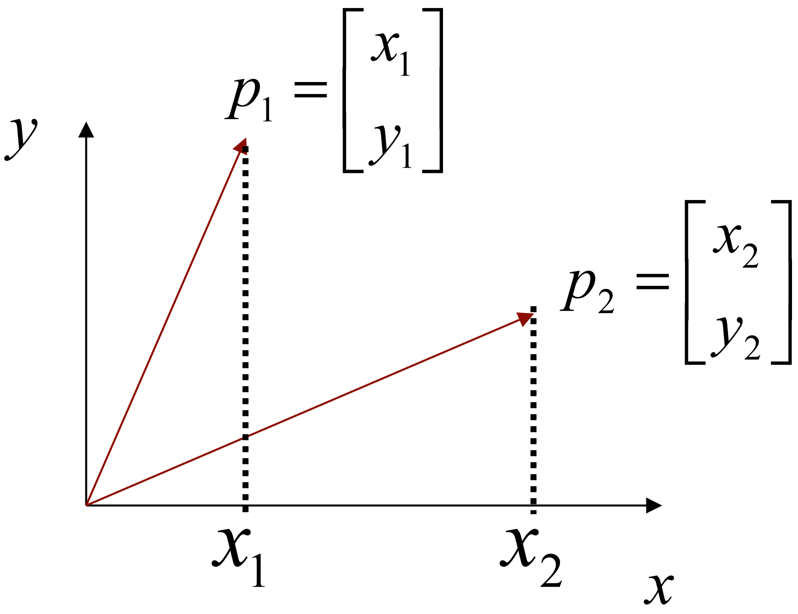
$$p_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$p_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

Note that $\theta = \theta_1 - \theta_2$

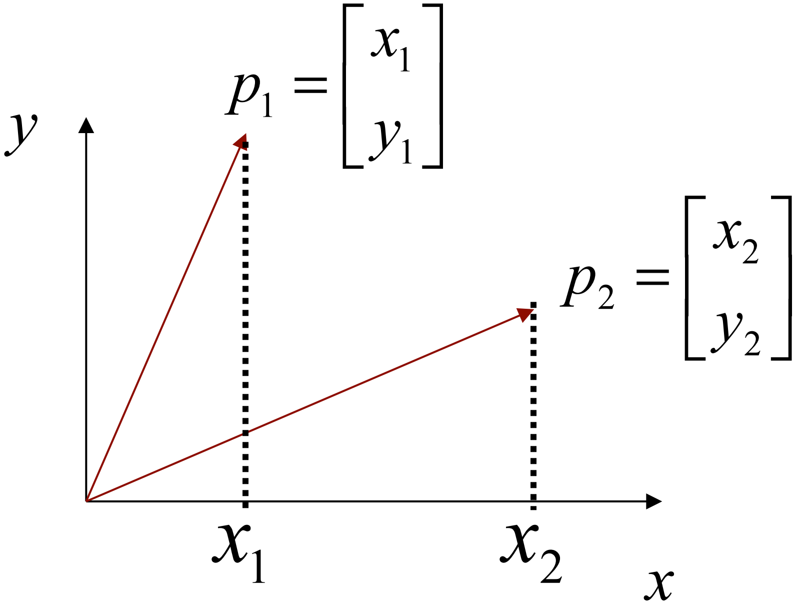
Also $\cos\theta = \cos(\theta_1 - \theta_2)$

Dot Product


$p_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$
$p_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$
y
x_1
x_2
x

$$\cos\theta = \cos(\theta_1 - \theta_2)$$
$$= \cos\theta_1 \cos\theta_2 + \sin\theta_1 \sin\theta_2$$

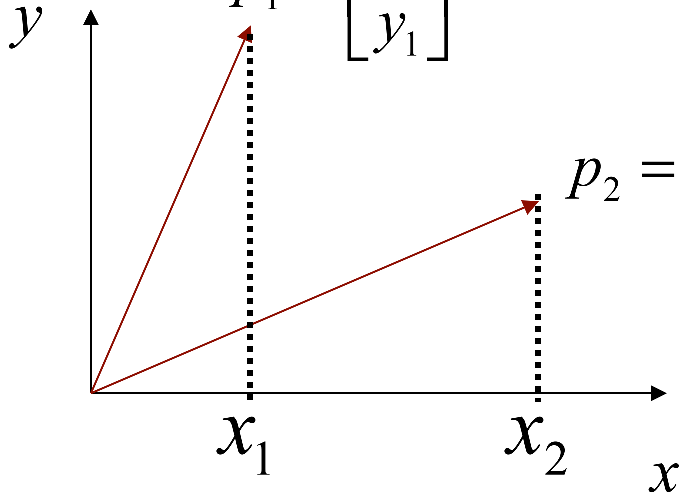
Dot Product


$p_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$
$p_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$
y
x_1
x_2
x

