Automatic Image Alignment (feature-based)

© Mike Nese

with a lot of slides stolen from Steve Seitz and Rick Szeliski

15-463: Computational Photography
Alexei Efros, CMU, Fall 2006

Today's lecture

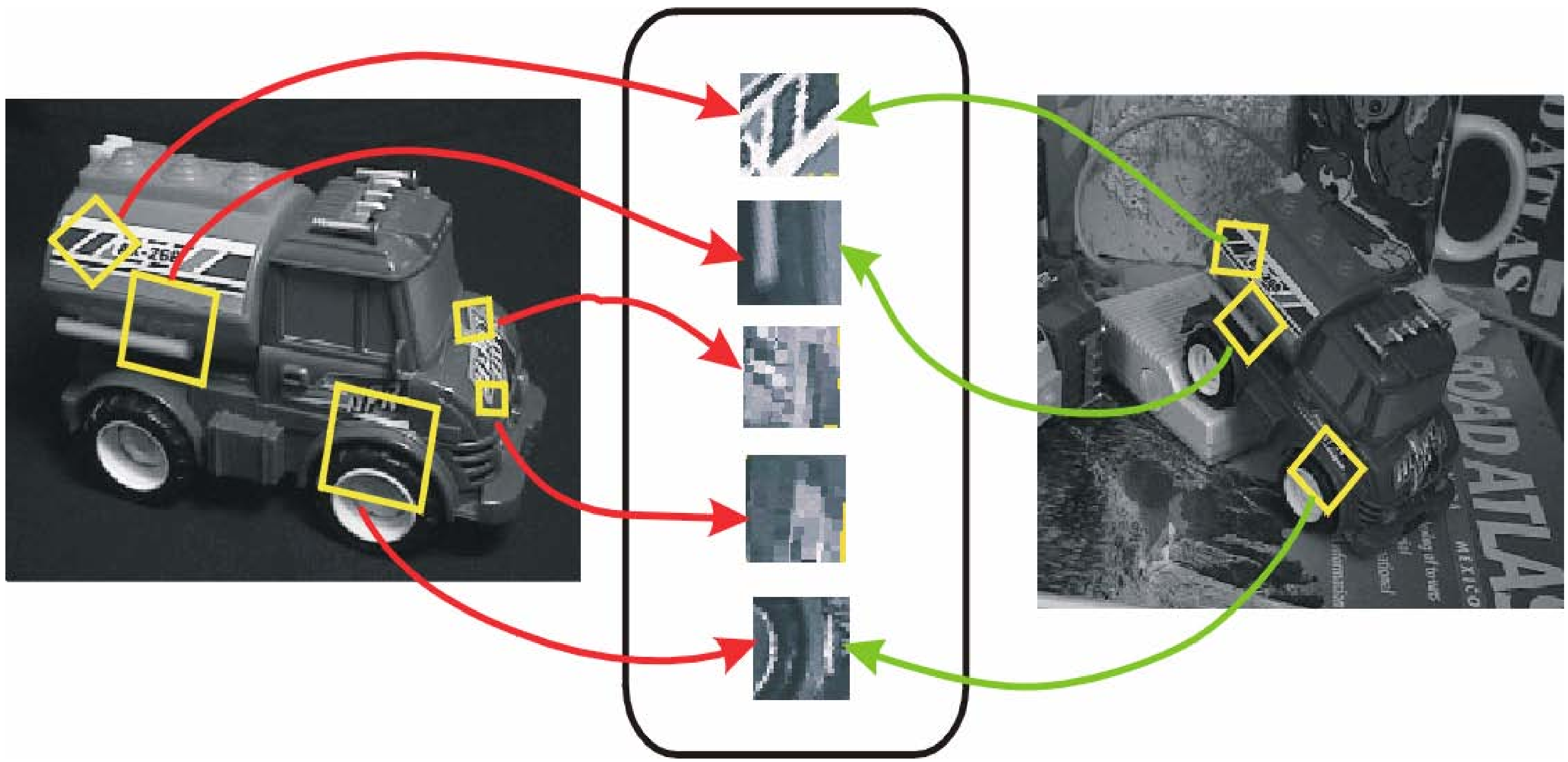
- Feature detectors
  - scale invariant Harris corners
- Feature descriptors
  - patches, oriented patches

Reading for Project #4:

Multi-image Matching using Multi-scale image patches, CVPR 2005

Invariant Local Features

Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



Features Descriptors

Advantages of local features

Locality: features are local, so robust to occlusion and clutter (no prior segmentation)

Distinctiveness: individual features can be matched to a large database of objects

Quantity: many features can be generated for even small objects

Efficiency: close to real-time performance

Extensibility: can easily be extended to wide range of differing feature types, with each adding robustness

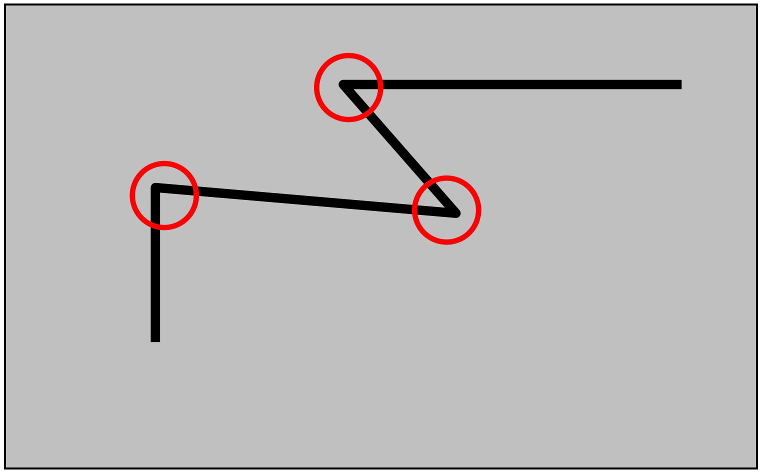
More motivation…

Feature points are used for:

- Image alignment (homography, fundamental matrix)
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation
- … other

Harris corner detector

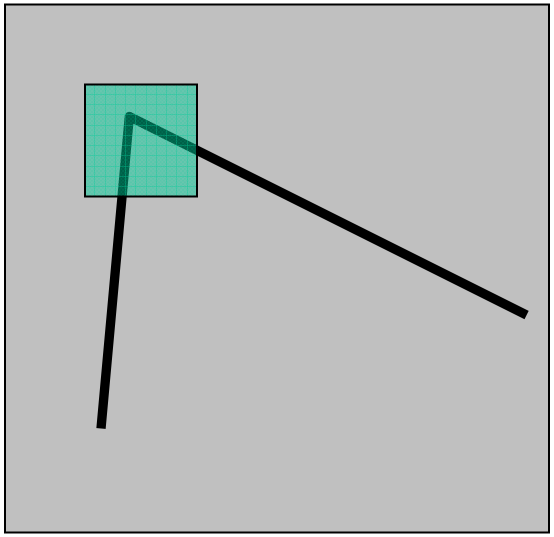
C.Harris, M.Stephens. "A Combined Corner and Edge Detector". 1988



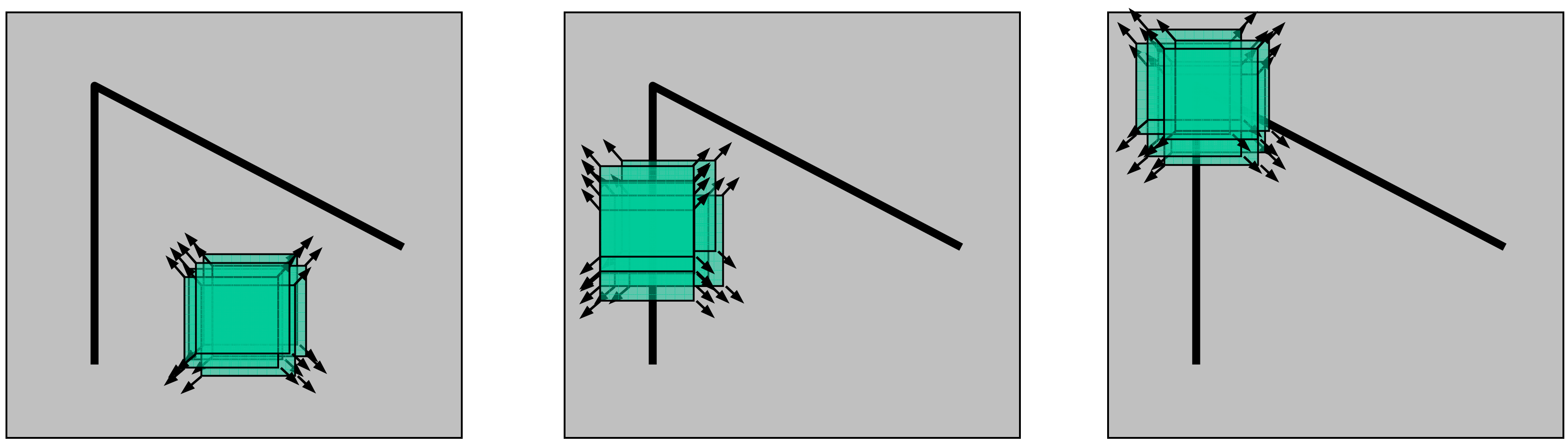
The Basic Idea

We should easily recognize the point by looking through a small window

Shifting a window in *any direction* should give *a large change* in intensity



Harris Detector: Basic Idea



"flat" region:
no change in
all directions

"edge":
no change along
the edge direction

"corner":
significant change
in all directions

Harris Detector: Mathematics

Change of intensity for the shift [u,v]:

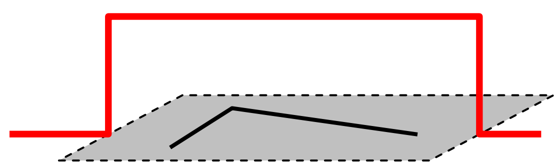
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

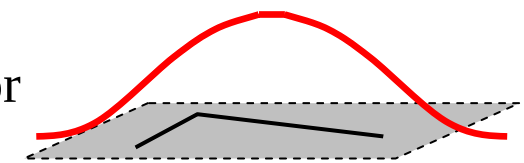
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Harris Detector: Mathematics

For small shifts $[u,v]$ we have a *bilinear* approximation:

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

where M is a 2×2 matrix computed from image derivatives:

