

# Designing Descriptors

Computer Vision

**Carnegie Mellon University (Kris Kitani)**

Last session, we learned how to detect ‘corners’...

*Give a set of detected features,  
what computer vision tasks can I solve?*

# Object instance recognition



Schmid and Mohr 1997



Sivic and Zisserman, 2003

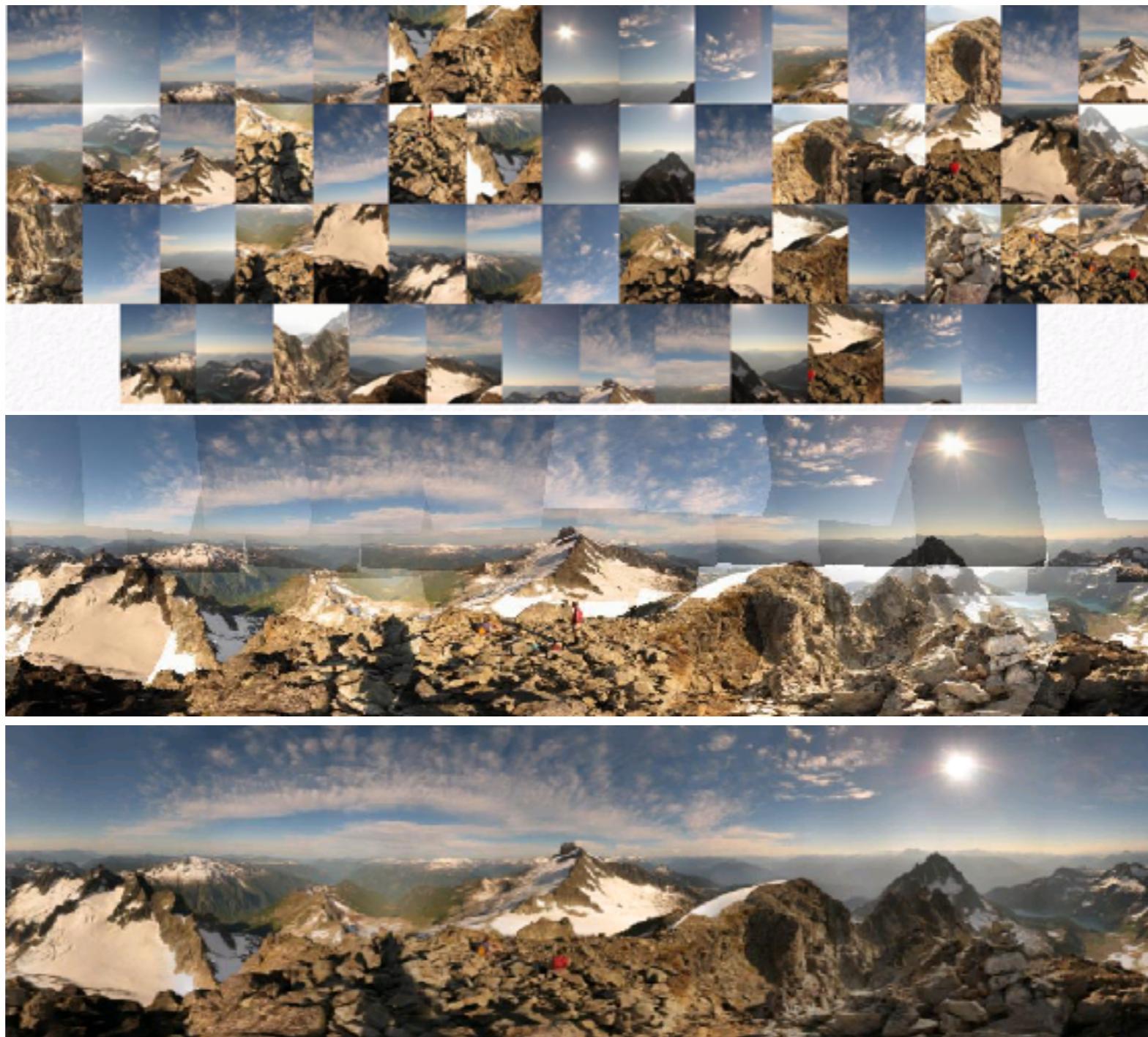


