POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

3D Structure from Visual Motion

M. Matteucci, *Vincenzo Caglioti,*
Marco Marcon, Domenico G. Sorrenti

matteucci@elet.polimi.it
Dipartimento di Elettronica e Informazione, Politecnico di Milano
Artificial Intelligence and Robotics Lab

Disclaimer …

These slides have been heavily “inspired” by the teaching material kindly provided by Davide Migliore:

Who has been heavily “inspired” by many colleagues (in order of appearance)

- Lowe, Dellaert, Harris, Stephens, Schmid, Mohrm, Bauckhage, Rosten, Drummond, Mikolajczyk, Burt, Adelson, Tuytelaars, Van Gool, Fischler, Bolles, Brown, …

Please refer to the ***original sources*** for a deeper analysis and further references on the topic …

POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

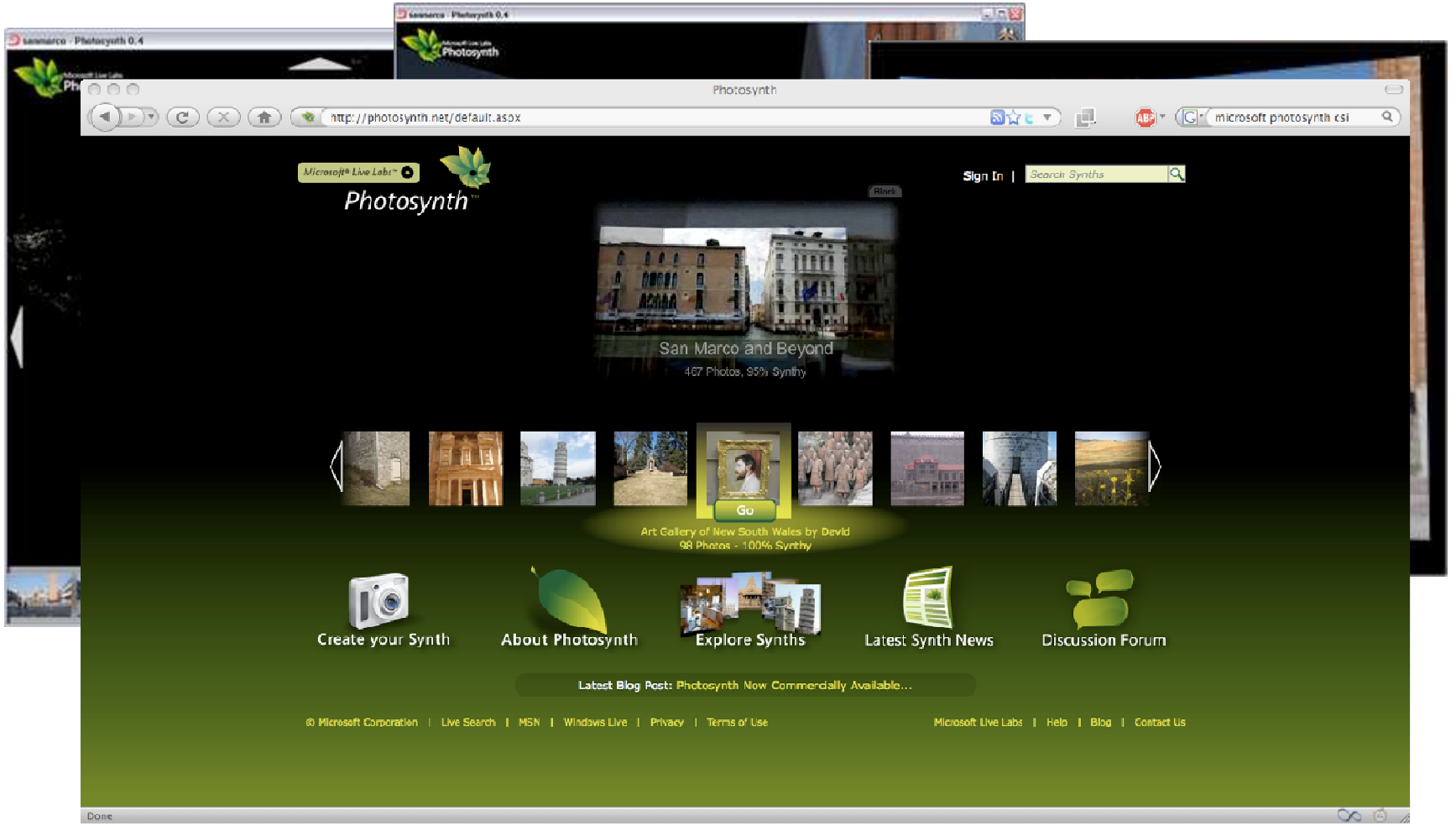
Feature Matching

Key idea: matching images captured from different poses without any human intervention

Feature = interesting piece of image

Science fiction?

No it is a commercial from Microsoft!

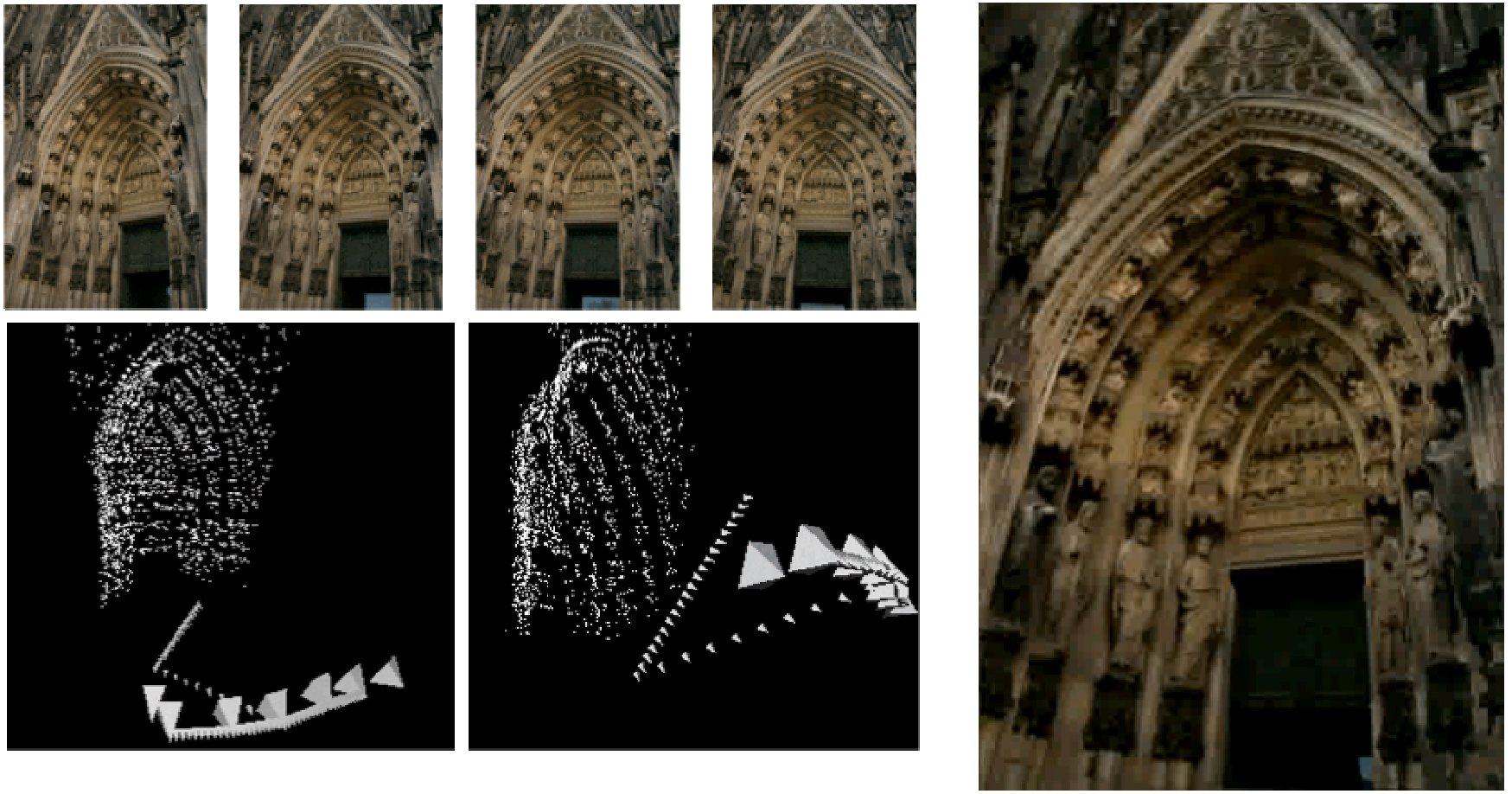


http://photosynth.net/

Step one: Object Recognition

Lowe, David G. (1999). "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision 2: 1150–1157. doi:10.1109/ICCV.1999.790410

Step Two: Structure From Motion



Images from prof. Dellaert Presentation

Other Examples: Panorama Stitch

http://hugin.sourceforge.net/

Other Examples: 3D Reconstruction

Summary

What is a feature?