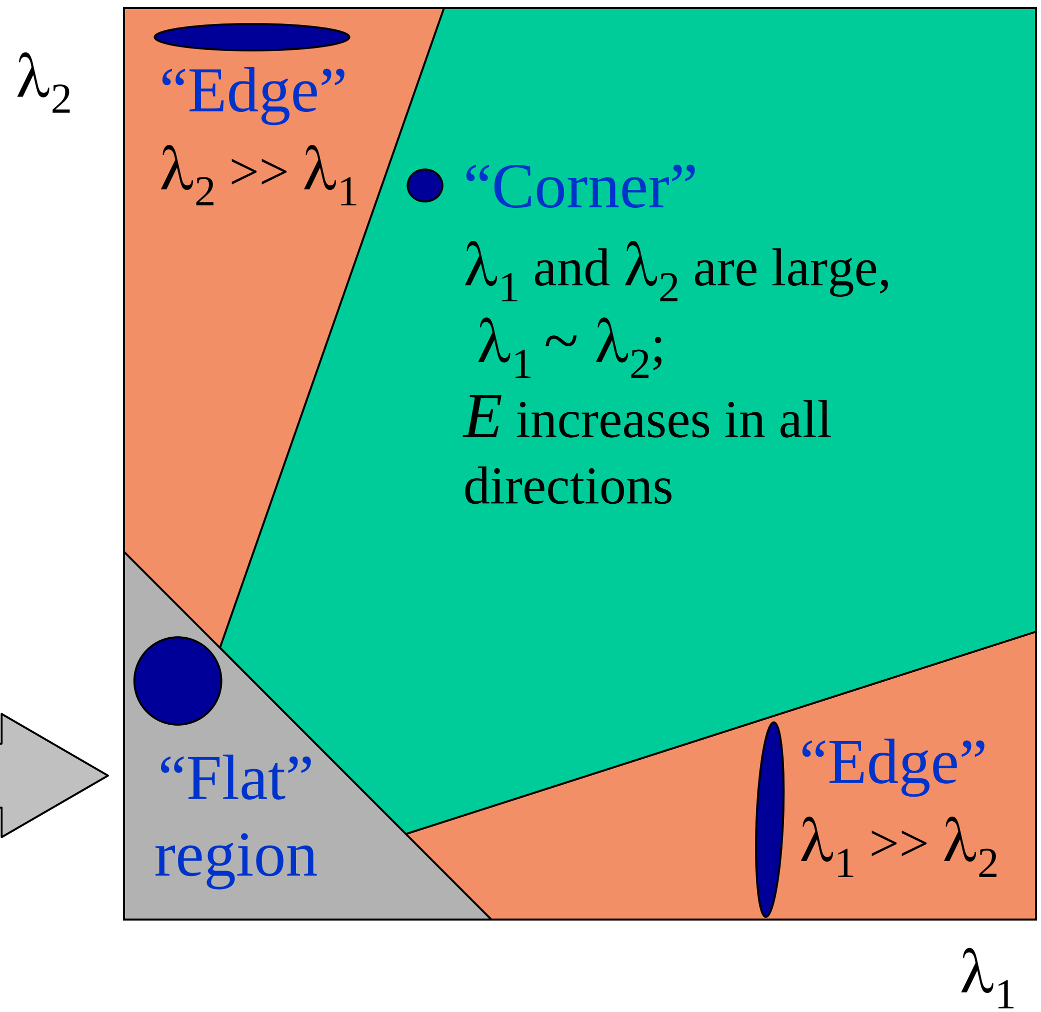
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

Harris Detector: Mathematics

Classification of image points using eigenvalues of *M*:

λ_1 and λ_2 are small;
E is almost constant in all directions



Harris Detector: Mathematics

Measure of corner response:

$$R = \frac{\det M}{\operatorname{Trace} M}$$

$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

Harris Detector

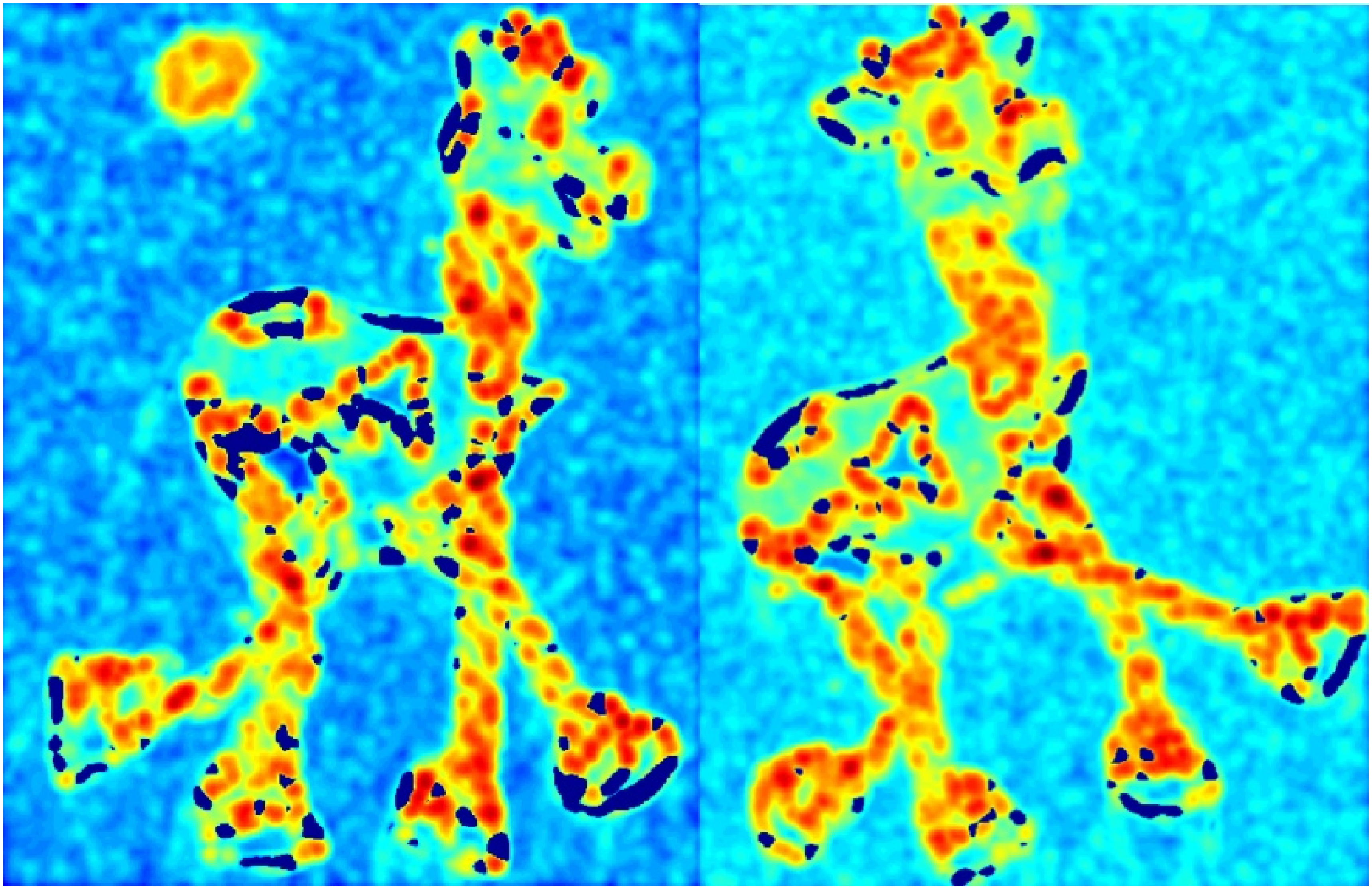
The Algorithm:

- Find points with large corner response function R (R > threshold)
- Take the points of local maxima of R