$$\cos\theta = \cos\theta_1 \cos\theta_2 + \sin\theta_1 \sin\theta_2$$

$$= \frac{x_1}{|p_1|}\frac{x_2}{|p_2|} + \frac{y_1}{|p_1|}\frac{y_2}{|p_2|} = \frac{x_1 x_2 + y_1 y_2}{|p_1||p_2|}$$

Dot Product

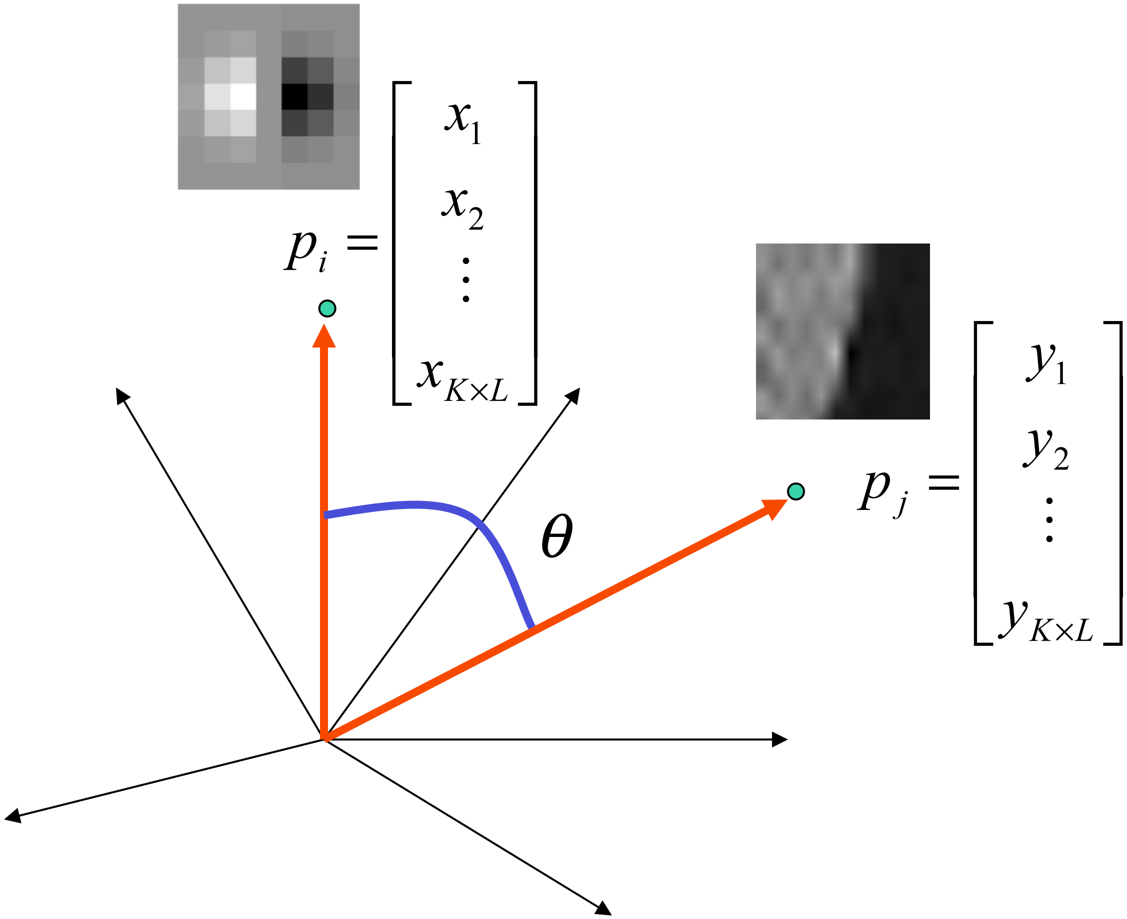


$$p_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$p_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

$$|p_1||p_2|\cos\theta = x_1x_2 + y_1y_2 = p_1 \cdot p_2$$

Dot Product



$$p_i = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{K \times L} \end{bmatrix}$$

$$p_j = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{K \times L} \end{bmatrix}$$

One more connection…

- Problem 4 in Asgn 1 uses corr2 which computes the correlation coefficient, r, defined as:

$$r = \frac{\sum_{k,l} (f(k,l) - \bar{f})(g(k,l) - \bar{g})}{\sqrt{\left(\sum_{k,l} (f(k,l) - \bar{f})^2\right)\left(\sum_{k,l} (g(k,l) - \bar{g})^2\right)}}$$

Where $\bar{g}$, $\bar{f}$ are the means of the patches.