How to find features (aka detectors)

- Harris
- Features from Accelerated Segment Test (FAST)
- Scale Invariant (SIFT)
- Affine Invariant (Harris Affine, MSER, …)

How to describe a feature (aka descriptors)

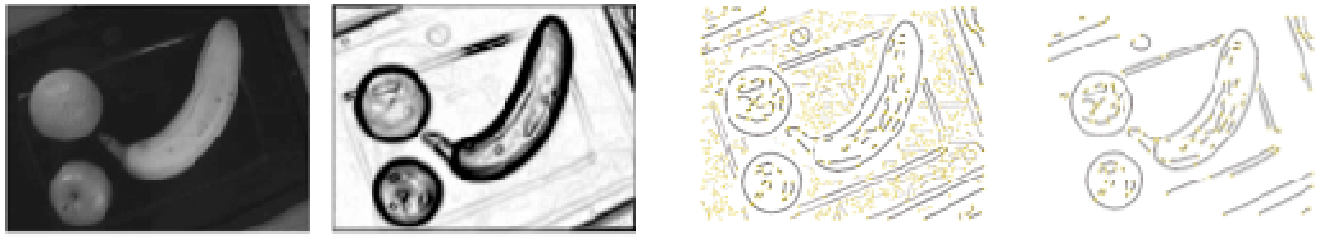
- Patch
- Scale Invariant Feature Transform (SIFT)

How to make robust the matching

- RANdom SAmple Consensus (RANSAC)

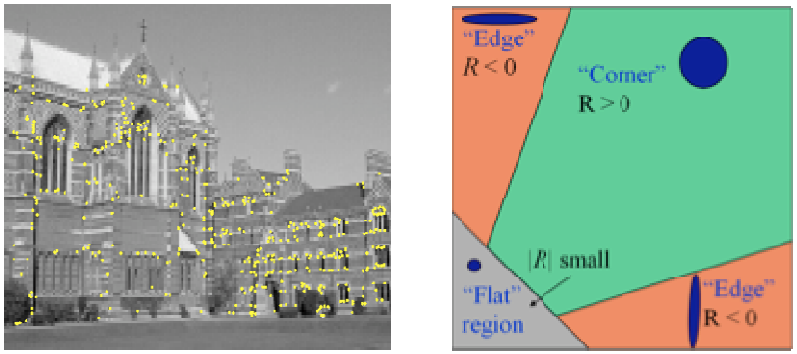
What is a feature?

Edges

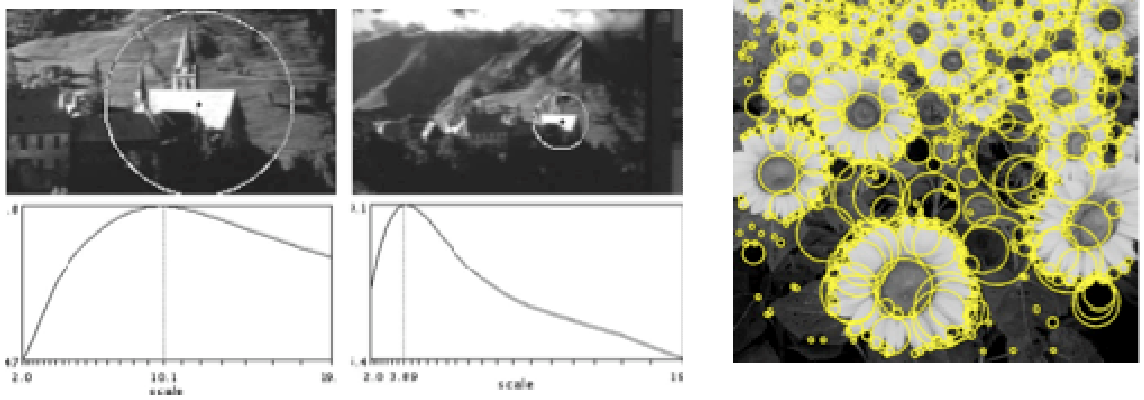


(a) Original image (b) Edge strength $|\nabla S|$ (c) Non-maximal suppression (d) Thresholding

Corners



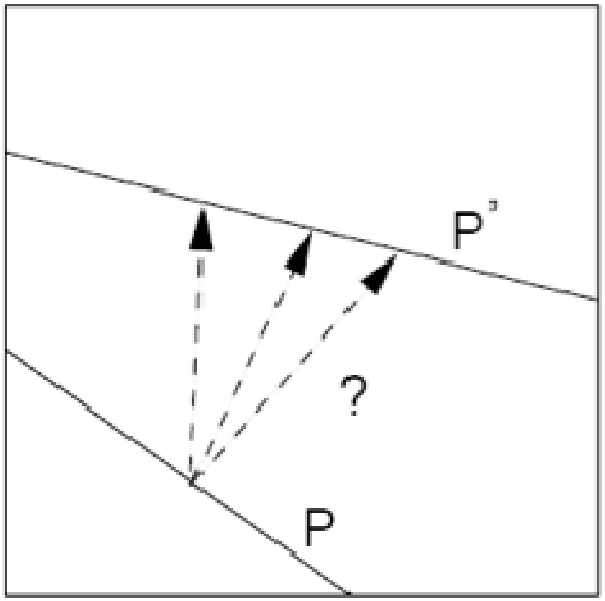
Local regions (Blobs)



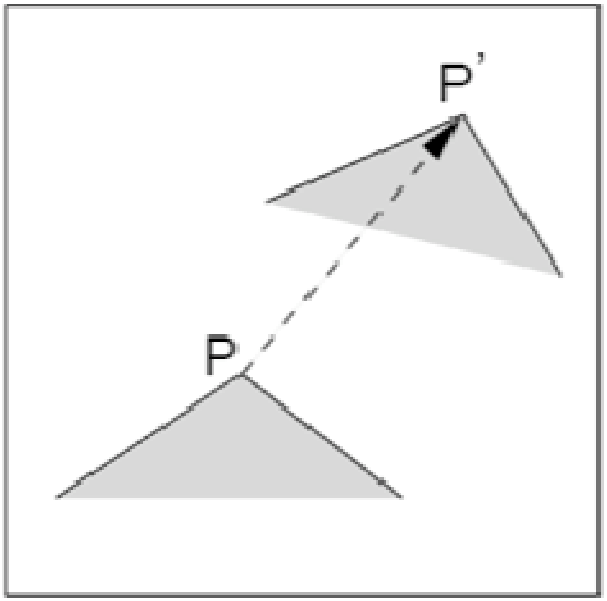
Why Corners?

We need to track image locations (points/pixels) between frames

- Edges are ambiguous due to the aperture problem (i.e., we can only measure the motion of a point normal to edge)
- Corners do not suffer this problem (i.e., easily found by correlation)



Edge



Corner

Summary

What is a feature?

How to find features (aka detectors)

- Harris
- Features from Accelerated Segment Test (FAST)
- Scale Invariant (SIFT)
- Affine Invariant (Harris Affine, MSER, …)

How to describe a feature (aka descriptors)

- Patch
- Scale Invariant Feature Transform (SIFT)

How to make robust the matching

- RANdom SAmple Consensus (RANSAC)

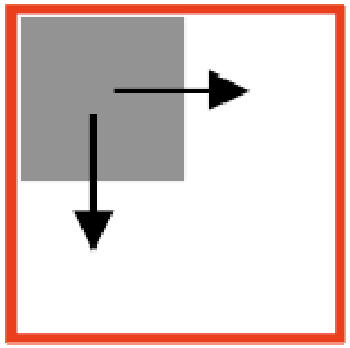
Corner Detection

Basic idea**:** points/pixels where two edges meet

- High gradient in two directions

However: "*Cornerness*" is undefined at a single pixel

- Look at the gradient behavior over a small window



Categories image windows based on gradient statistics

- Constant: Little or no brightness change
- Edge: Strong brightness change in single direction
- Flow: Parallel stripes
- Corner/spot: Strong brightness changes in orthogonal directions

What about the gradient?

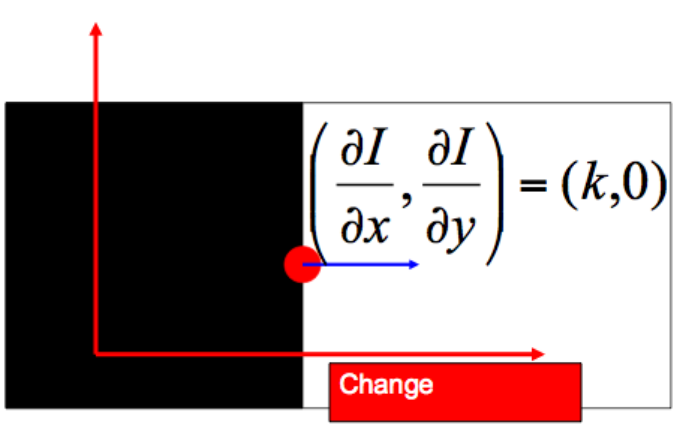
Think of image intensities as a function

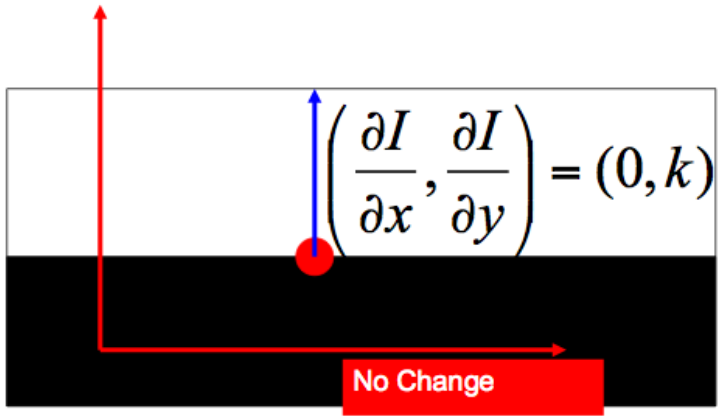
$$\mathbf{I}(x, y)$$

The gradient of image is a vector field as for a normal 2-D function:

$$\nabla \mathbf{I} = \left(\frac{\partial \mathbf{I}}{\partial x}, \frac{\partial \mathbf{I}}{\partial y}\right)^T = (\mathbf{I}_x, \mathbf{I}_y)^T$$