Harris Detector: Workflow

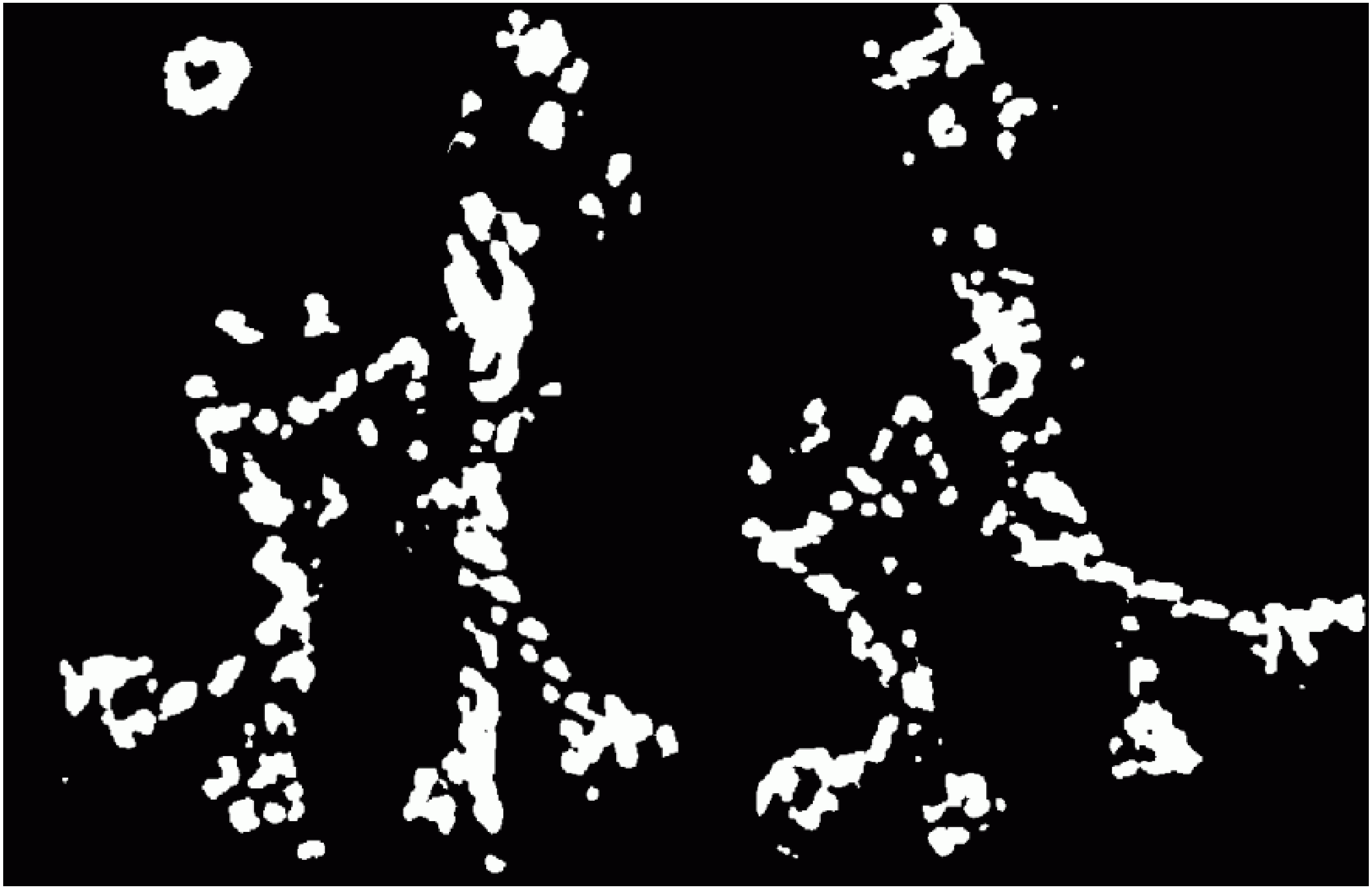
Harris Detector: Workflow

Compute corner response R



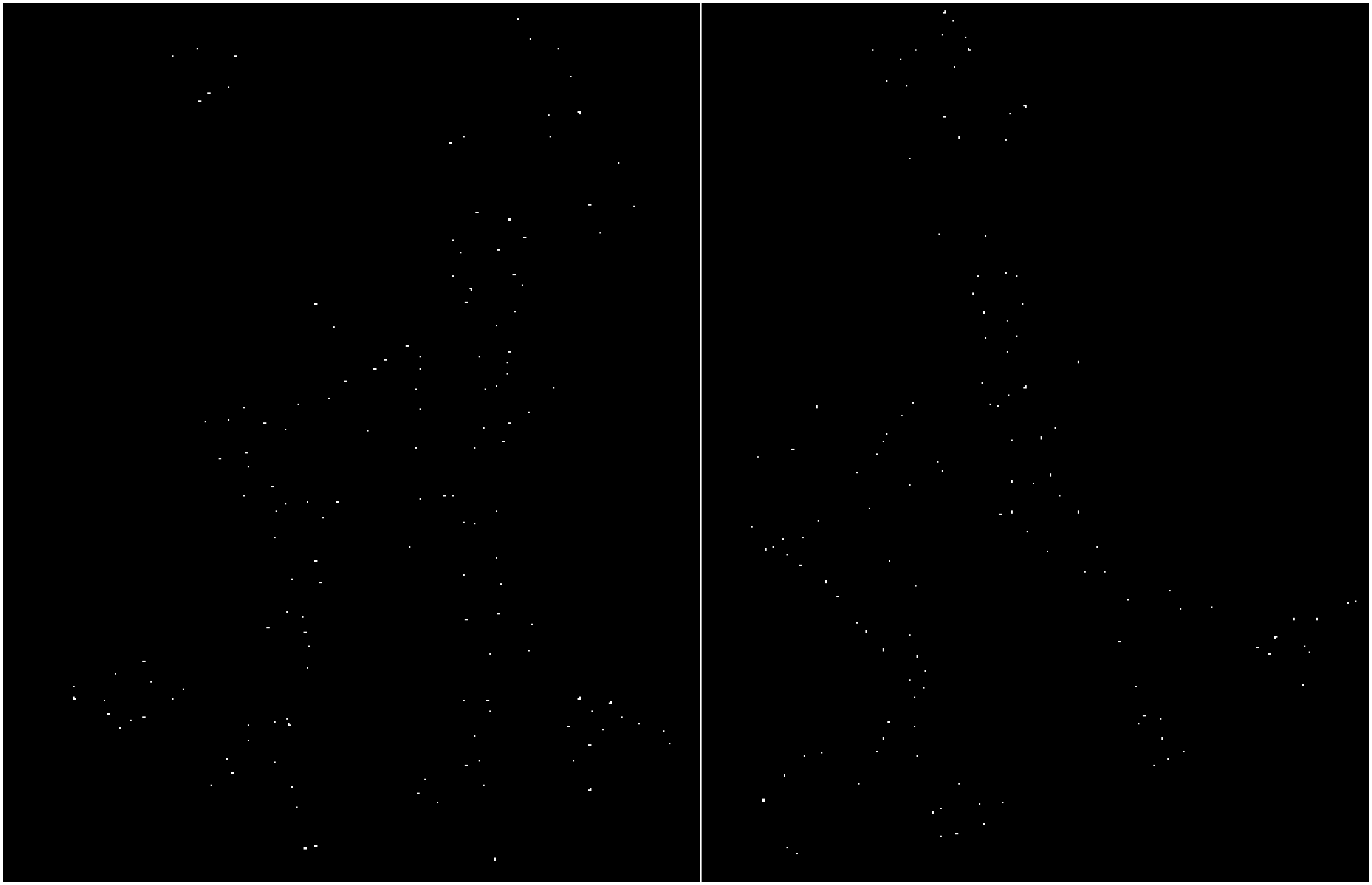
Harris Detector: Workflow

Find points with large corner response: R>threshold



Harris Detector: Workflow

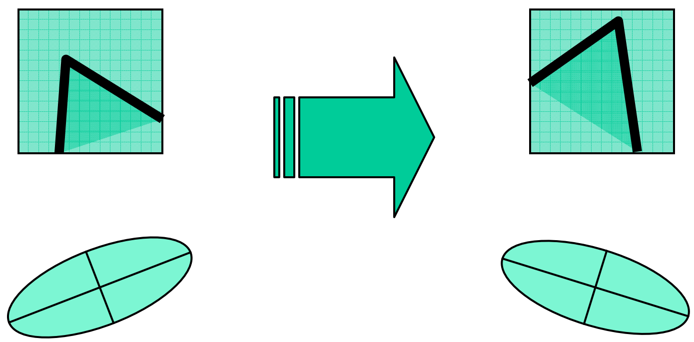
Take only the points of local maxima of R



Harris Detector: Workflow

Harris Detector: Some Properties

Rotation invariance



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

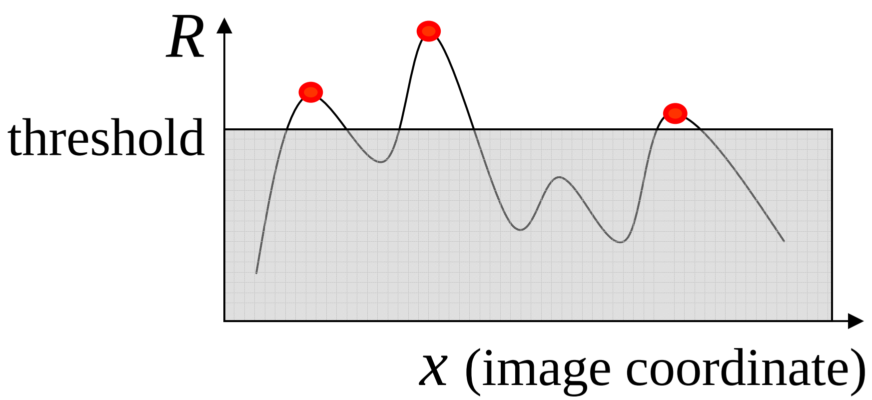
Corner response R is invariant to image rotation

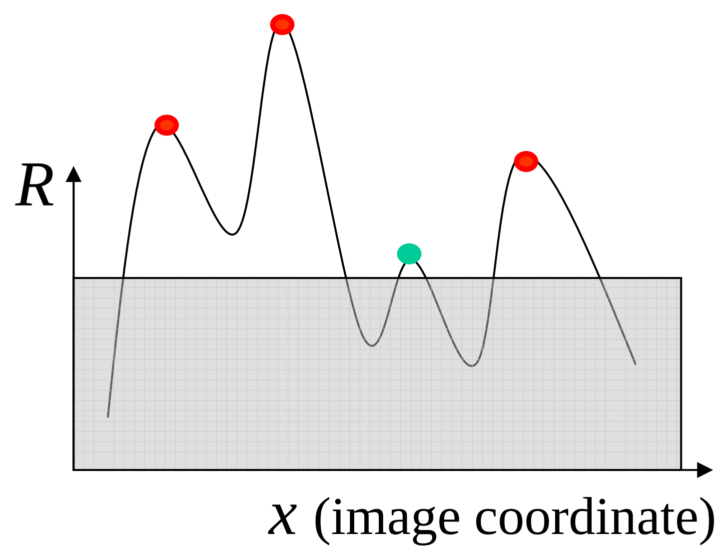
Harris Detector: Some Properties

Partial invariance to *affine intensity* change

✓ Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

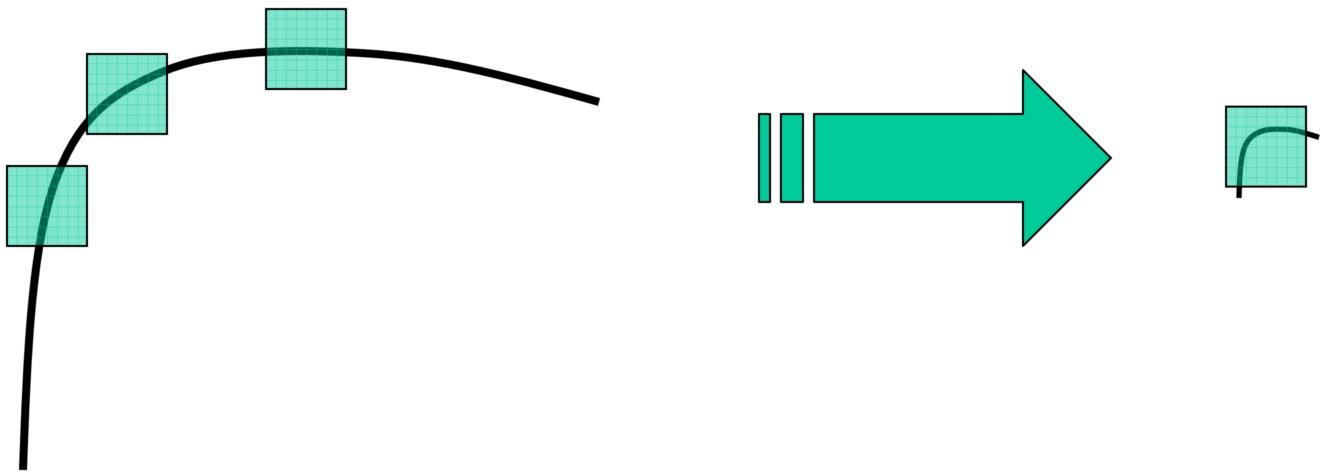
✓ Intensity scale: $I \rightarrow a\, I$





Harris Detector: Some Properties

But: non-invariant to *image scale*!