One more connection…

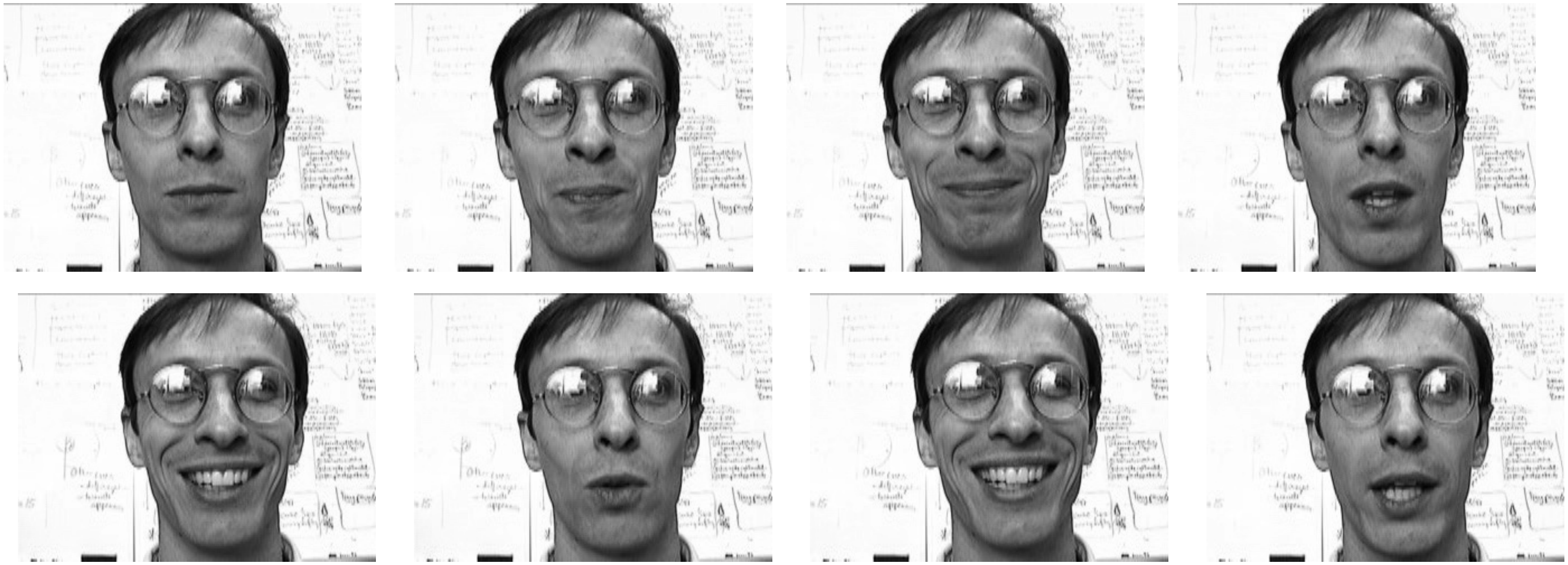
If g, f are zero mean, then

$$r = \frac{\sum_{k,l} (f(k,l) - \bar{f})(g(k,l) - \bar{g})}{\sqrt{\left(\sum_{k,l} (f(k,l) - \bar{f})^2\right)\left(\sum_{k,l} (g(k,l) - \bar{g})^2\right)}}$$