Edges: "places", orthogonal to gradient direction, with high gradient magnitude





Discrete Implementation of Image Gradient

Sobel mask operators

Gx = Sobel x * I
Gy = Sobel y * I

1	0	-1
2	0	-2
1	0	-1

Sobel x

1	2	1
0	0	0
-1	-2	-1

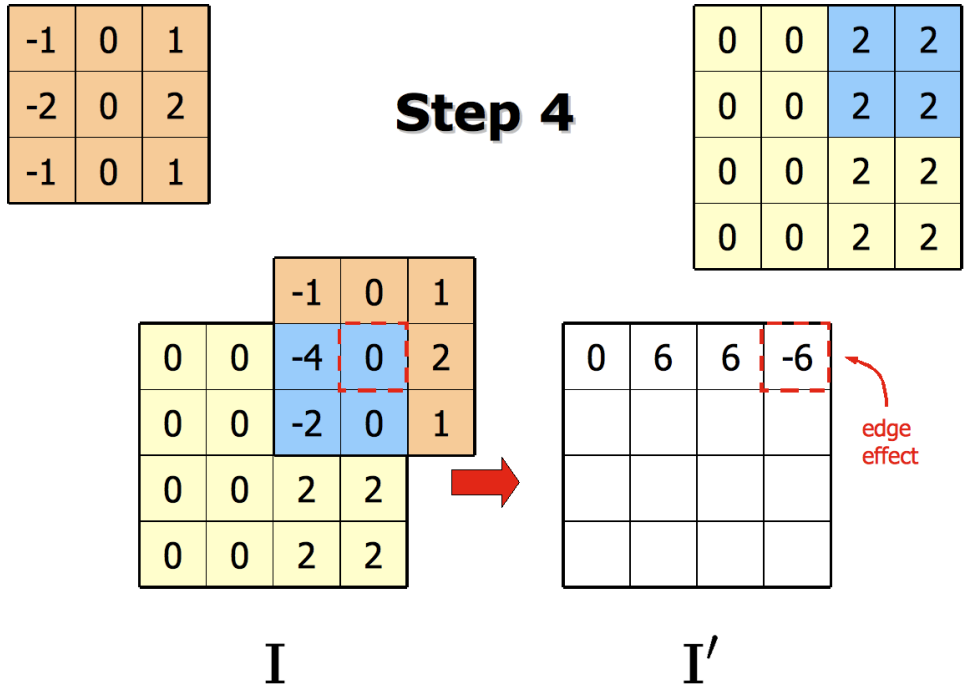
Sobel y

Magnitudo

$$\mathbf{G} = \sqrt{\mathbf{G}_x^{\,2} + \mathbf{G}_y^{\,2}}$$

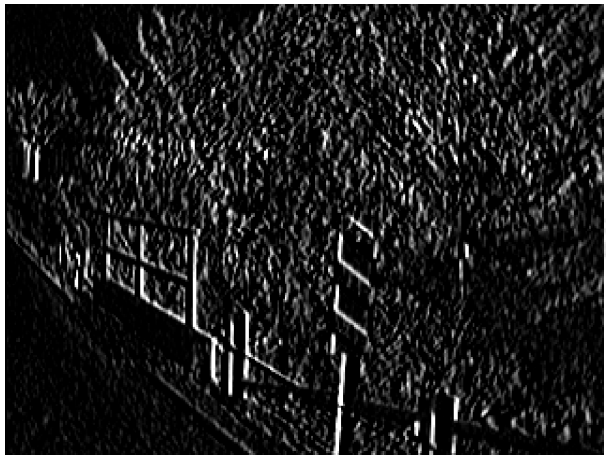
Orientation

$$\mathbf{\Theta} = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x)$$



Corners Form Gradients

Original image



Horizontal

Points/pixels where two edges meet, i.e., high gradient in two directions

Vertical

Corners from Auto-Correlation (Harris Corners)

Important difference in all directions => interest point

C. Harris and M. Stephens (1988). "A combined corner and edge detector". Proceedings of the 4th Alvey Vision Conference: pp 147--151.

Harris Detector

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

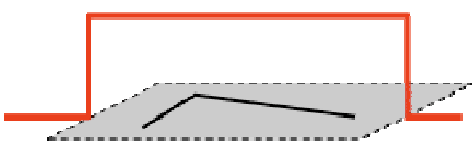
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Window function

Shifted intensity

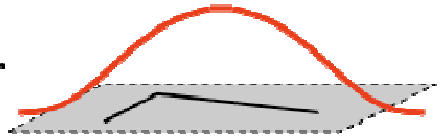
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Harris Detector

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

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

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

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

Harris Detector

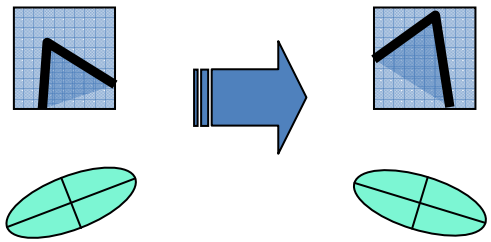
Intensity change in shifting window = eigenvalue analysis

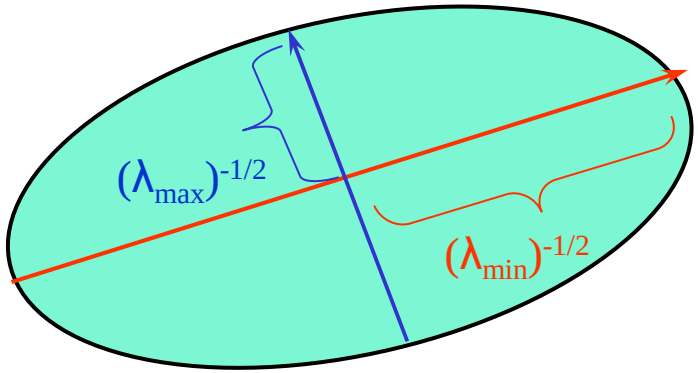
$$E(u,v) \cong \begin{bmatrix} u, v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

λ_1, λ_2 – eigenvalues of M

If we try every possible orientation **n**, the max. change in intensity is λ_2

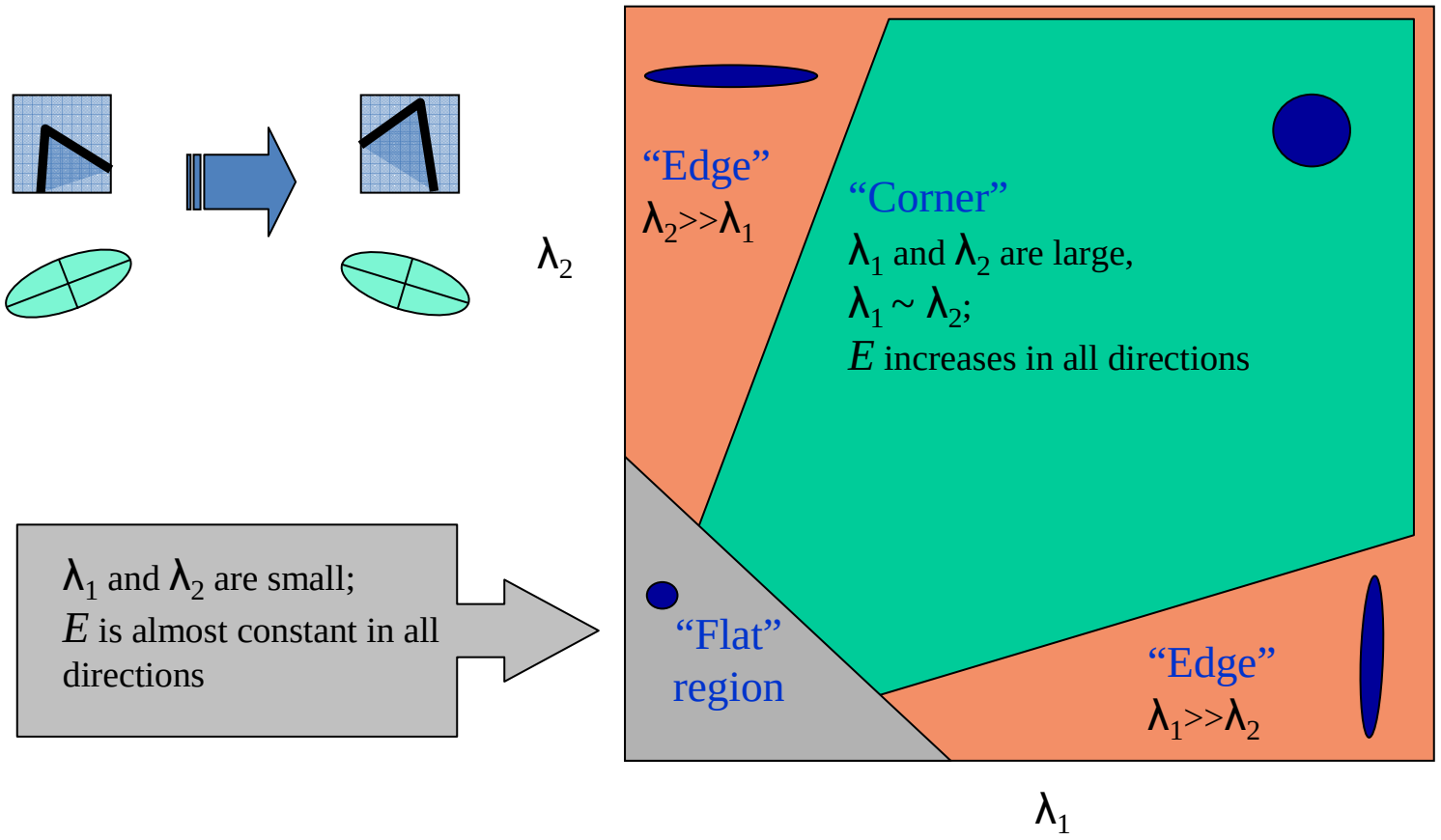
Ellipse $E(u,v)$ = const





Harris Detector

Classification of image points using eigenvalues of M:



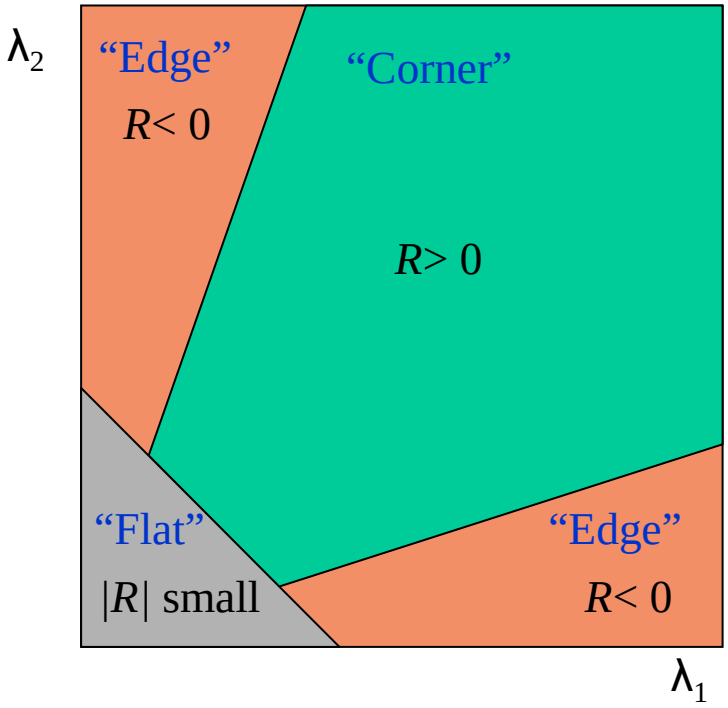
Harris Detector

Speeding up eigenvalues analysis by corner response:

$$R = \det M - k\left(\text{trace}\, M\right)^2$$

$$\det M = \lambda_1 \lambda_2$$
$$\text{trace}\, M = \lambda_1 + \lambda_2$$

(k – empirical constant, k = 0.04-0.06)



R depends only on the (not required) eigenvalues of M

- R is large for a corner
- R is negative with large magnitude for an edge
- |R| is small for a flat region

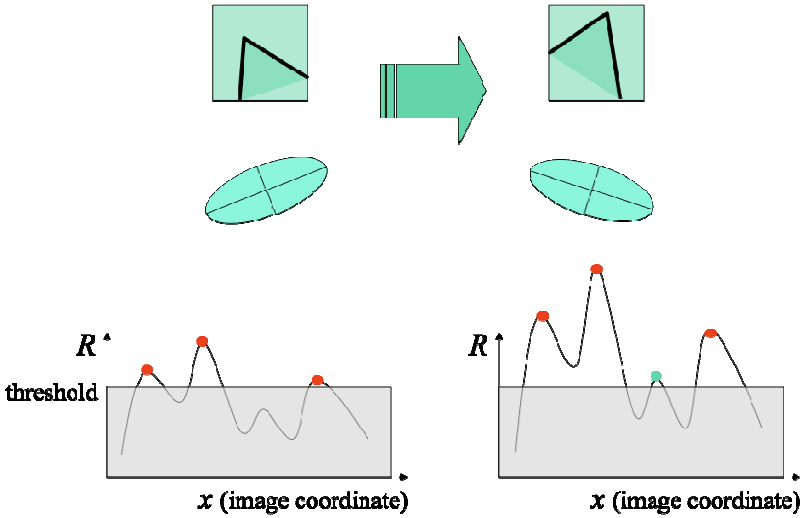
Harris Detector

Summary of steps

1. Compute Gaussian derivatives at each pixel
2. Compute second moment matrix M in a Gaussian window around each pixel
3. Compute corner response function R
4. Threshold R
5. Find local maxima of response function by non maxima suppression

Invariance Properties:

- Rotation
- Affine intensity change

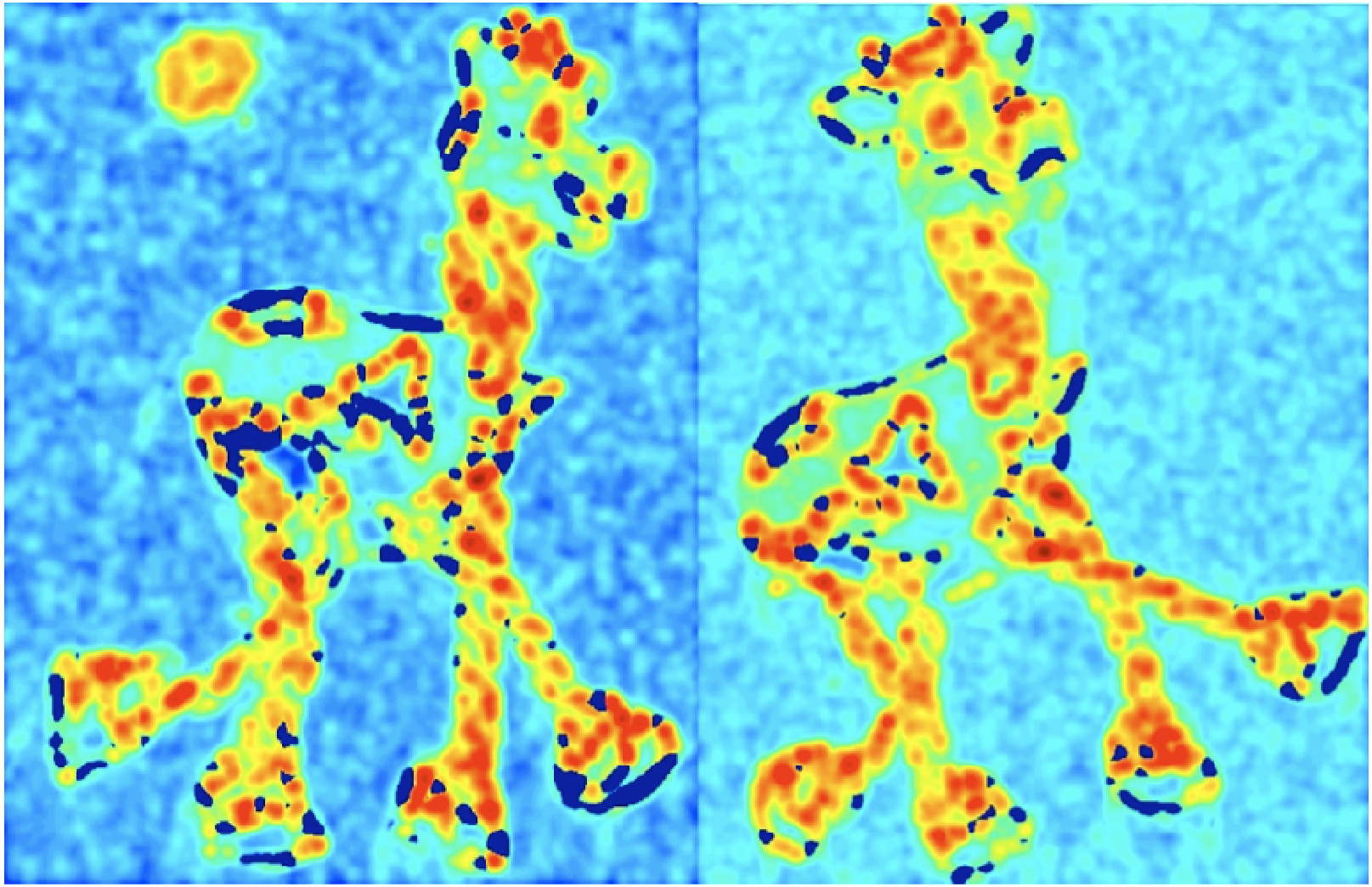


Harris Detector

Original Images

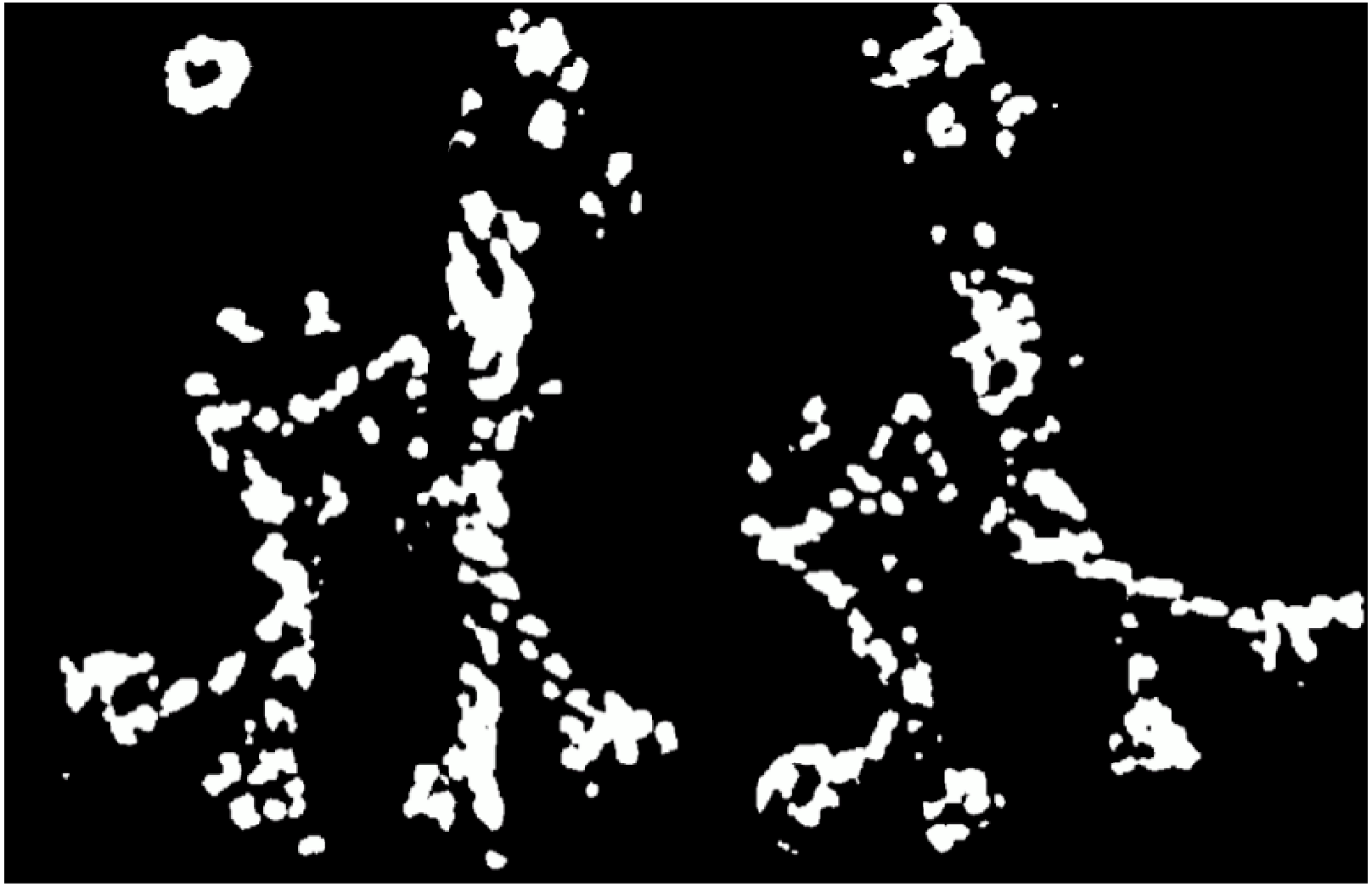
Harris Detector

Compute corner response R



Harris Detector

Find points with large corner response: R>threshold

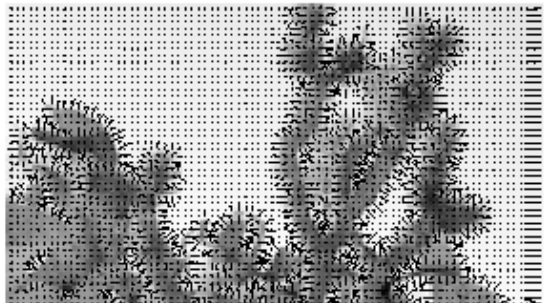


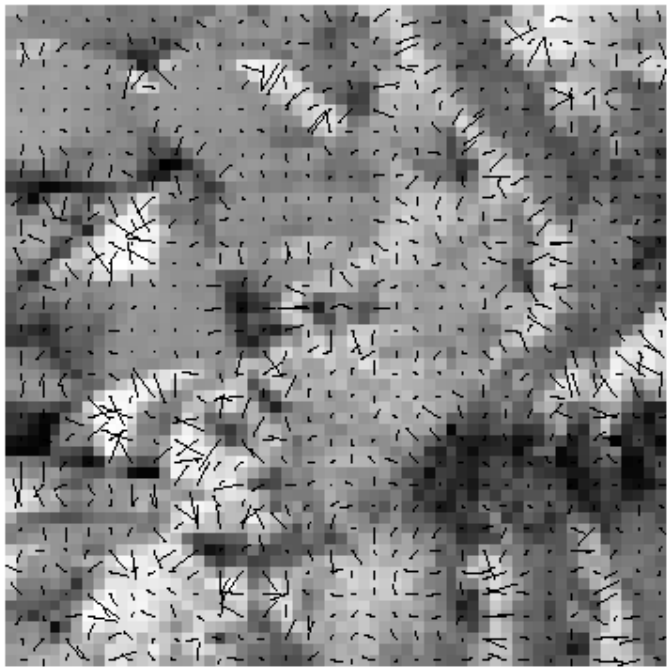
Harris Detector

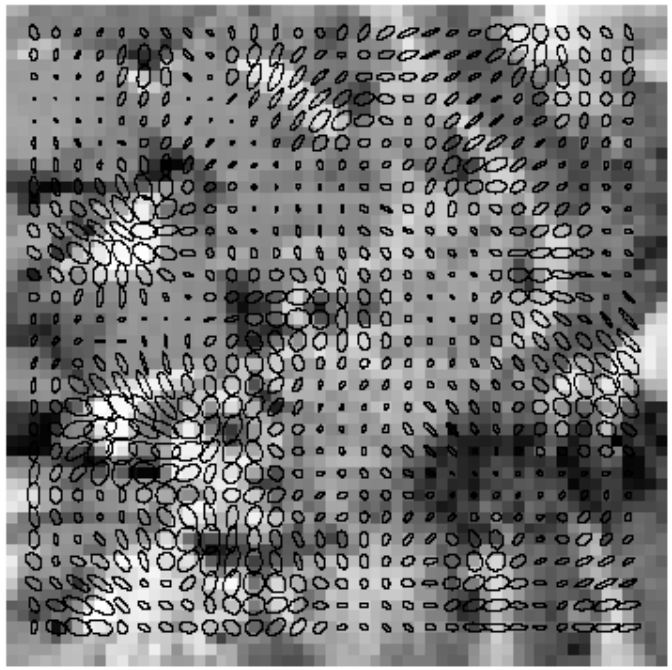
Take only the points of local maxima of R