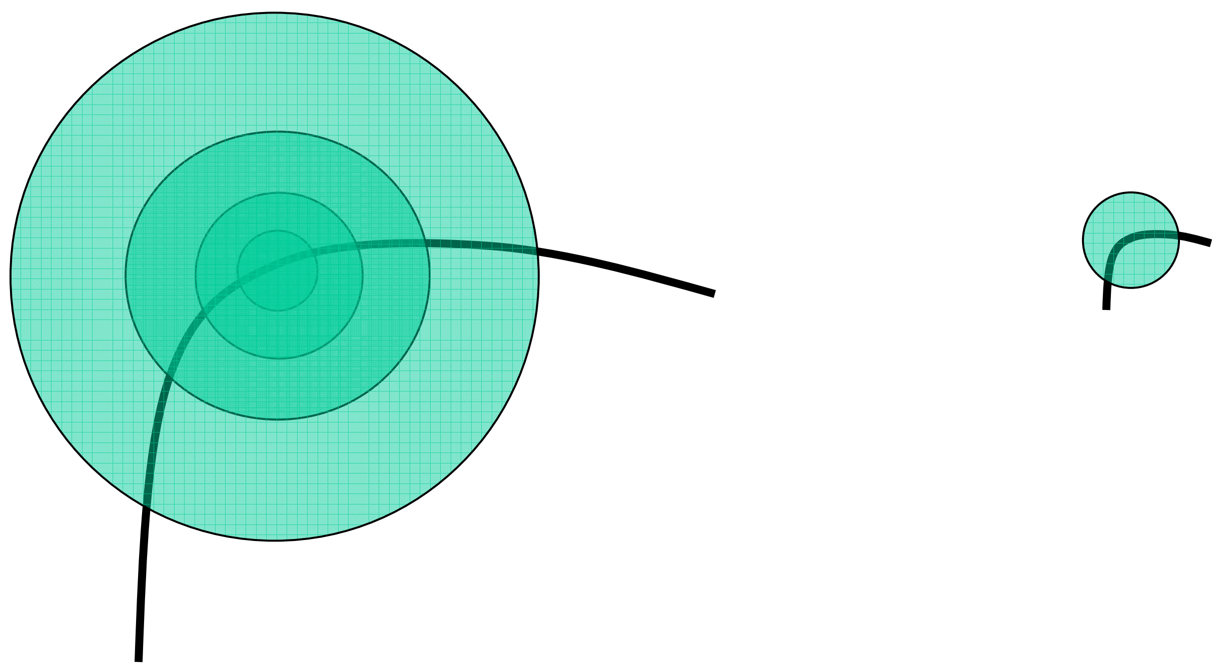


All points will be
classified as edges

Corner !

Scale Invariant Detection

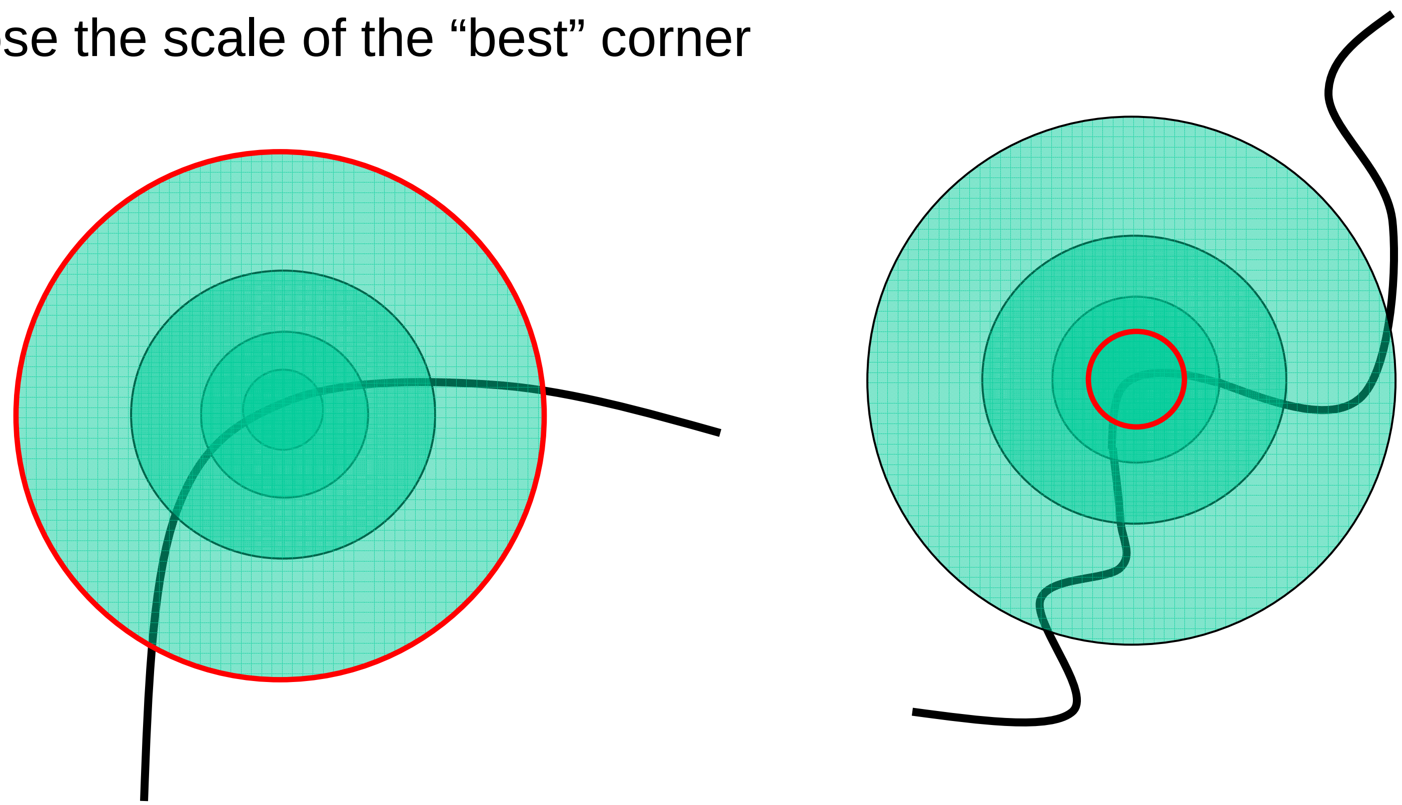
Consider regions (e.g. circles) of different sizes around a point
Regions of corresponding sizes will look the same in both images



Scale Invariant Detection

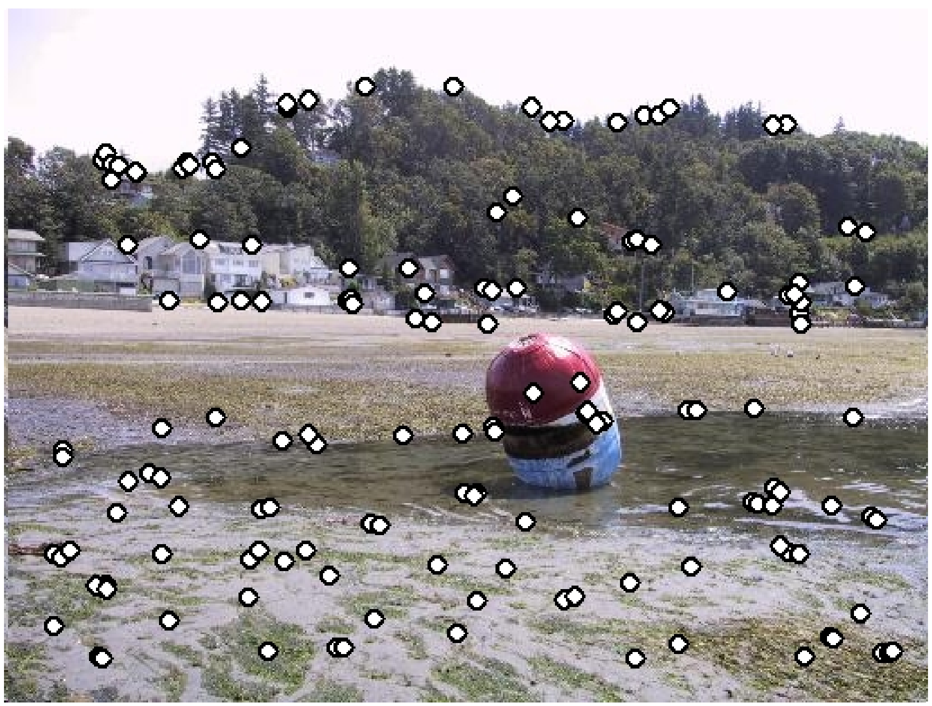
The problem: how do we choose corresponding circles *independently* in each image?

Choose the scale of the "best" corner



Feature selection

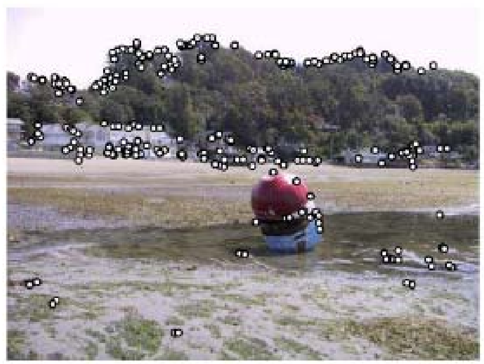
Distribute points evenly over the image



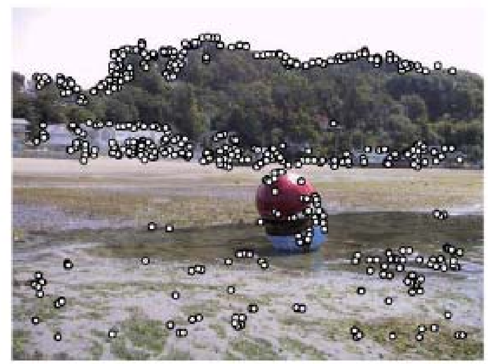
Adaptive Non-maximal Suppression

Desired: Fixed # of features per image

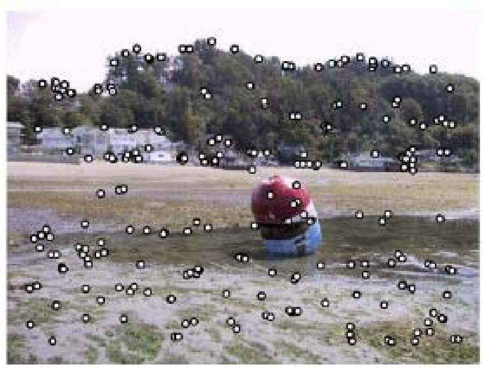
- Want evenly distributed spatially…
- Sort ponts by non-maximal suppression radius [Brown, Szeliski, Winder, CVPR'05]