Rothganger et al. 2003

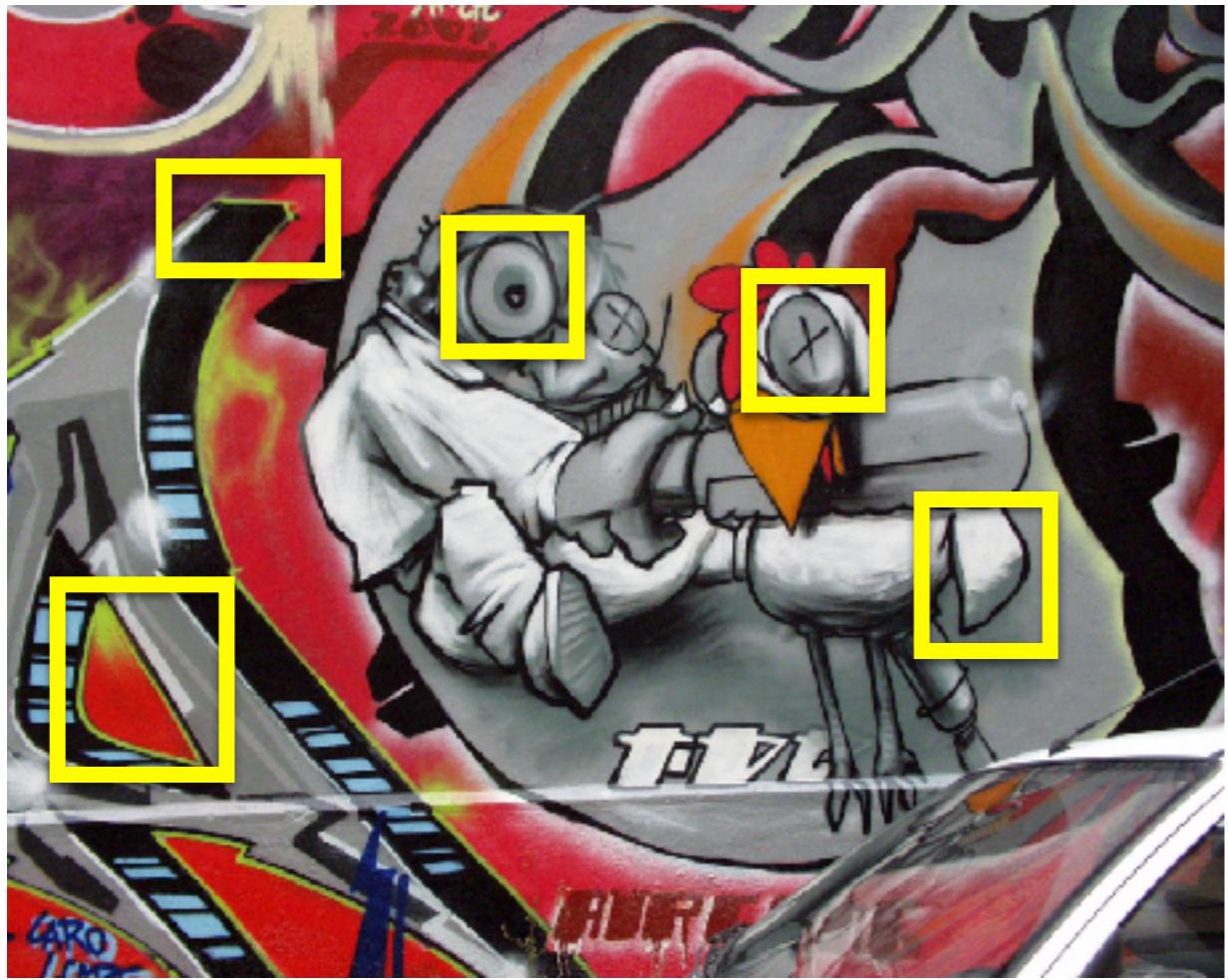


Lowe 2002

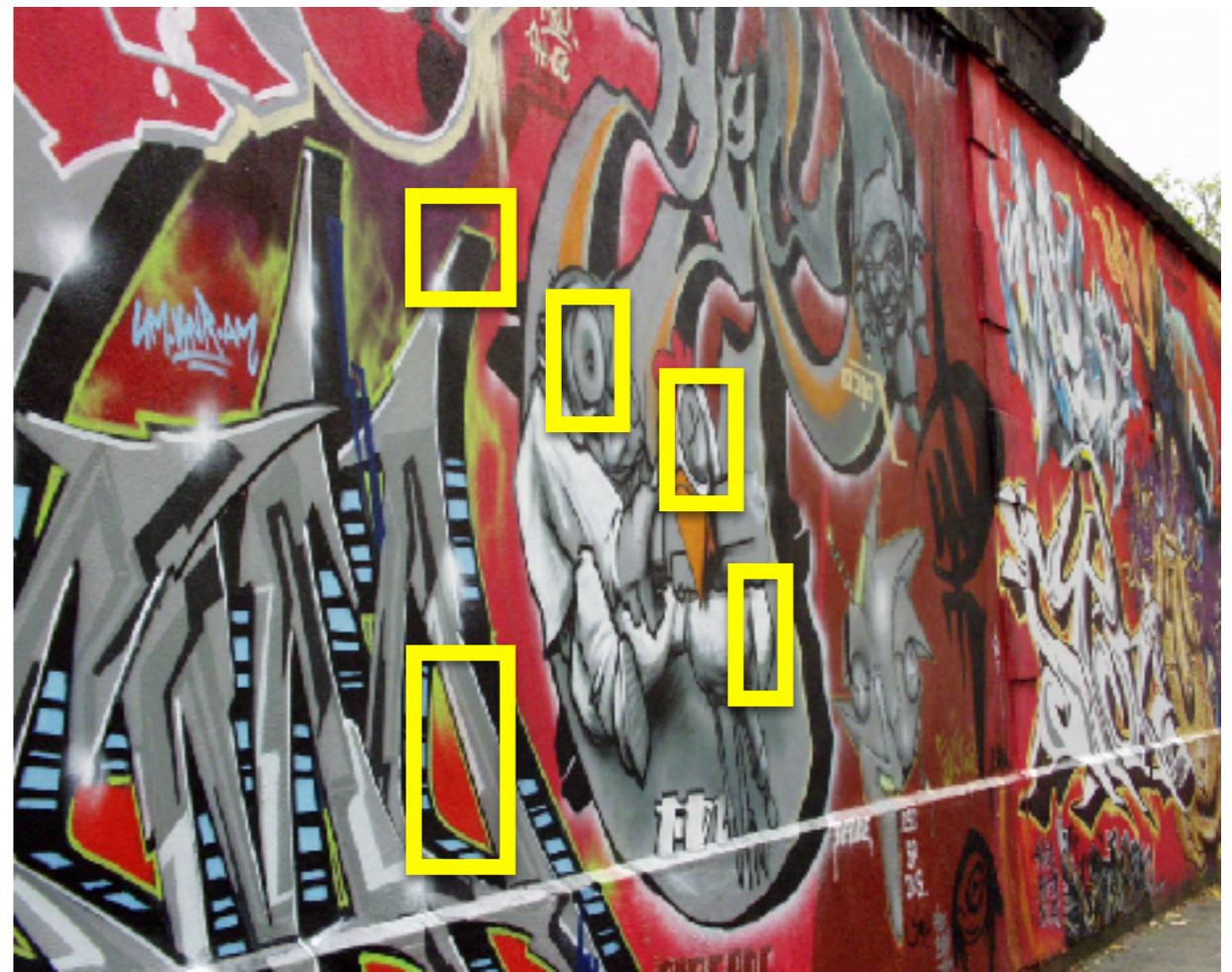
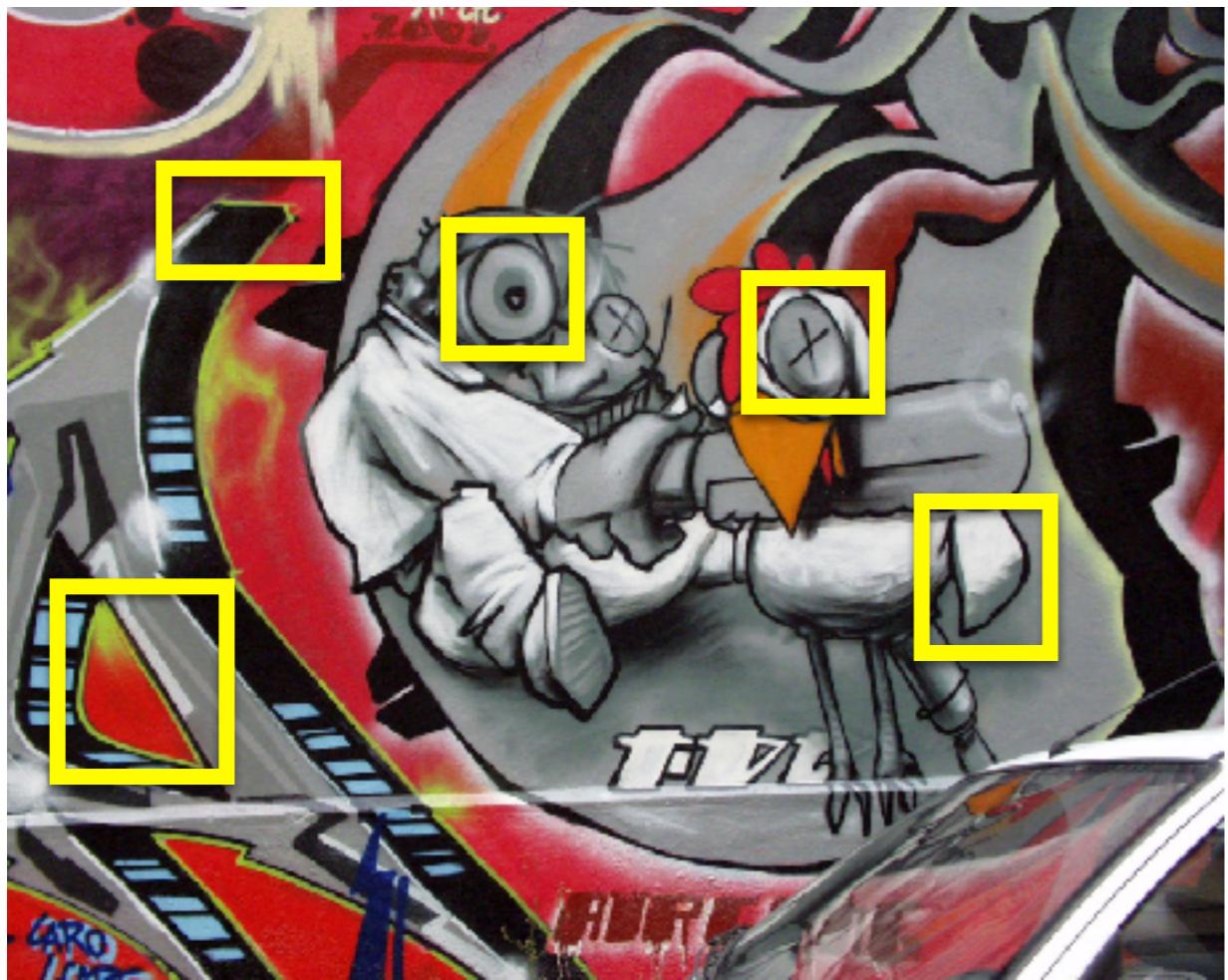
# Image mosaicing