Harris Detector

Another example

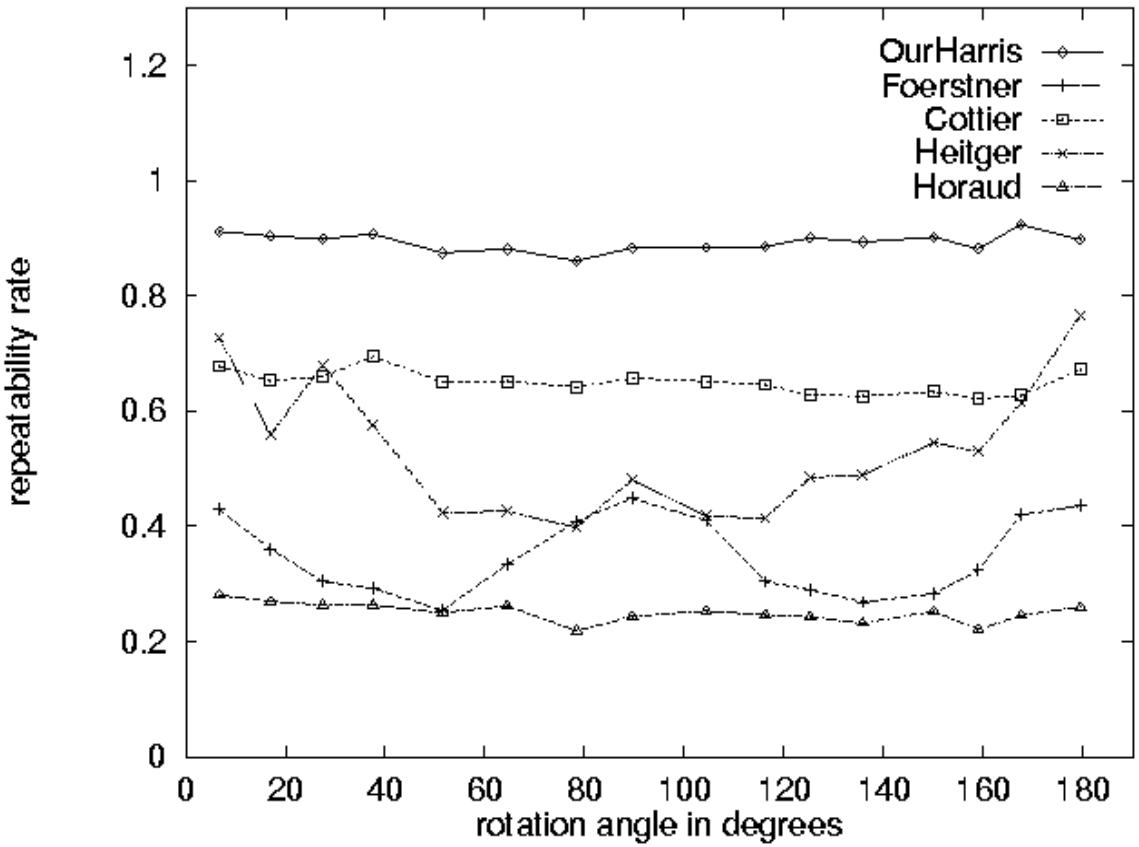






Harris Detector

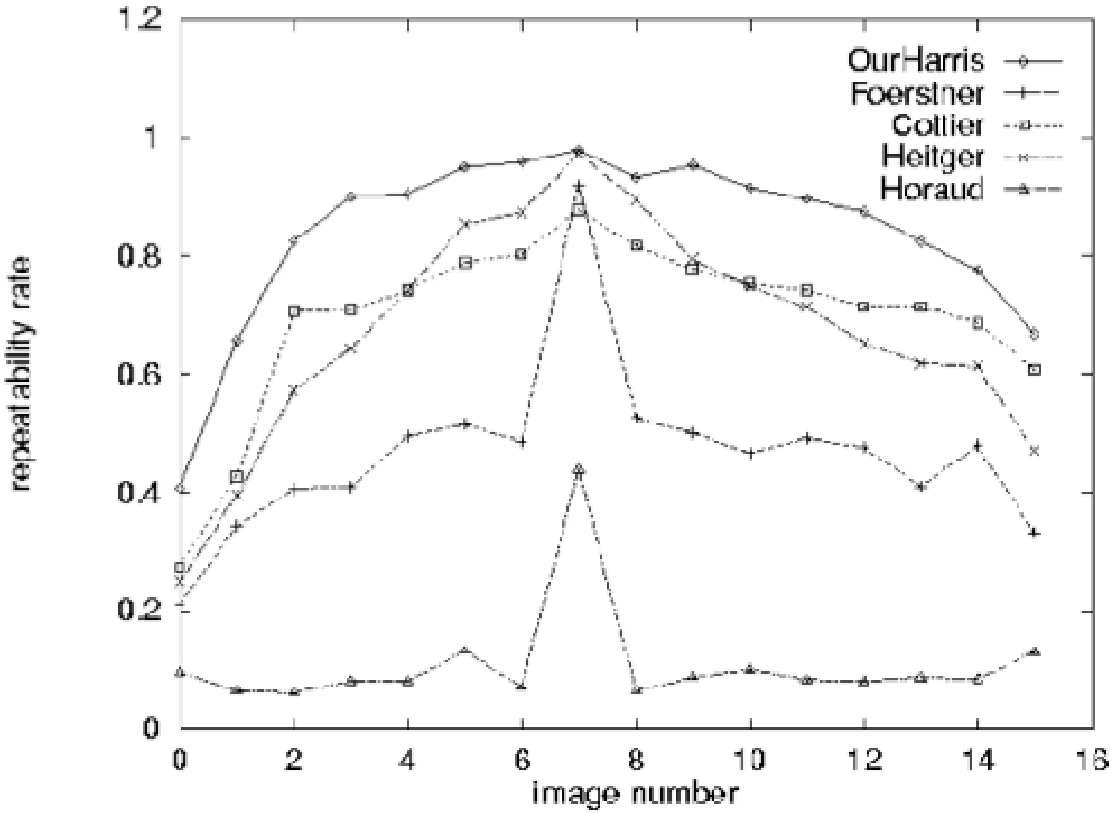
Repeatability - image rotation



[Comparing and Evaluating Interest Points, Schmid, Mohr & Bauckhage, ICCV 98]

Harris Detector

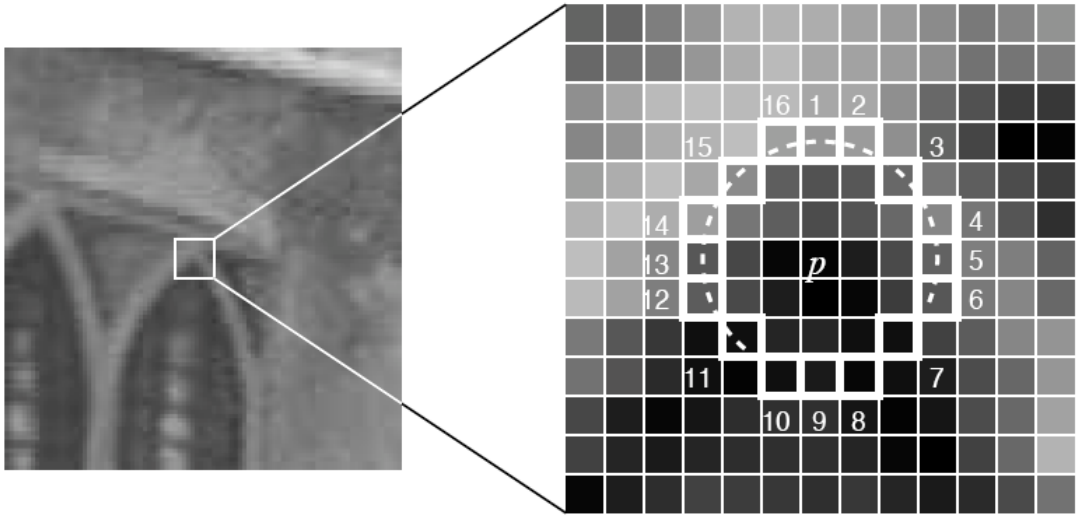
Repeatability - perspective transformation



[Comparing and Evaluating Interest Points, Schmid, Mohr & Bauckhage, ICCV 98]

Features from Accelerated Segment Test (FAST)

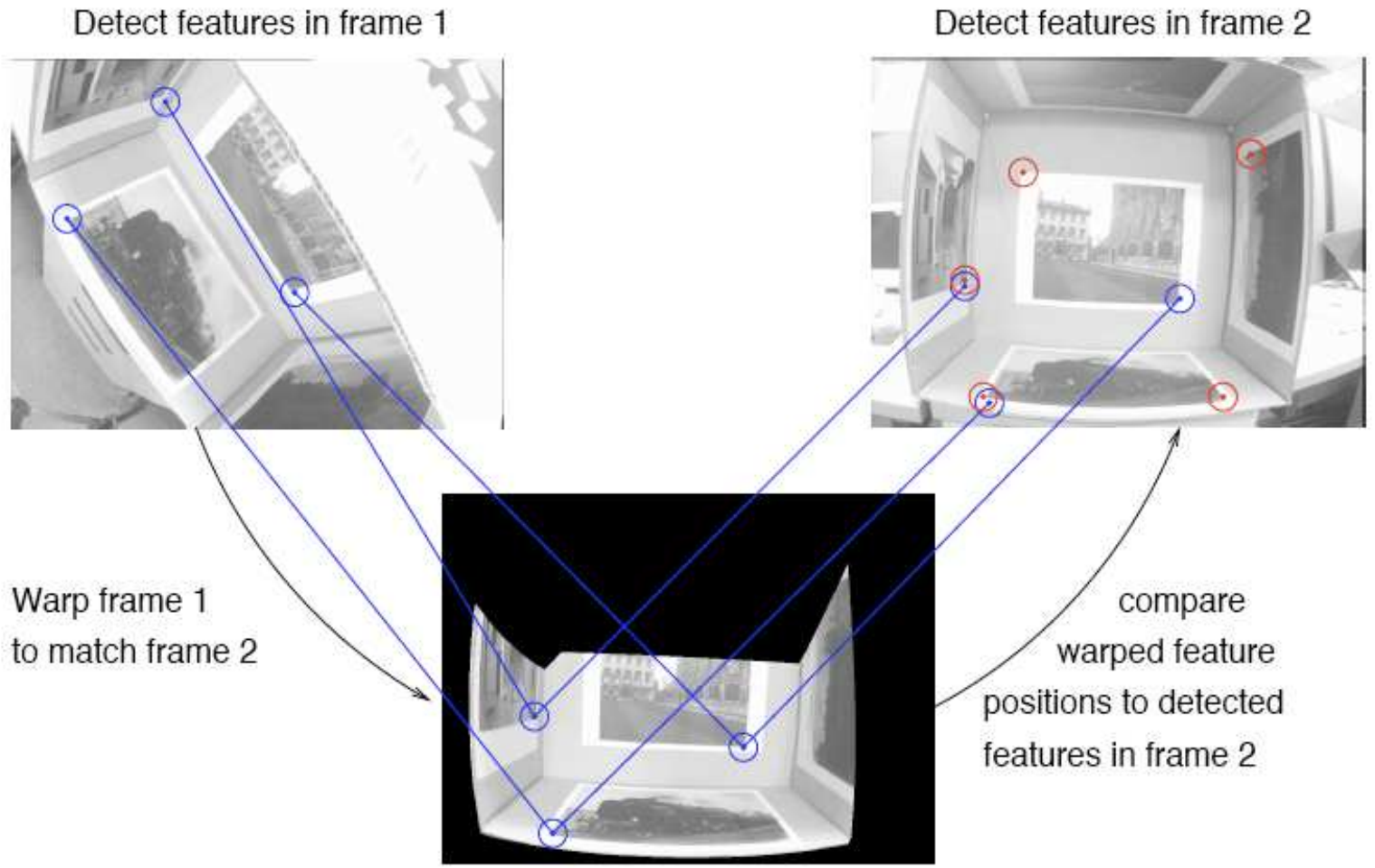
A bright ides: consider pixels in a Bresenham circle of radius *r* around the candidate point.



If n contiguous pixels are all brighter than the nucleus by at least t or all darker than the nucleus by t, then the pixel under the nucleus is considered to be a feature.

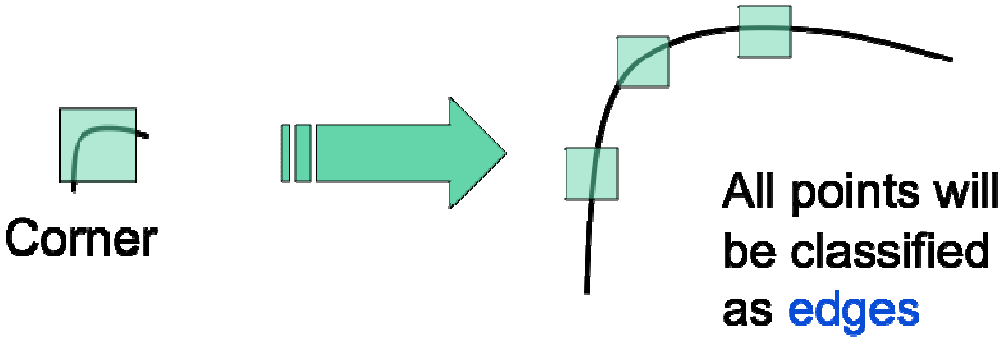
E. Rosten and T. Drummond (May 2006). "Machine learning for high-speed corner detection,". European Conference on Computer Vision.