(a) Strongest 250



(b) Strongest 500



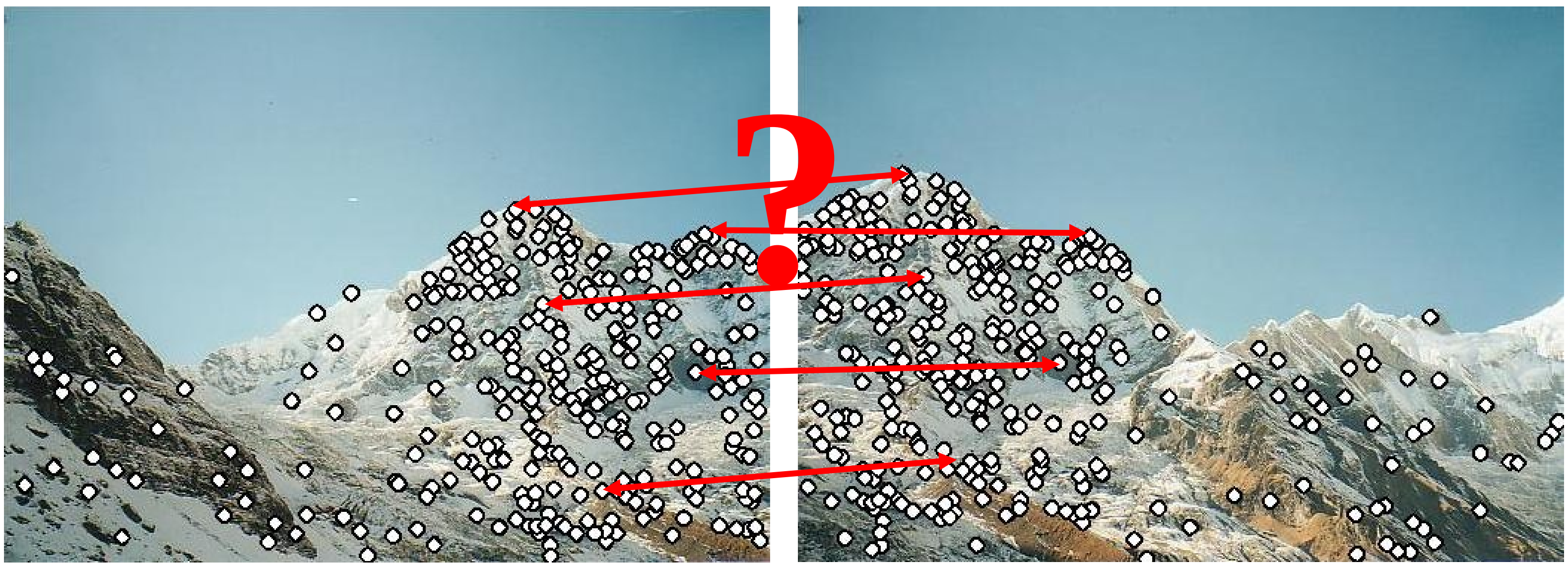
(c) ANMS 250, $r = 24$

(d) ANMS 500, $r = 16$

Feature descriptors

We know how to detect points

Next question: **How to match them?**



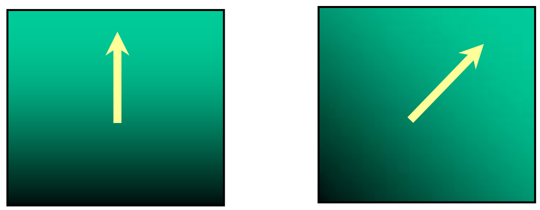
Point descriptor should be:

1. Invariant
2. Distinctive

Descriptors Invariant to Rotation

Find local orientation

Dominant direction of gradient



- Extract image patches relative to this orientation

Multi-Scale Oriented Patches

Interest points

- Multi-scale Harris corners
- Orientation from blurred gradient
- Geometrically invariant to rotation

Descriptor vector

- Bias/gain normalized sampling of local patch (8x8)
- Photometrically invariant to affine changes in intensity

[Brown, Szeliski, Winder, CVPR’2005]

Multi-Scale Oriented Patches

Interest points

- Multi-scale Harris corners
- Orientation from blurred gradient
- Geometrically invariant to rotation

Descriptor vector

- Bias/gain normalized sampling of local patch (8x8)
- Photometrically invariant to affine changes in intensity

[Brown, Szeliski, Winder, CVPR'2005]

Descriptor Vector

Orientation = blurred gradient

Rotation Invariant Frame

- Scale-space position (x, y, s) + orientation (θ)

Detections at multiple scales

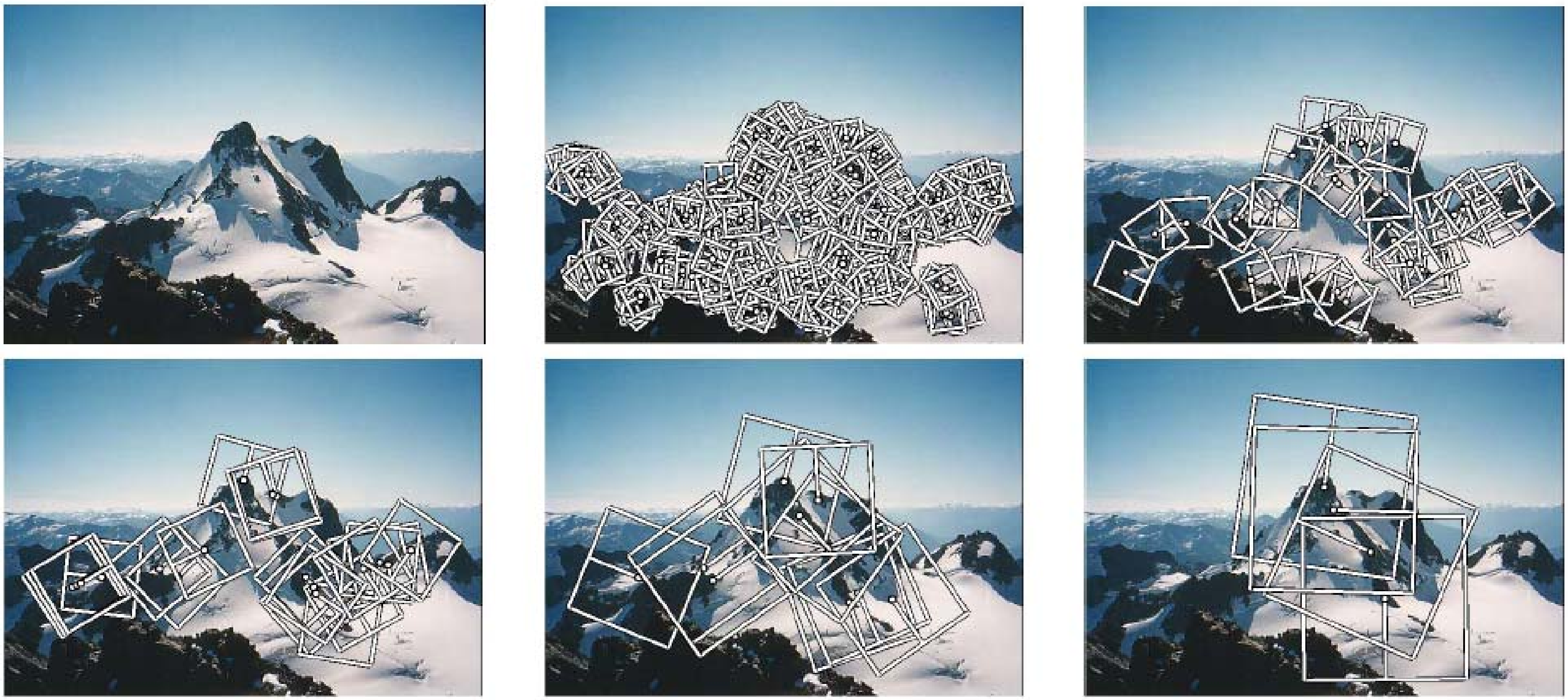


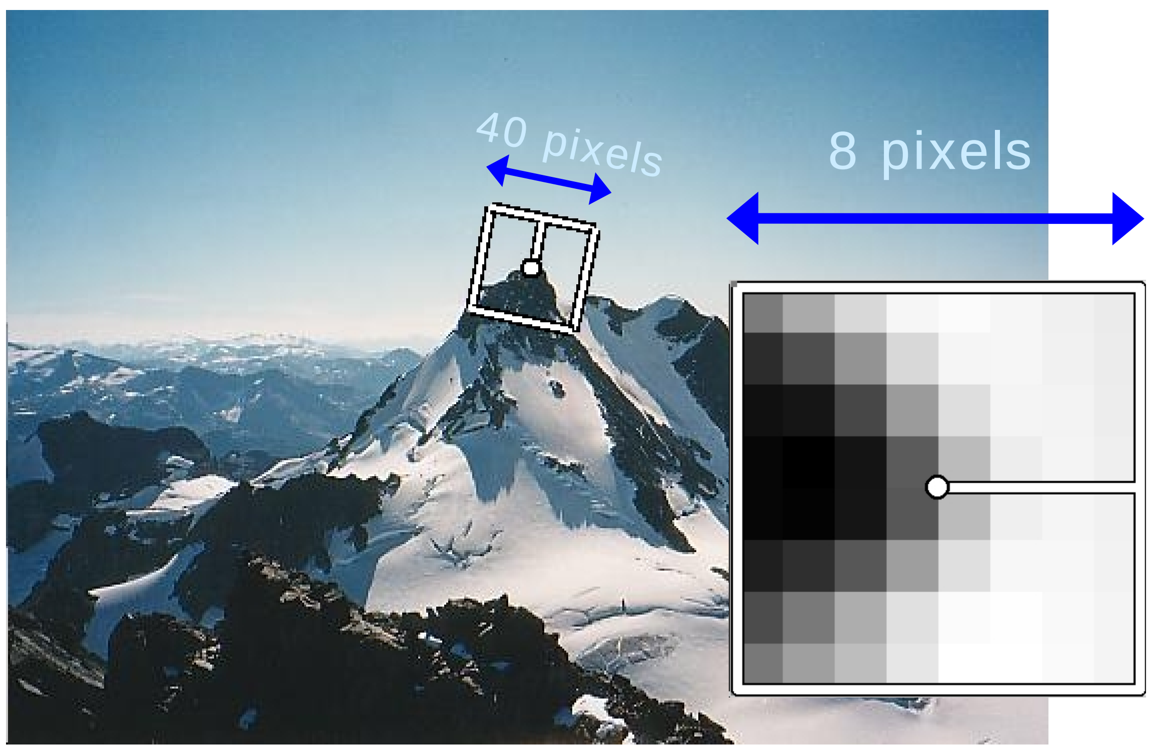
Figure 1. Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels from one of the Matier images. The boxes show the feature orientation and the region from which the descriptor vector is sampled.