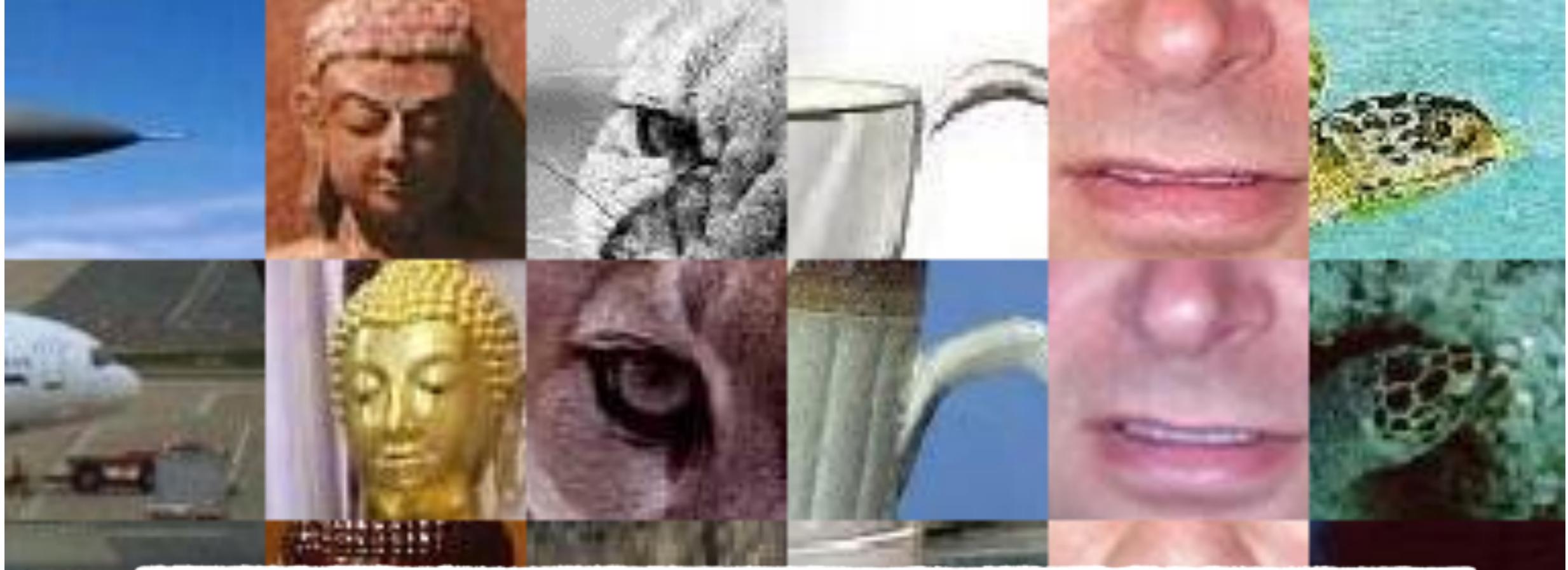
Many applications require that we  
**match feature descriptors.**



If we know where the good features are,

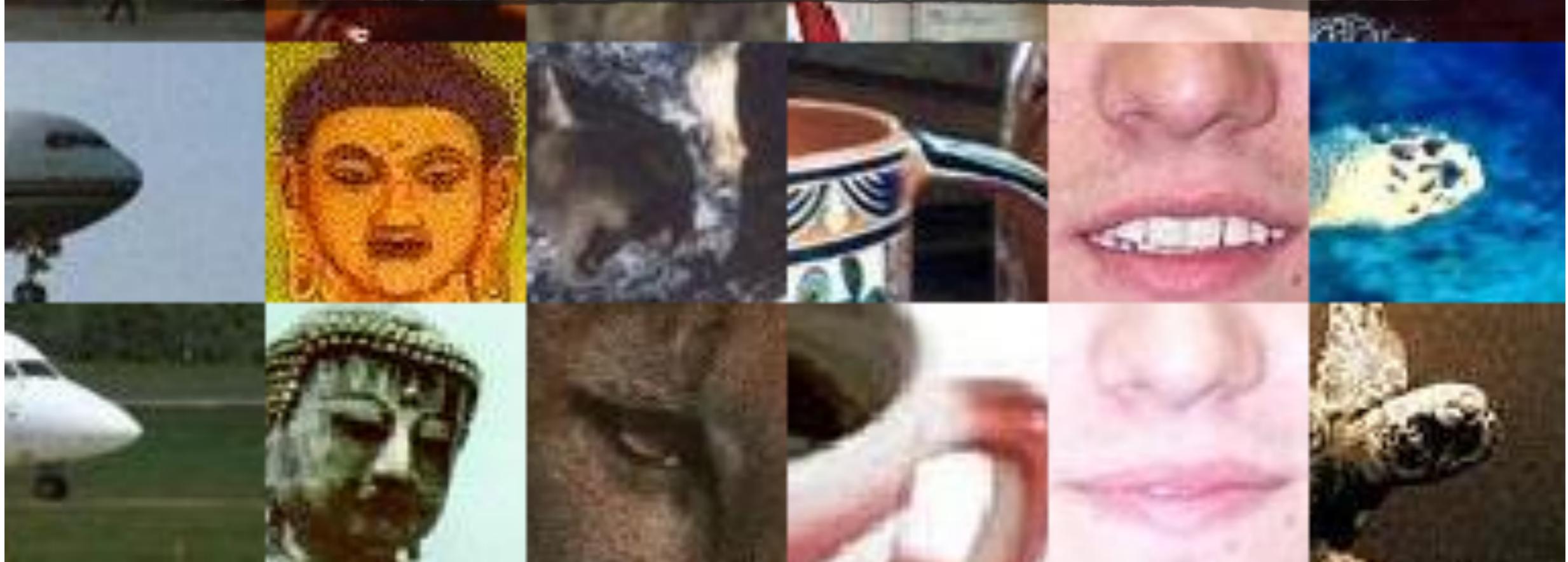


If we know where the good features are,  
how do we match them?