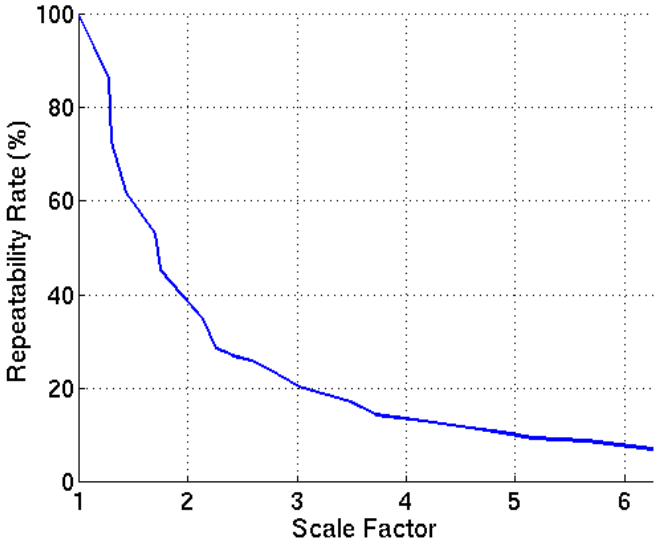
FAST!

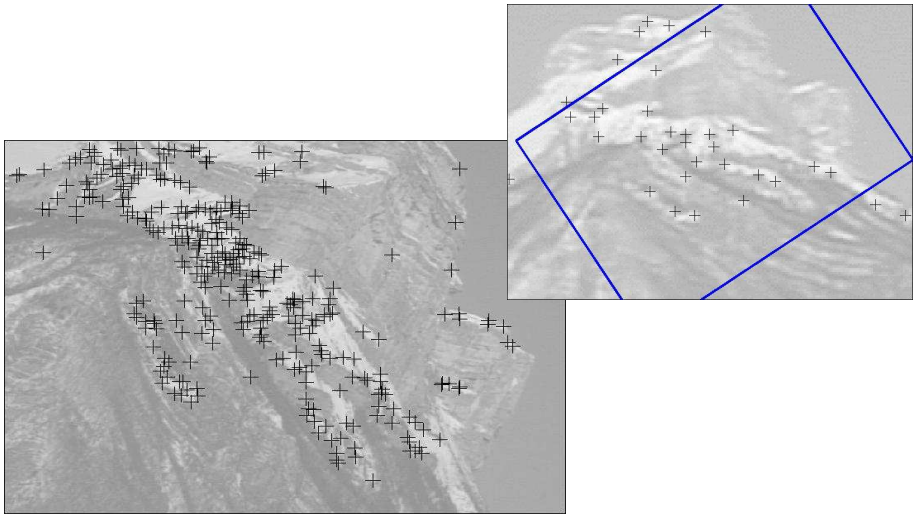

Detect features in frame 1
Detect features in frame 2
Warp frame 1
to match frame 2
compare
warped feature
positions to detected
features in frame 2

Scaling Troubles

So far no corner detector is invariant to scaling

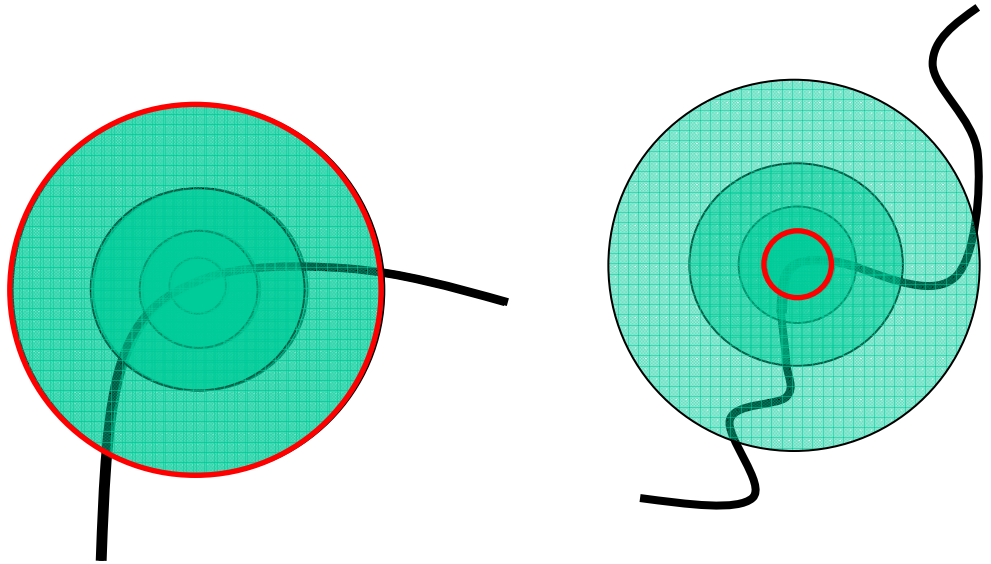






Scaling Invariance

Test regions (e.g., circles) of different size around each candidate point; regions of correct size will “look” the same in both images



But how do we choose corresponding circles independently in each image?

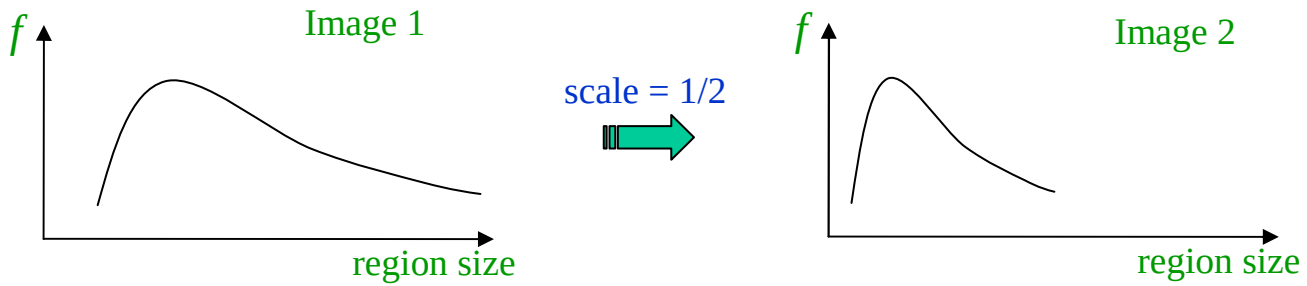
Scaling Invariant Transform

Solution:

- *Design a function on the region (circle), which is "scale invariant" (i.e., the same for corresponding regions, even if they are at different scales)*

Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

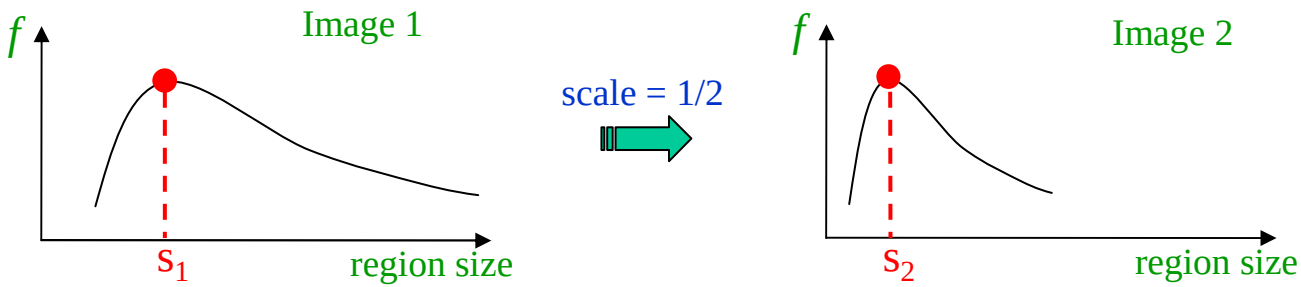
For each point in the image, can consider this as a function of the region size (circle radius)



Scaling Invariant Transform

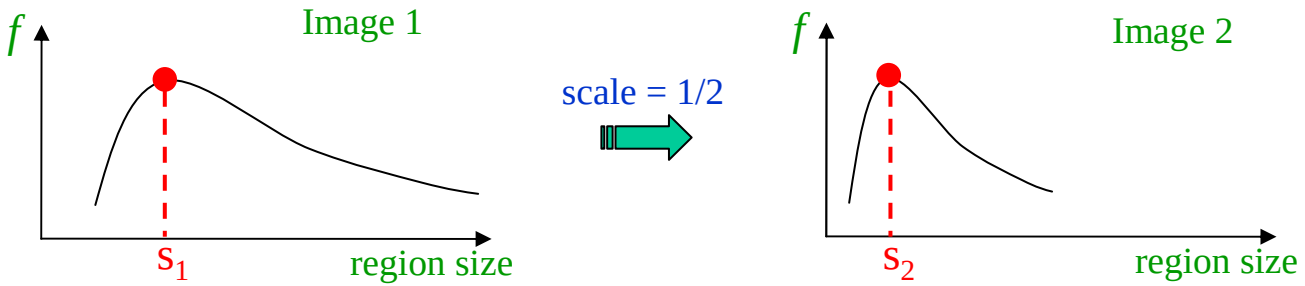
Common approach:

- Take a local maximum of this function; the region size, for which the maximum is achieved, should be invariant to image scaling.



Important: this scale invariant region size is found in each image independently!