MOPS descriptor vector

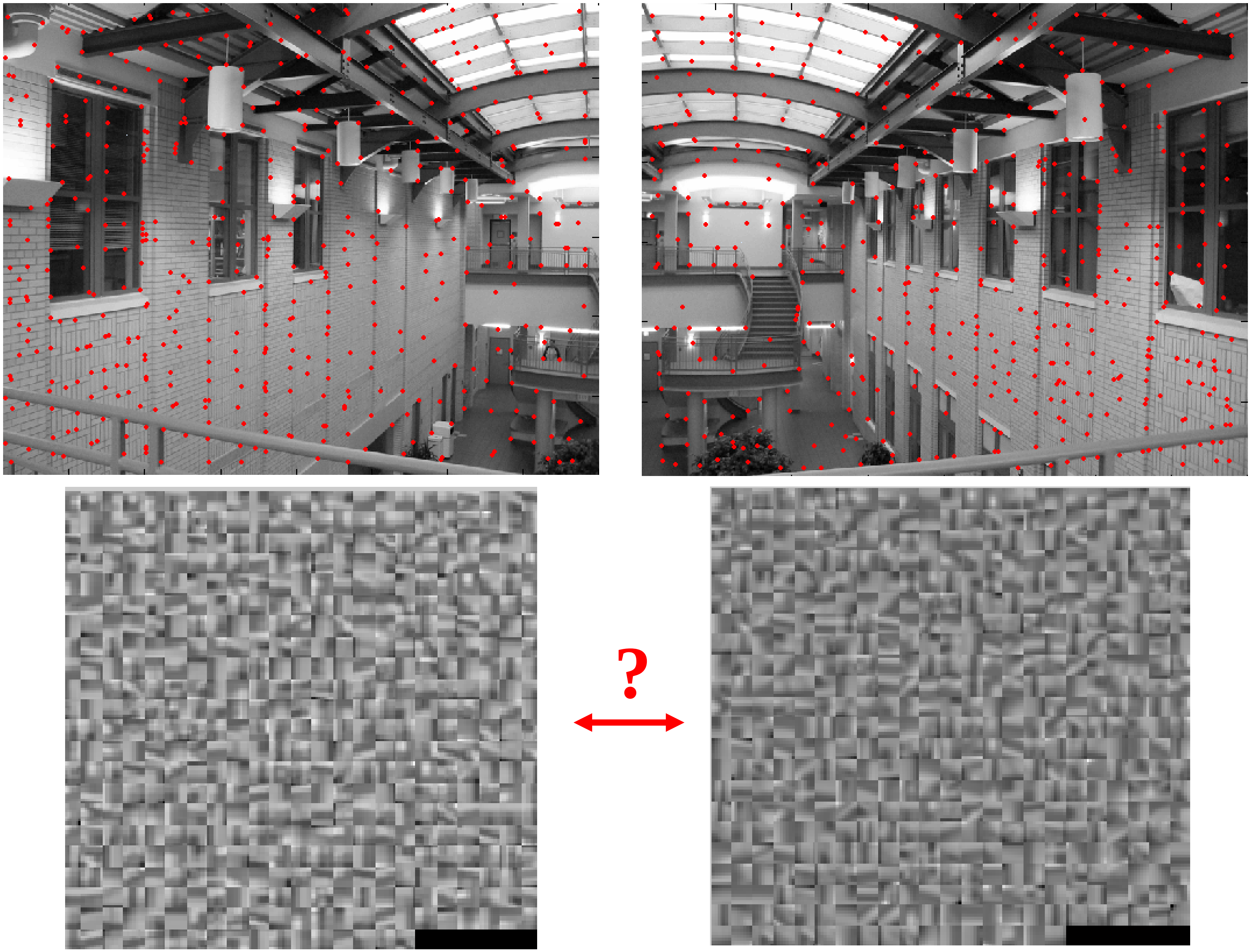
8x8 oriented patch

- Sampled at 5 x scale

Bias/gain normalisation: $I' = (I - \mu)/\sigma$



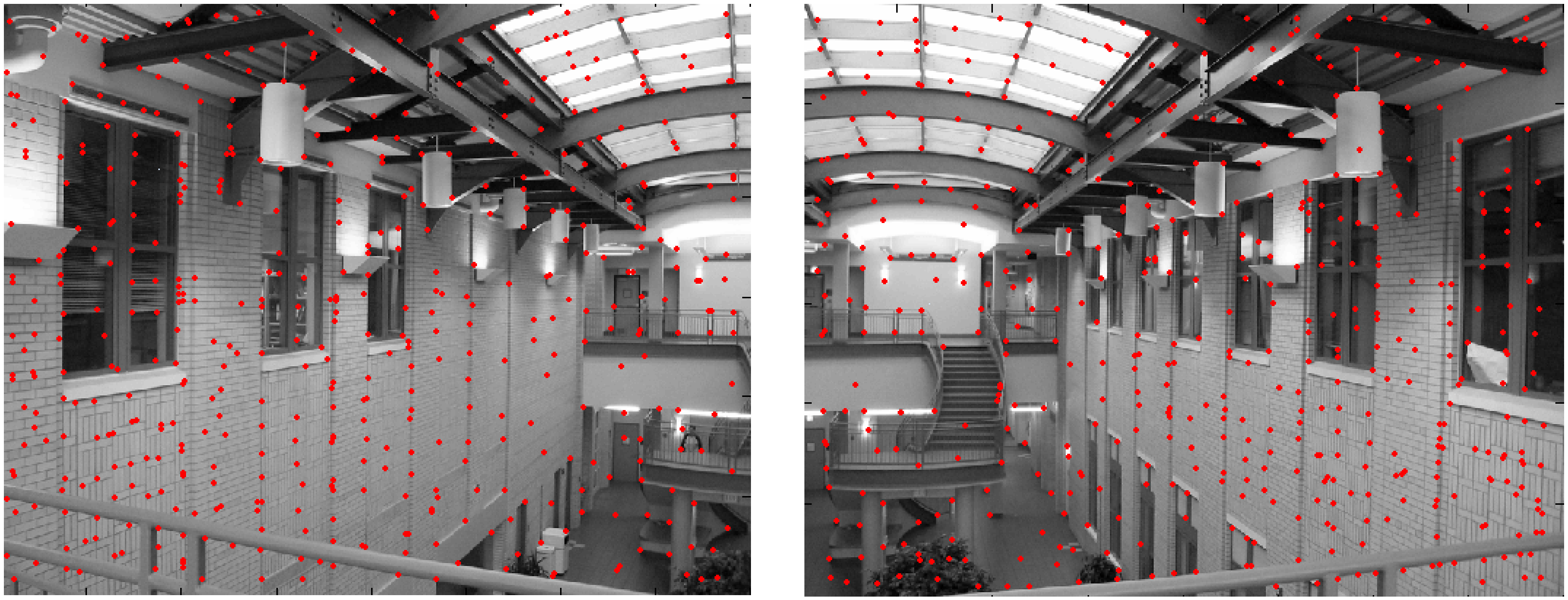
Feature matching

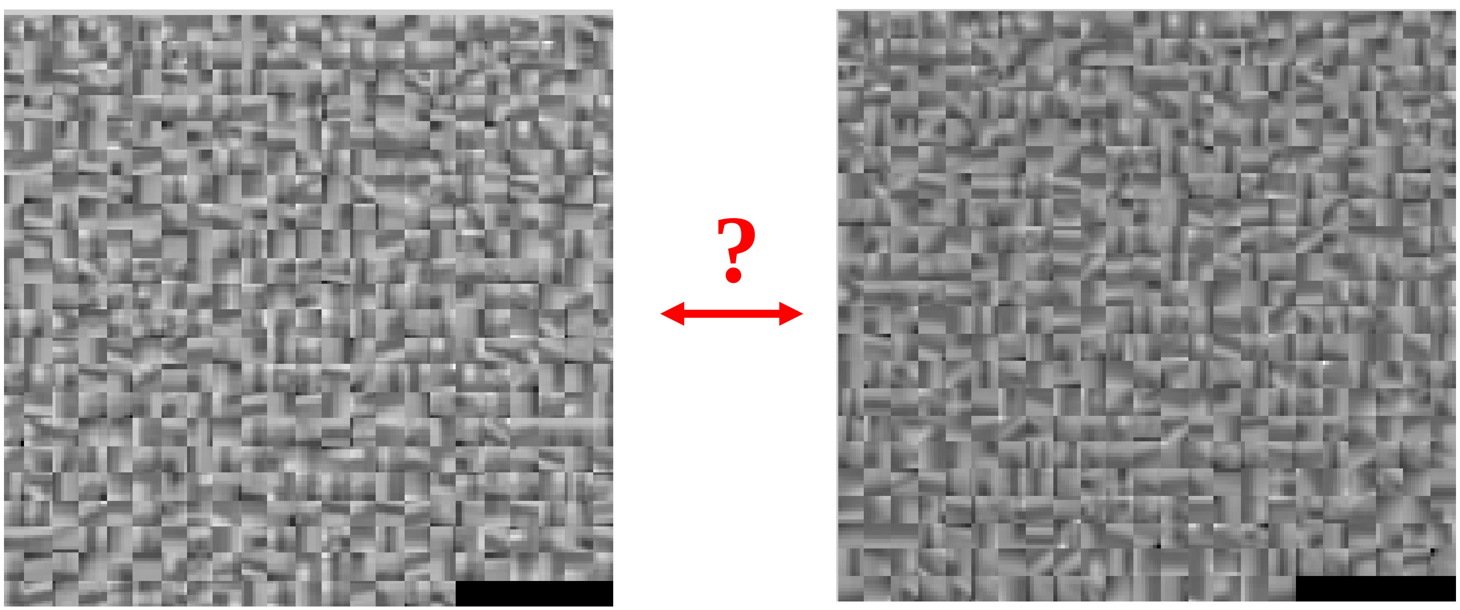


Feature matching

- Exhaustive search
  - for each feature in one image, look at *all* the other features in the other image(s)
- Hashing
  - compute a short descriptor from each feature vector, or hash longer descriptors (randomly)
- Nearest neighbor techniques
  - *k*-trees and their variants

What about outliers?



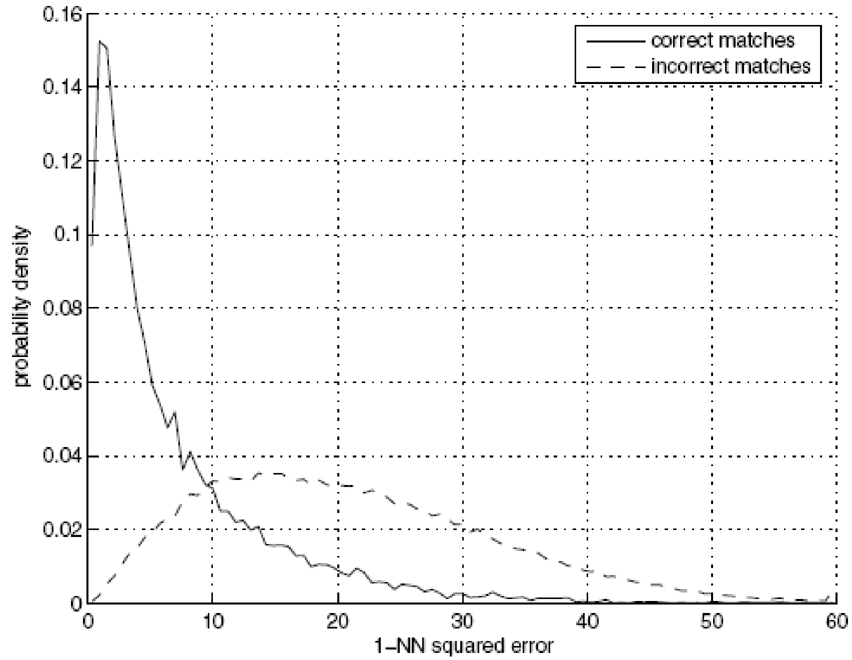


Feature-space outlier rejection

Let's not match all features, but only these that have "similar enough" matches?