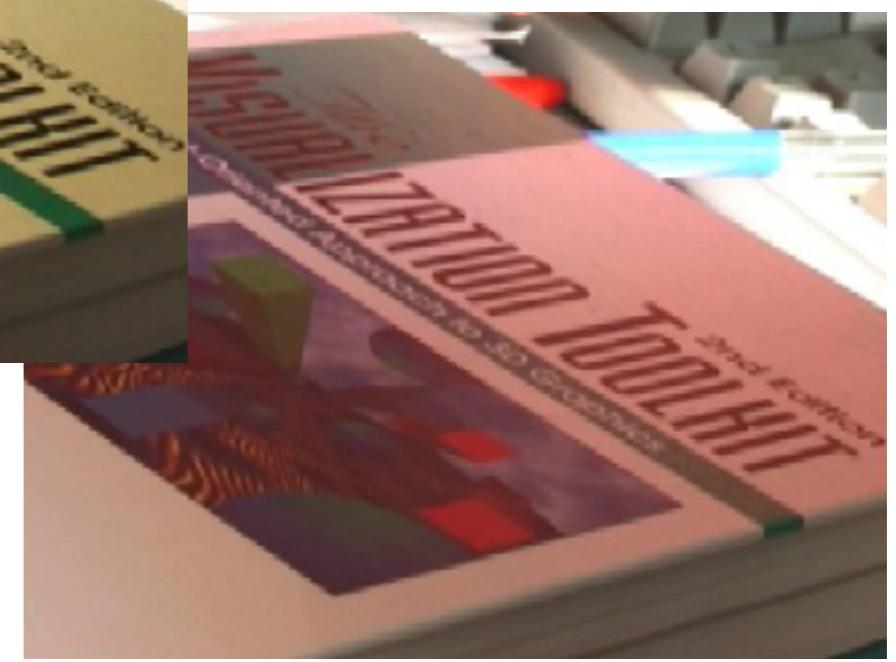
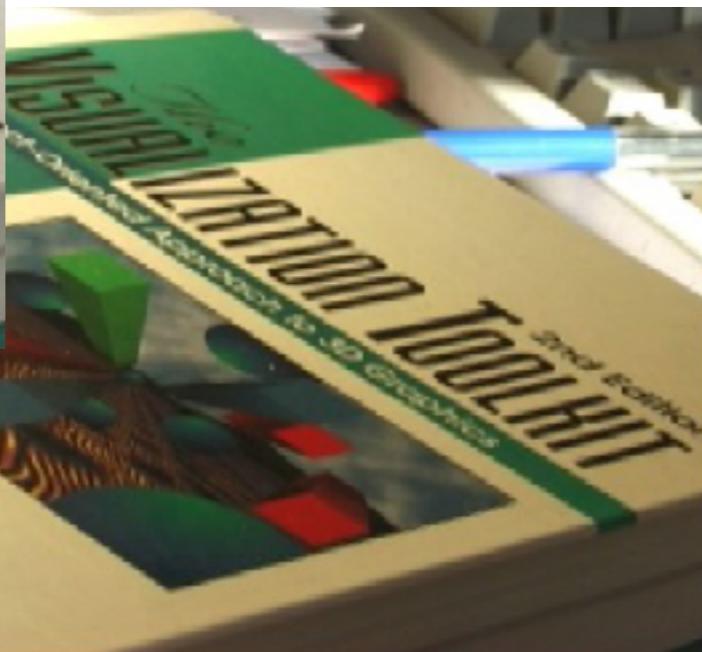
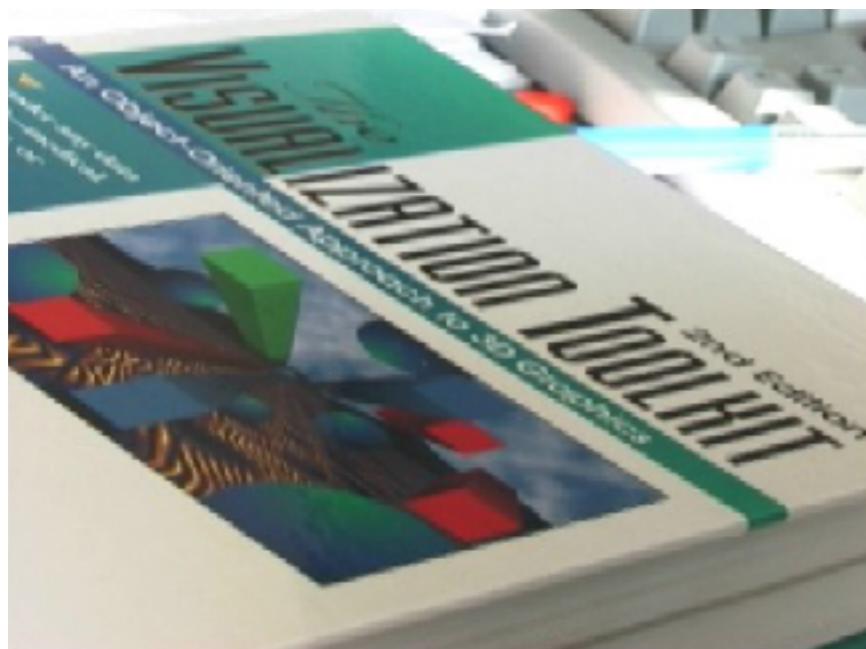
# How do we describe an image patch?

Patches with similar content should have similar descriptors.