This is just normalized correlation:

$$\frac{f \cdot g}{|f||g|} = \cos\theta$$

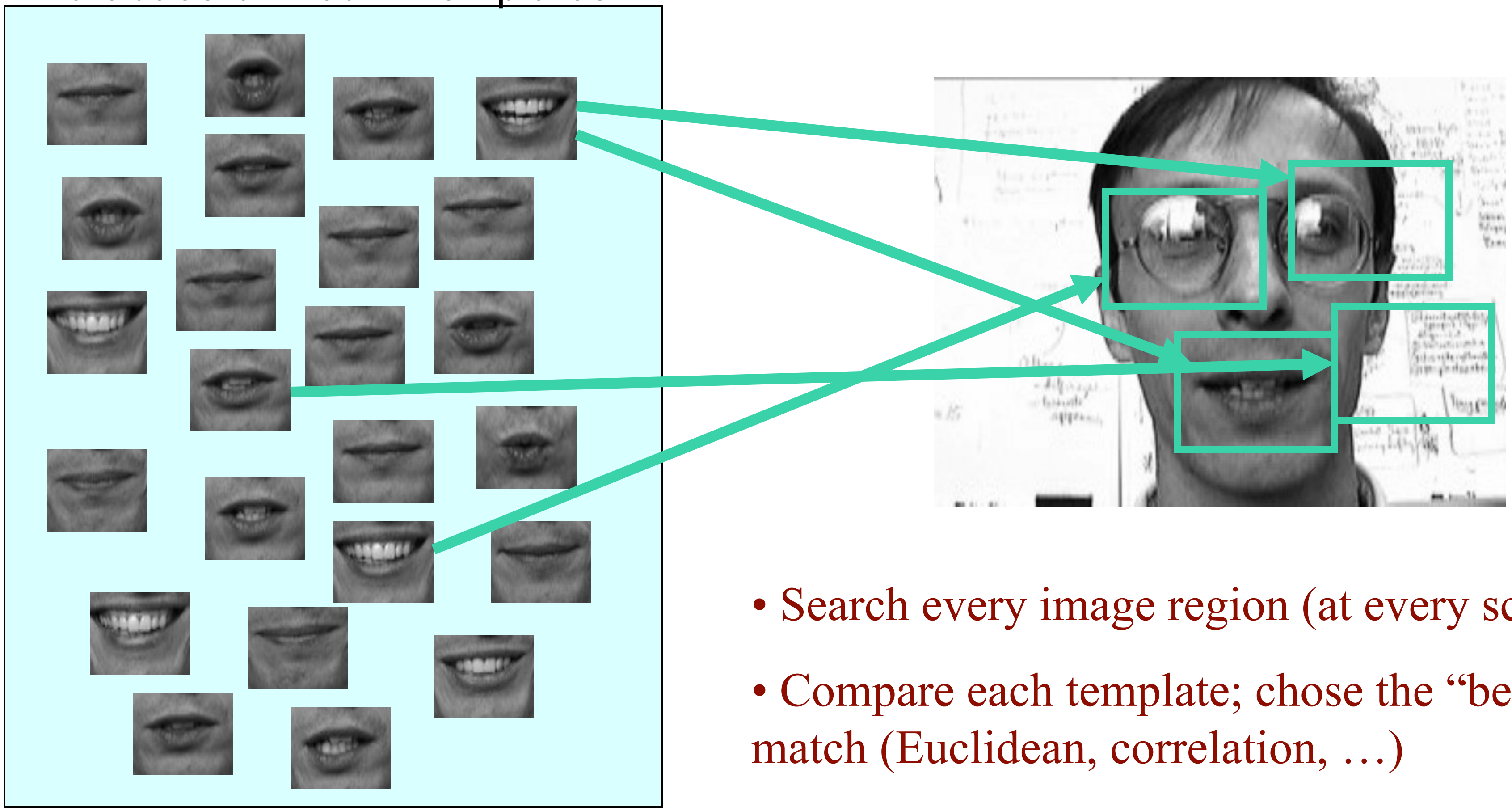
Search and Recognition.



1. How can we find the mouth?
2. How can we recognize the "expression"?

Naïve Appearance-Based Approach

Database of mouth “templates”



- Search every image region (at every scale).
- Compare each template; chose the “best” match (Euclidean, correlation, …)

Appearance-Based Methods

Represent objects by their appearance in an ensemble of images, including different poses, illuminants, configurations of shape, …