How can we do it?

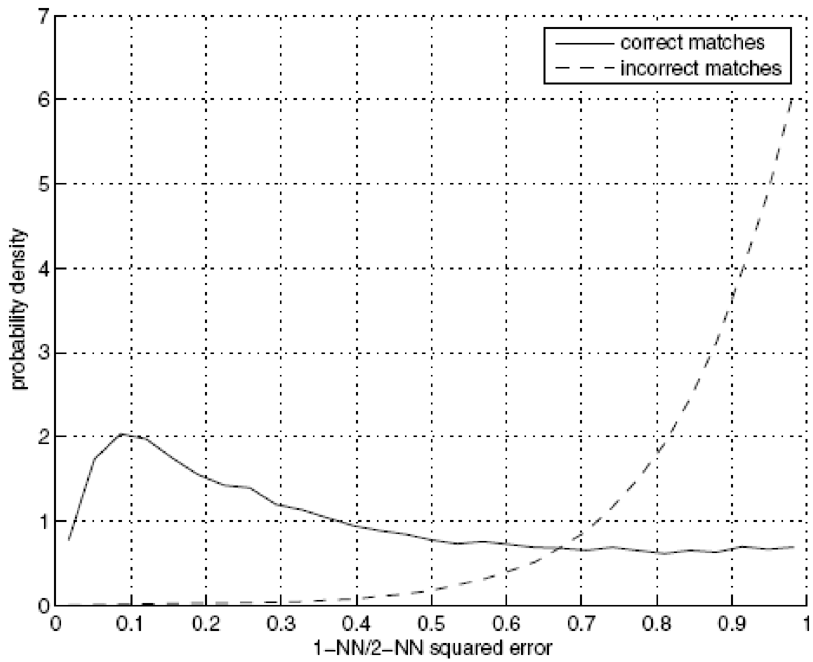
- SSD(patch1,patch2) < threshold
- How to set threshold?



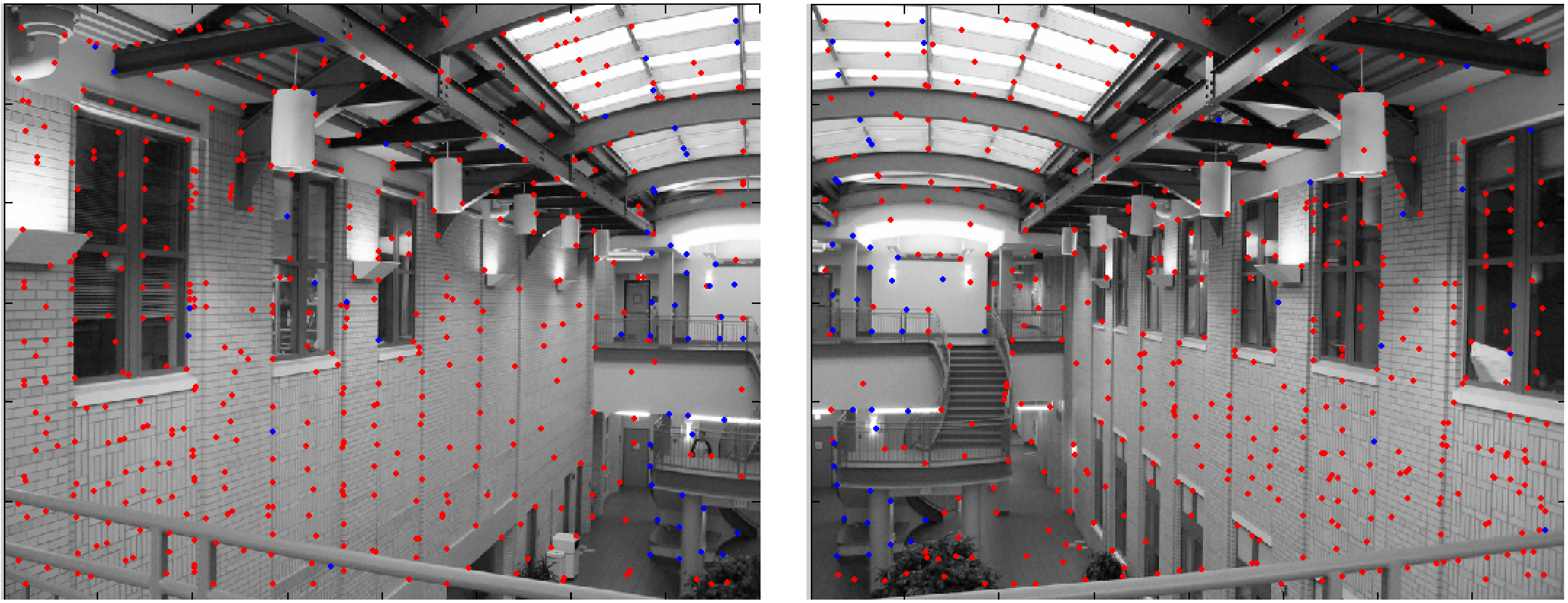
Feature-space outlier rejection

A better way [Lowe, 1999]:

- 1-NN: SSD of the closest match
- 2-NN: SSD of the <u>second-closest</u> match
- Look at how much better 1-NN is than 2-NN, e.g. 1-NN/2-NN
- That is, is our best match so much better than the rest?



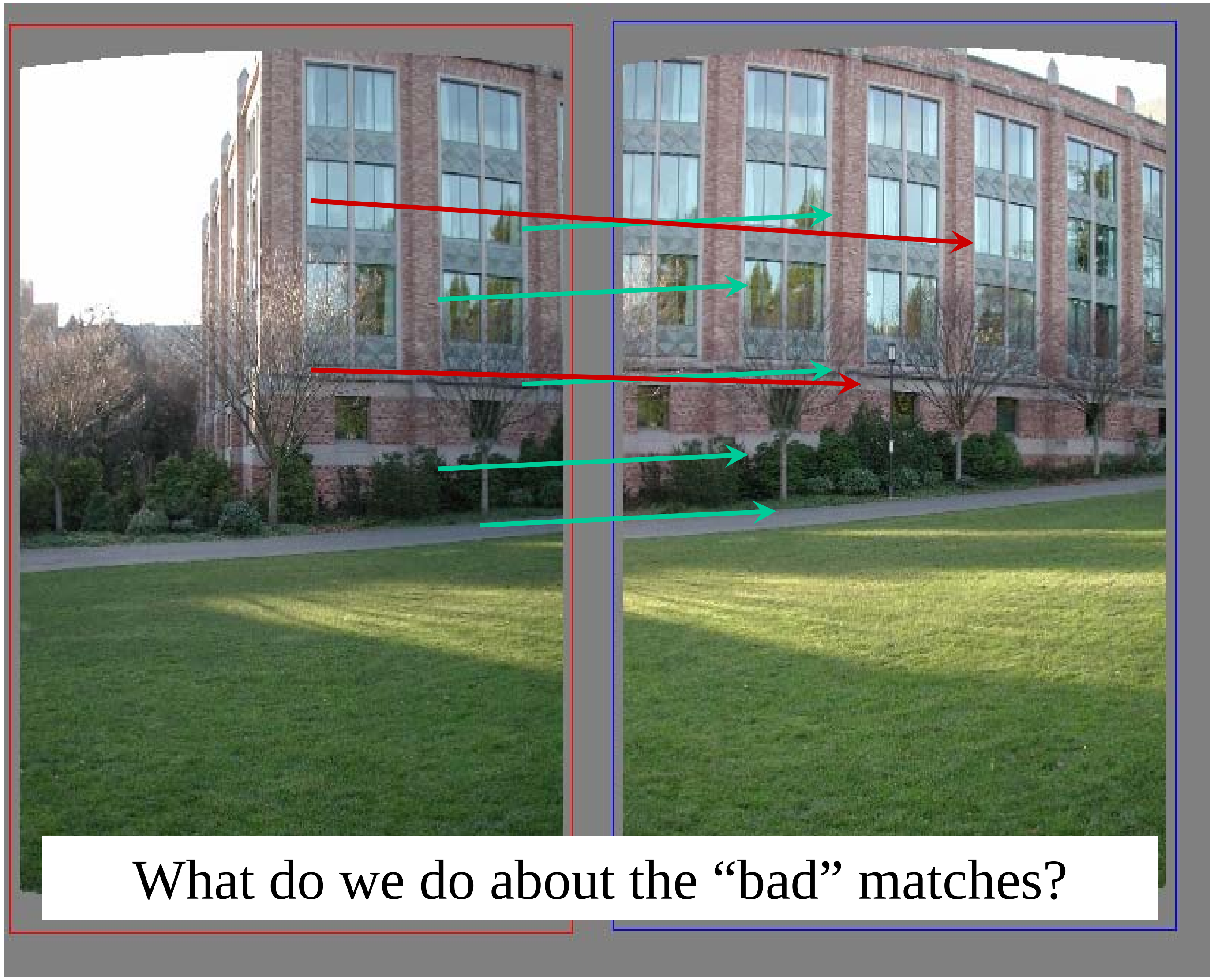
Feature-space outliner rejection



Can we now compute H from the blue points?