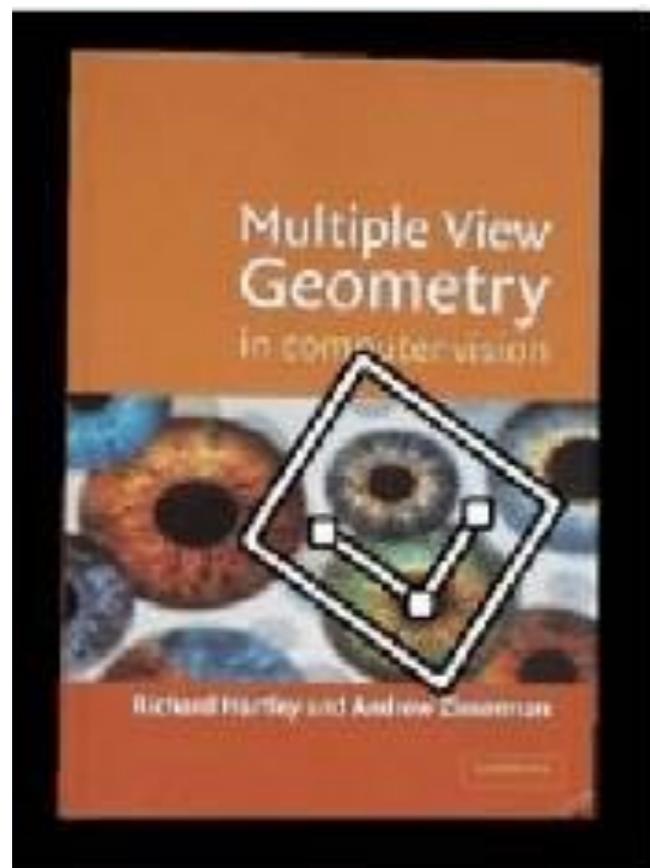


Challenges of designing a feature descriptor

# Photometric transformations



# Geometric transformations



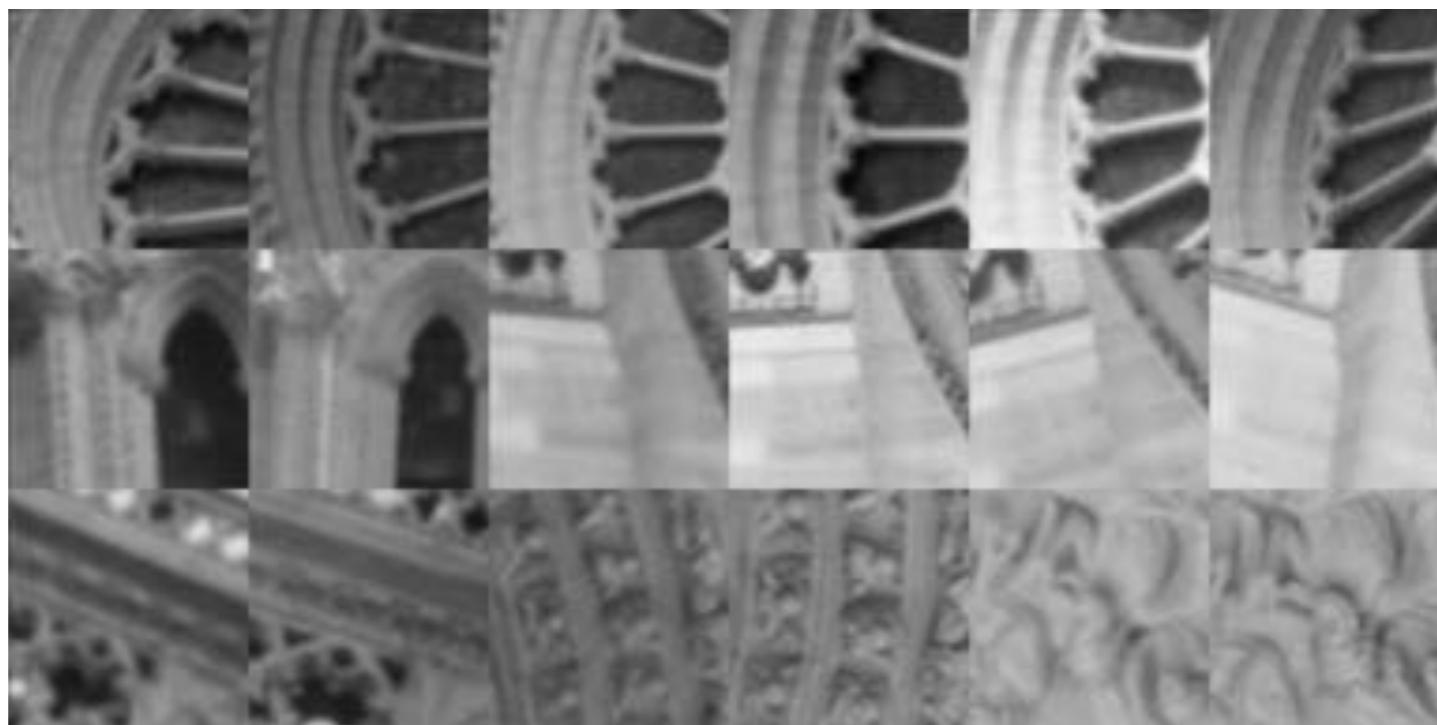
objects will appear at different scales,  
translation and rotation

# Designing a feature descriptor

(the search for image invariants)



*What is the best descriptor for an image feature?*



# Tiny Images



Just down-sample it!



Down-sampling (average pooling) is ...

Simple.

Fast.

Robust to small affine transformation.