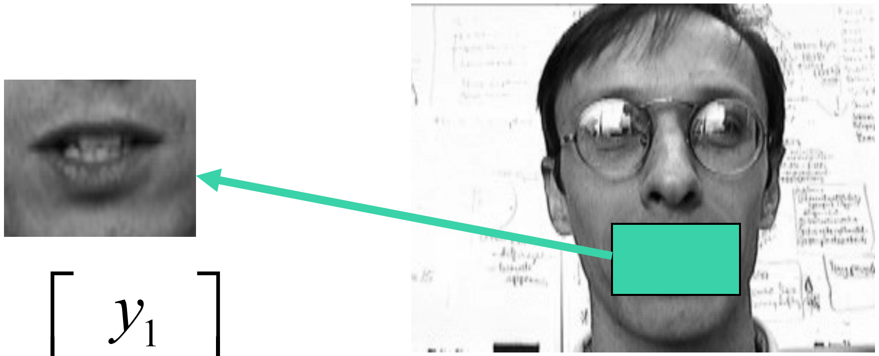
Approaches covered here:

- Subspace (eigen) Methods
- Local Invariant Image Features

Fleet & Szeliski

Images as Vectors

$$= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n\times m} \end{bmatrix}$$

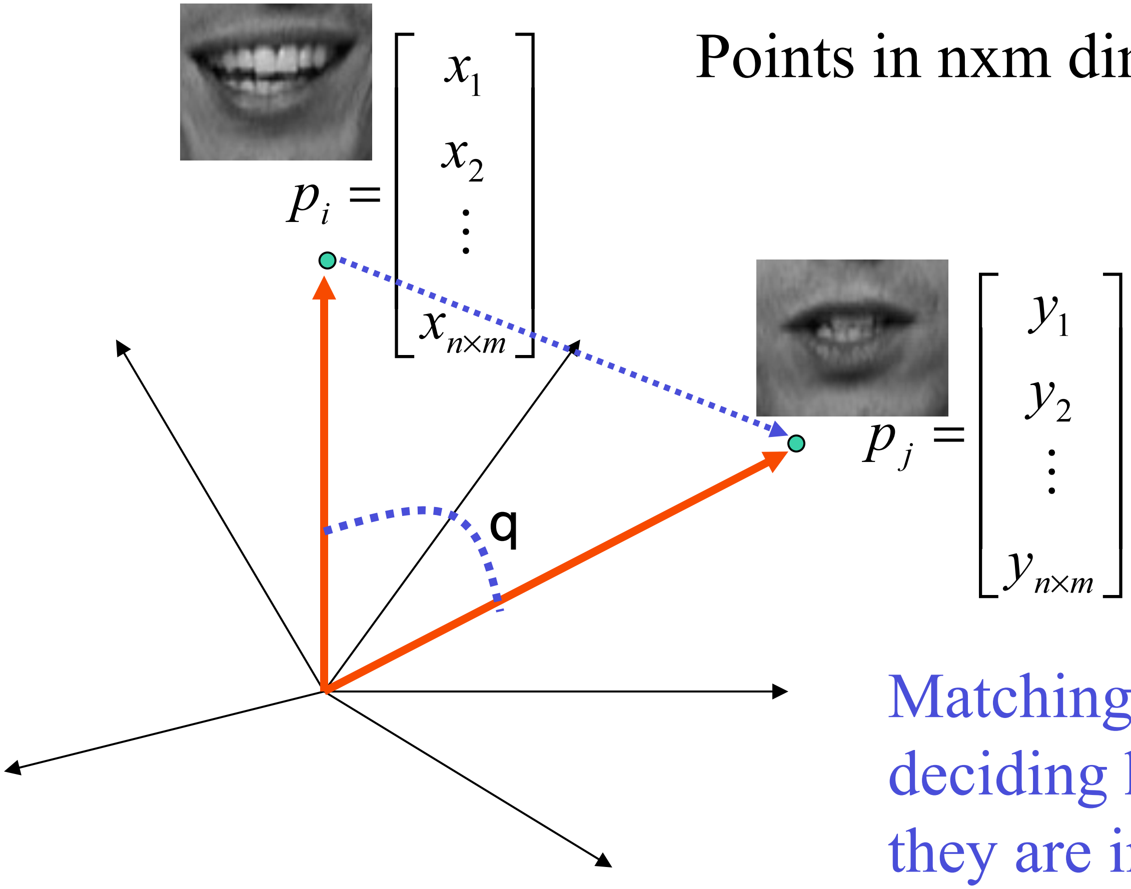


$$= \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n\times m} \end{bmatrix}$$

e.g. standard lexicographic ordering

Images as Points

Points in nxm dimensional space



$$p_i = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n\times m} \end{bmatrix}$$

$$p_j = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n\times m} \end{bmatrix}$$

Matching involves deciding how far apart they are in this space.