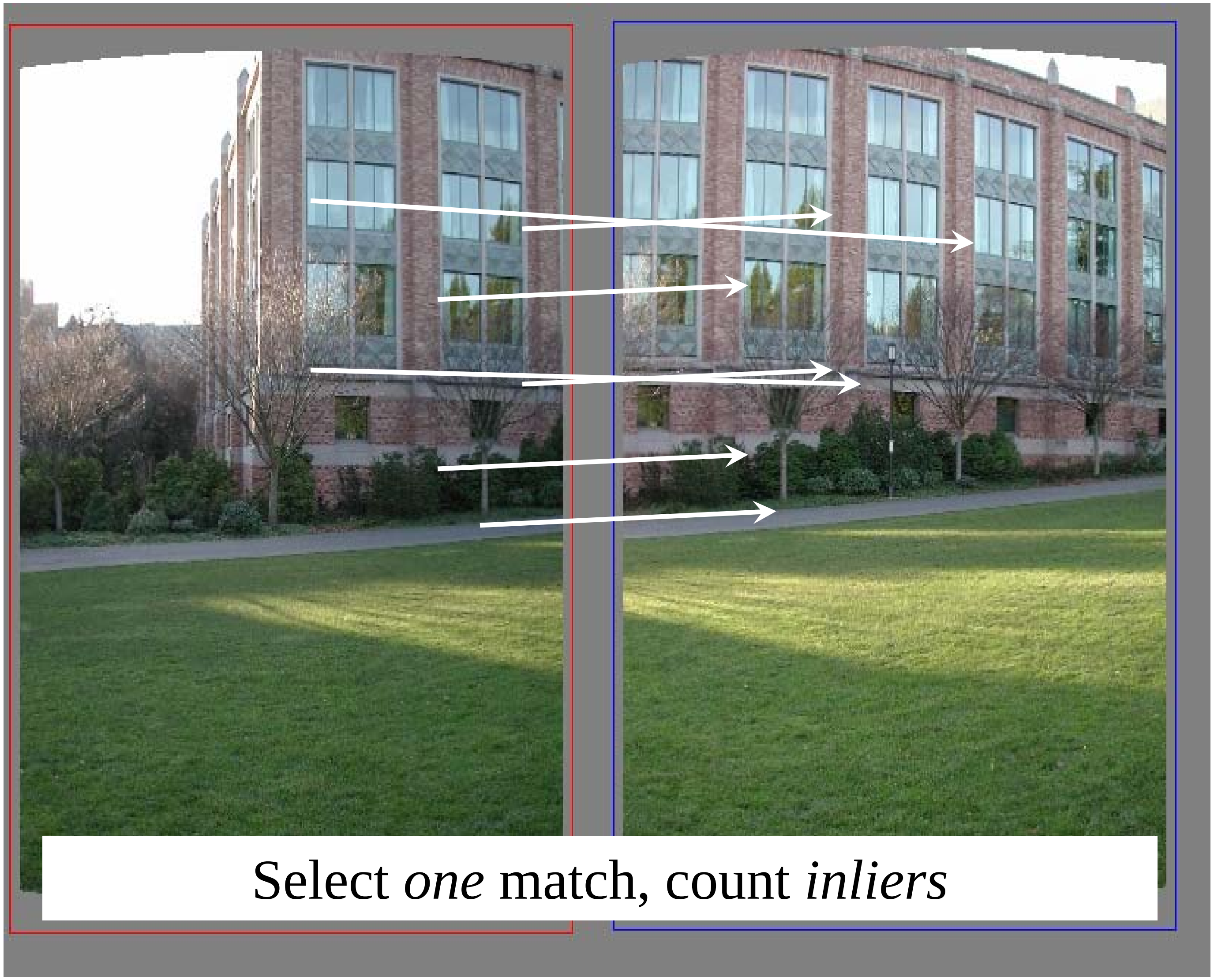
- No! Still too many outliers…
- What can we do?

Matching features



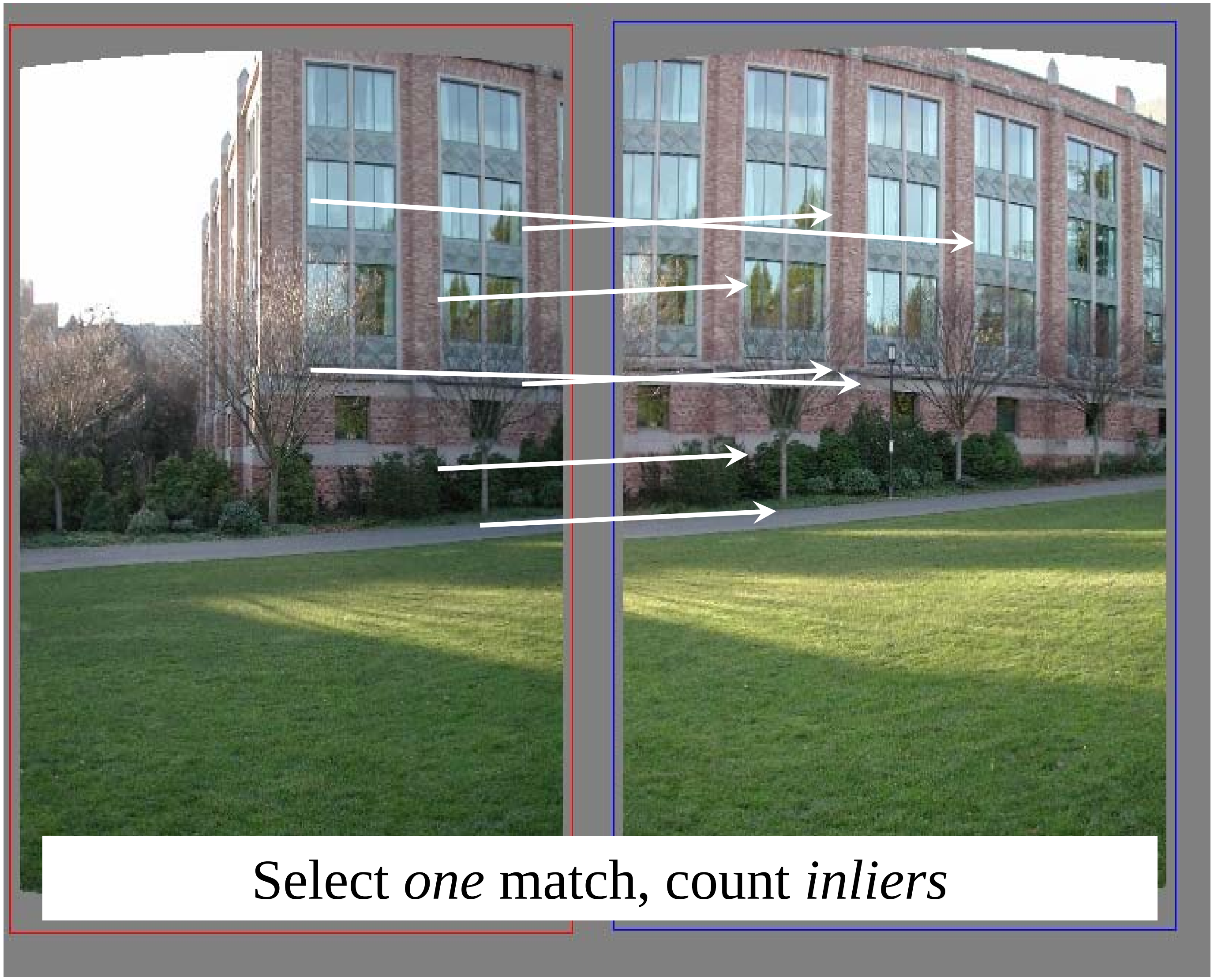
What do we do about the “bad” matches?

<u>RA</u>ndom <u>SA</u>mple <u>C</u>onsensus



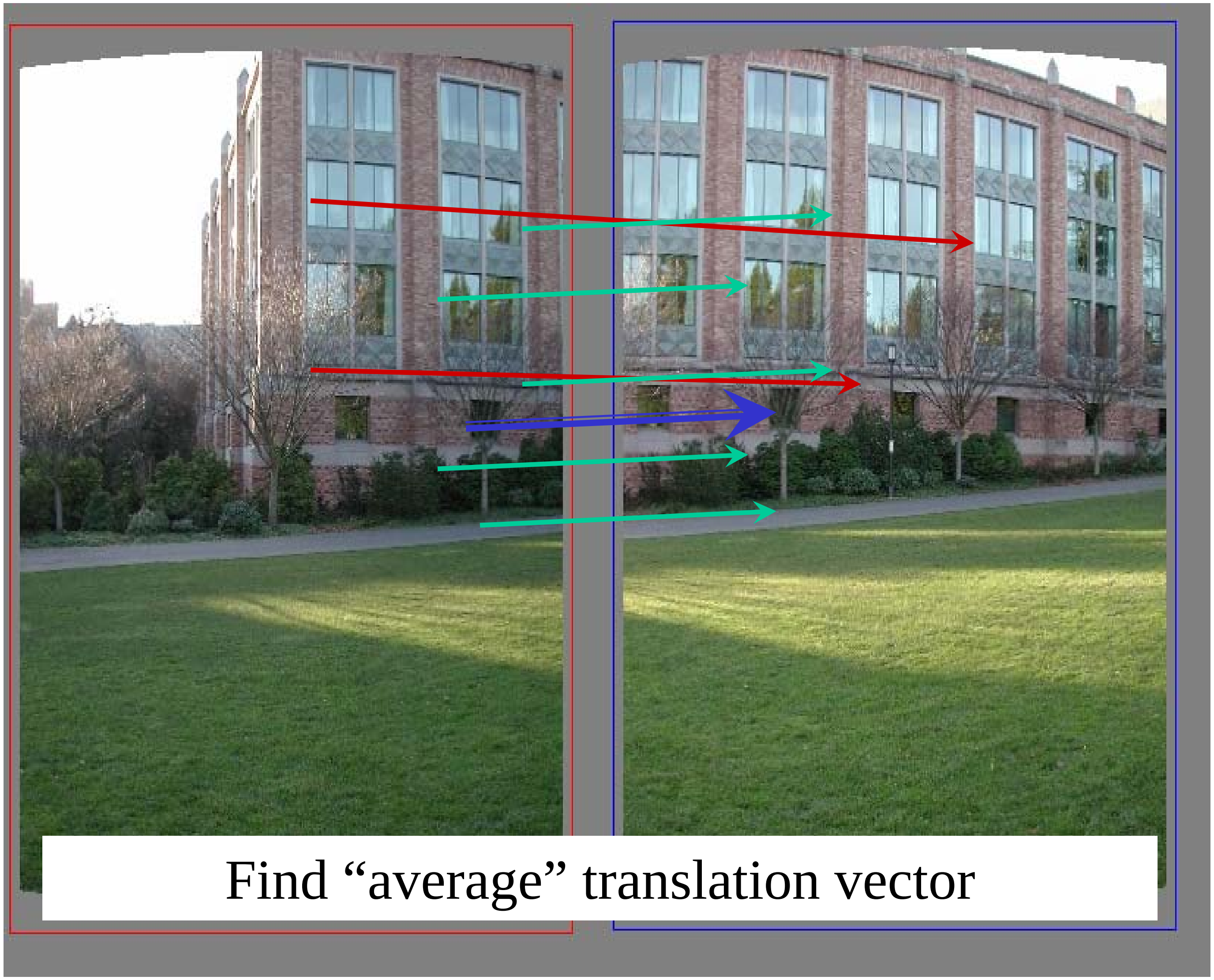
Select *one* match, count *inliers*

RAndom SAmple Consensus



Select *one* match, count *inliers*

Least squares fit



RANSAC for estimating homography

RANSAC loop:

1. Select four feature pairs (at random)
2. Compute homography H (exact)
3. Compute *inliers* where $SSD(p_i', \mathbf{H}\, p_i) < \varepsilon$
4. Keep largest set of inliers
5. Re-compute least-squares H estimate on all of the inliers

RANSAC