*What are the problems?*

# Designing a feature descriptor

(the search for image invariants)

# Pixel Intensities

Just use the pixel values of the patch



Perfectly fine if geometry and appearance is unchanged  
(a.k.a. template matching)

*What are the problems?*

# Pixel Intensities

Just use the pixel values of the patch



Perfectly fine if geometry and appearance is unchanged  
(a.k.a. template matching)

*What are the problems?*

*How can you be less sensitive to absolute intensity values?*

# Image gradients

Use pixel differences

1	2	3
4	5	6
7	8	9



$$( \begin{array}{cccccc} 0 & 1 & 1 & 0 & 0 & 1 \\ - & + & + & - & - & + \end{array} )$$

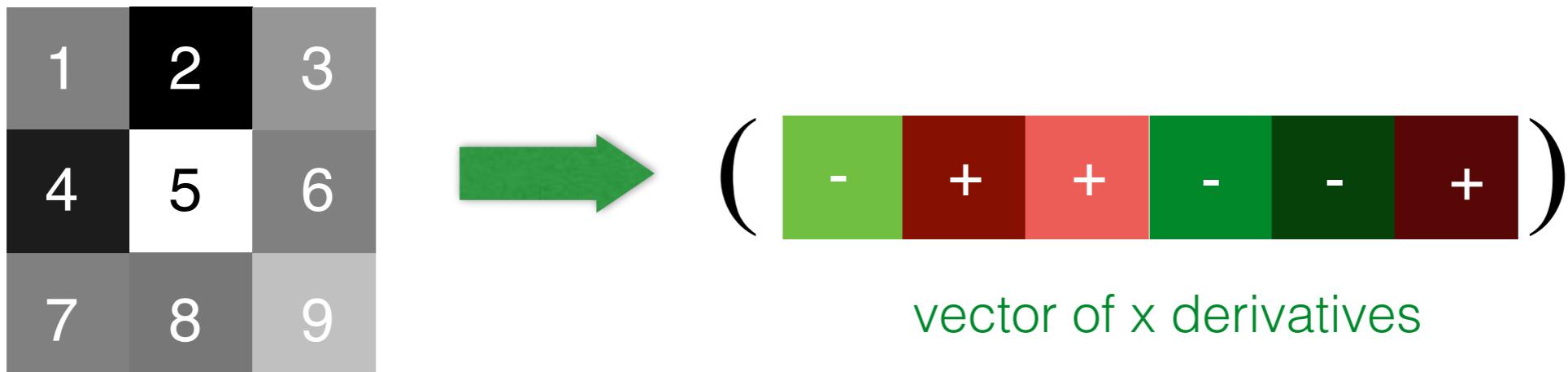
vector of x derivatives  
'binary descriptor'

Feature is invariant to absolute intensity values

*What are the problems?*

# Image gradients

Use pixel differences



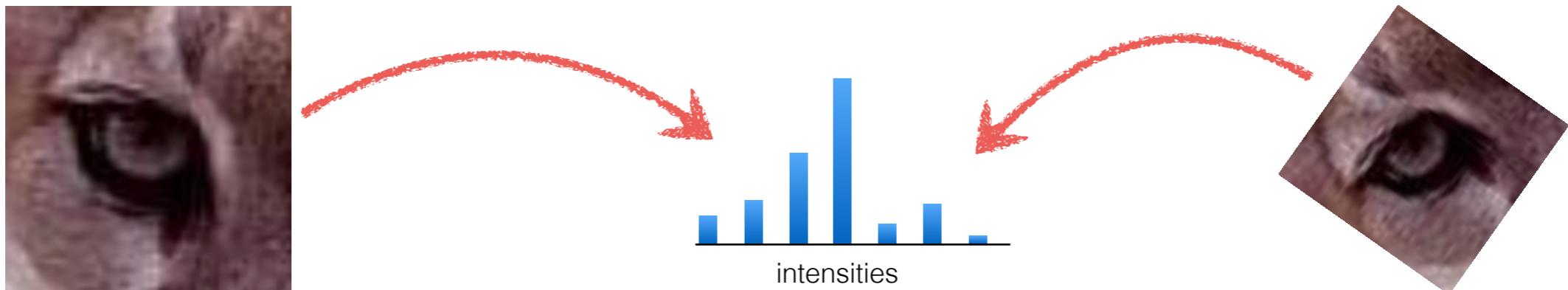
Feature is invariant to absolute intensity values

*What are the problems?*

*How can you be less sensitive to deformations?*

# Intensity histogram

Count the intensities in the image using a histogram

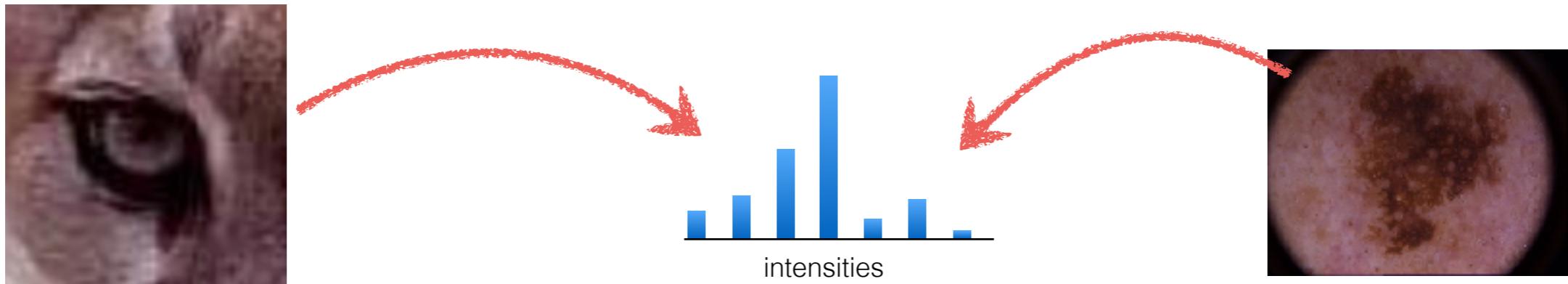


Invariant to changes in scale and rotation

*What are the problems?*

# Intensity histogram

Count the intensities in the image using a histogram

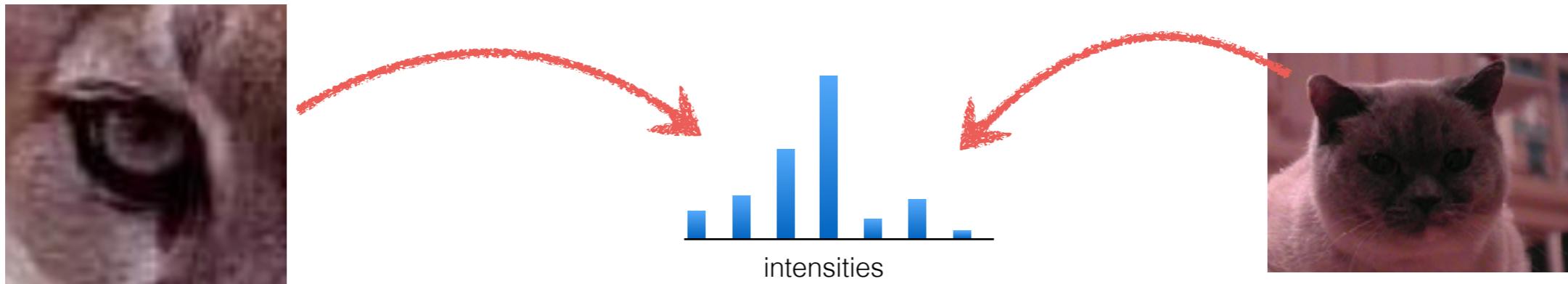


Invariant to changes in scale and rotation

*What are the problems?*

# Intensity histogram

Count the intensities in the image using a histogram



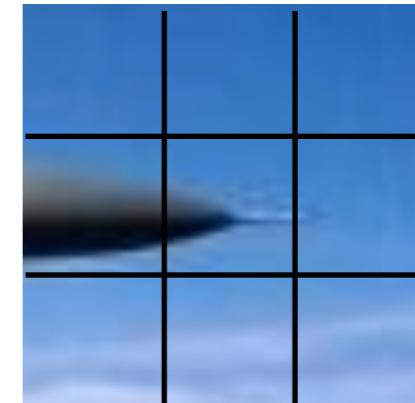
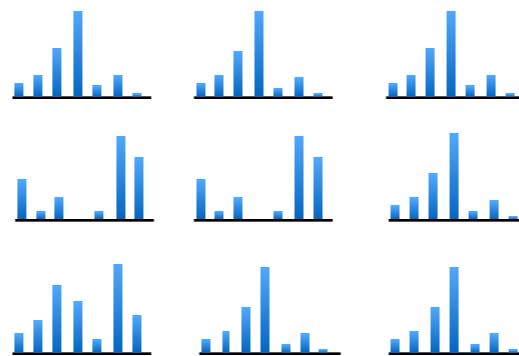
Invariant to changes in scale and rotation

*What are the problems?*

*How can you be more sensitive to spatial layout?*

# Spatial histograms

Compute histograms over spatial ‘cells’

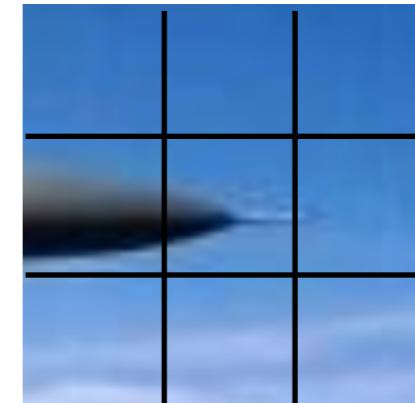
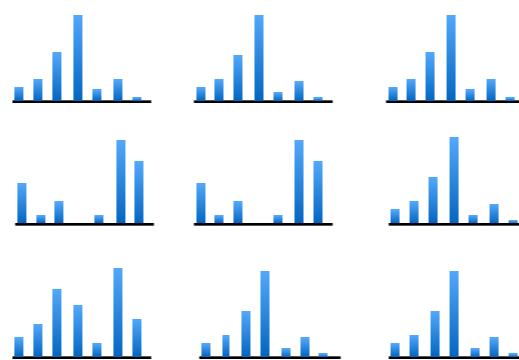


Retains rough spatial layout  
Some invariance to deformations

*What are the problems?*

# Spatial histograms

Compute histograms over spatial ‘cells’



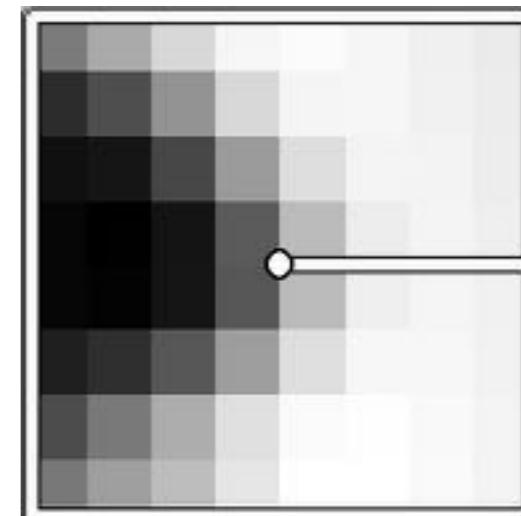
Retains rough spatial layout  
Some invariance to deformations

*What are the problems?*

*How can you be completely invariant to rotation?*

# Orientation normalization

Use the dominant image gradient direction to normalize the orientation of the patch



save the orientation angle  $\theta$  along with  $(x, y, s)$

*What are the problems?*