Scaling Invariant

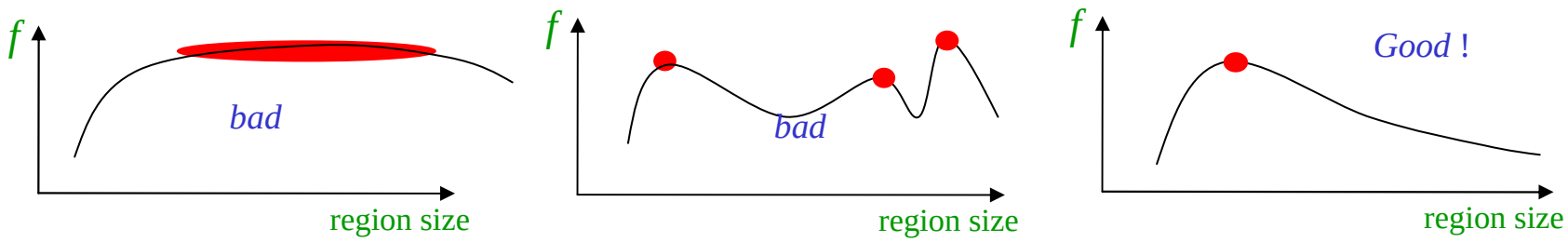


- The ratio of the scales, at which the extrema were found for corresponding points in two rescaled images, is equal to the scale factor between the images.
- **Characteristic Scale**: Given a point in an image, compute the function responses for several factors s_n The characteristic scale is the local max. of the function (can be more than one).
- Easy to look for zero-crossings of 2nd derivative than maxima.

Scaling Invariant

A "good" function for scale detection:

- has one stable sharp peak



For usual images:

a good function would be the one with contrast
(sharp local intensity change)

Scale Invariant Kernels

$$f = \text{Kernel} * \text{Image}$$

Usual Kernels:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

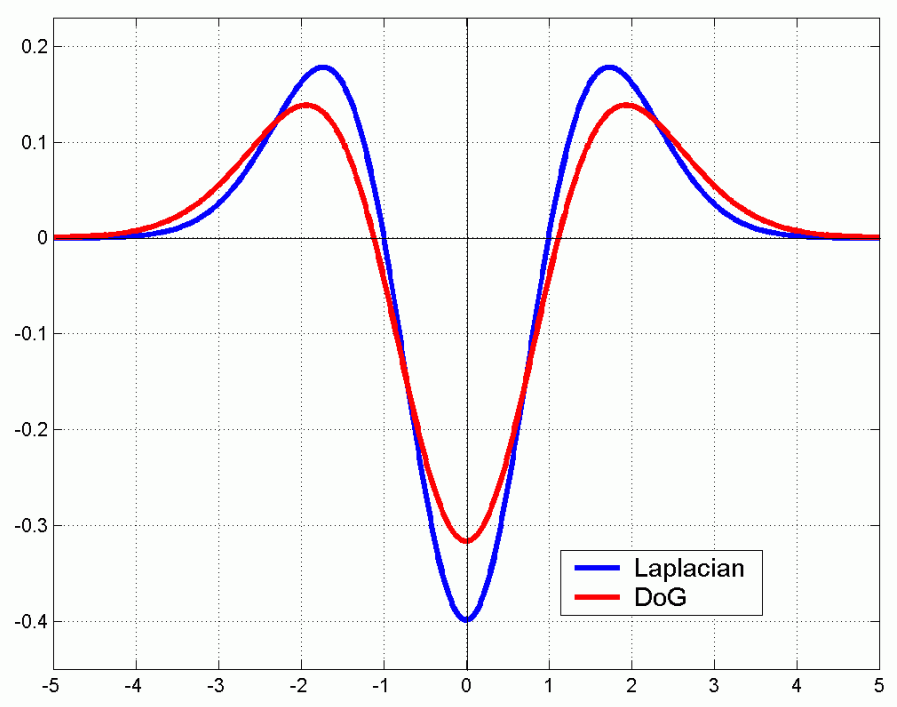
(Laplacian)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)

where Gaussian is defined as $G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2 + y^2}{2\sigma^2}}$

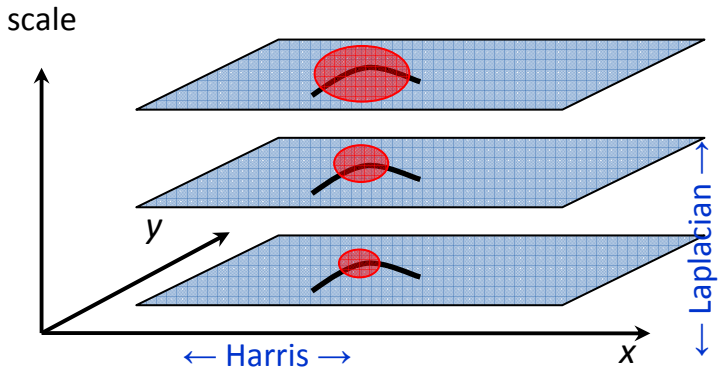
Note: both kernels are invariant to *scale* and *rotation*



Scaling Invariant Approaches

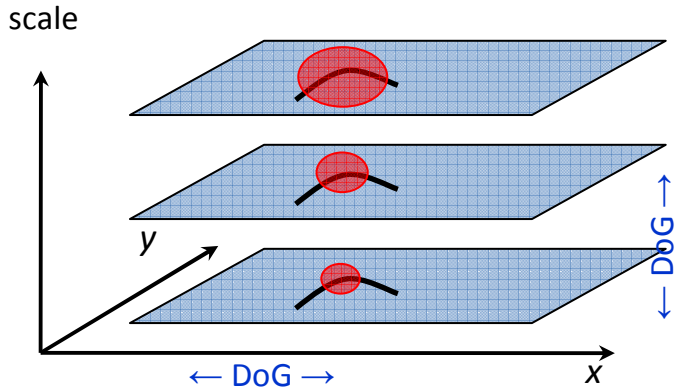
Harris-Laplacian [1]

- Harris corner detector in space (image coordinates)
- Laplacian in scale



Difference of Gaussians (SIFT) [2]

- Difference of Gaussians in space and scale

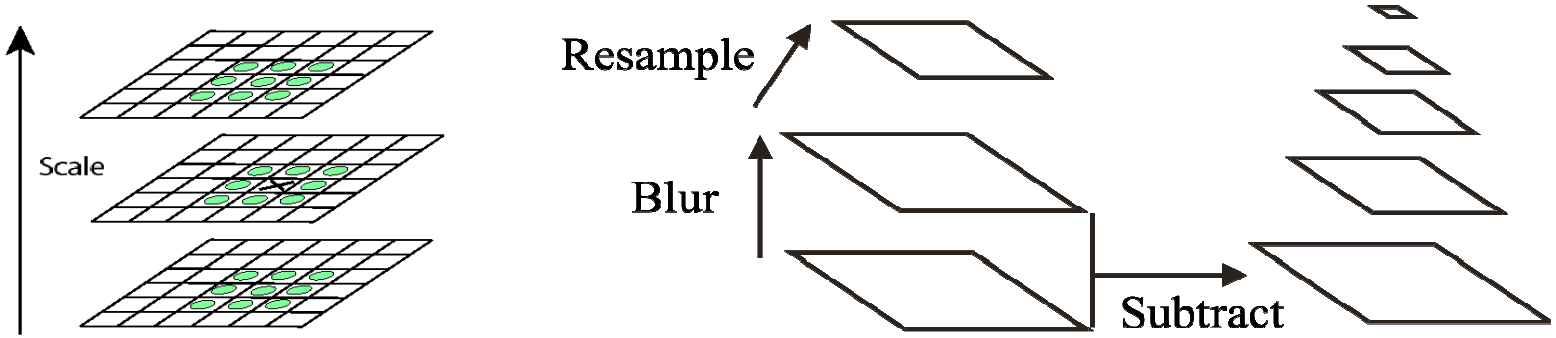


[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

Scale-Space Pyramid

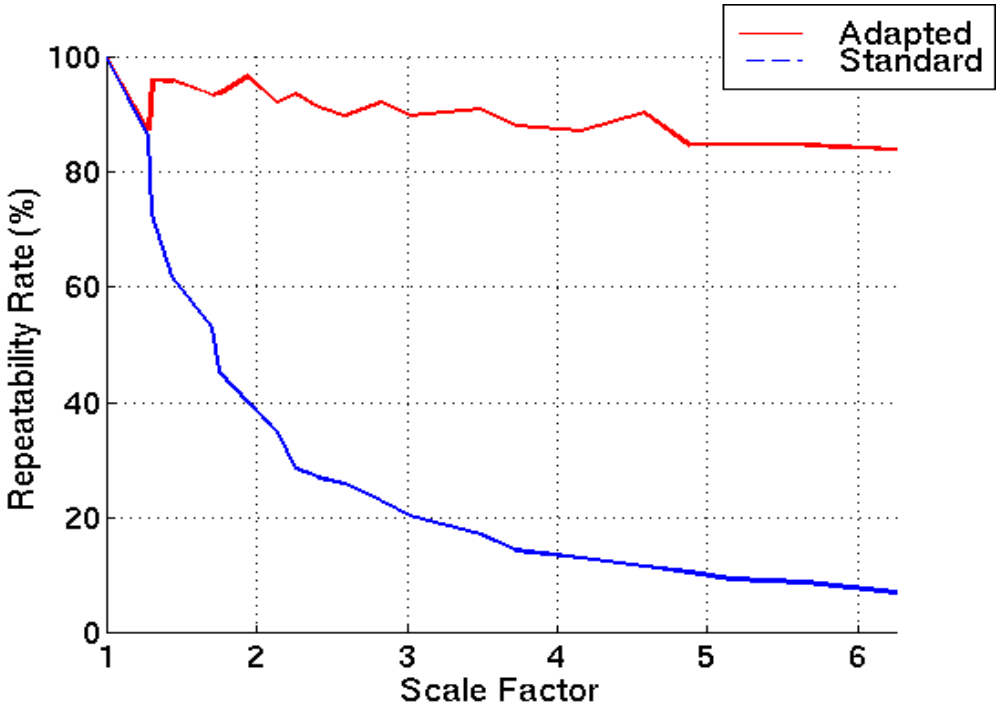
SIFT detector optimized taking into account

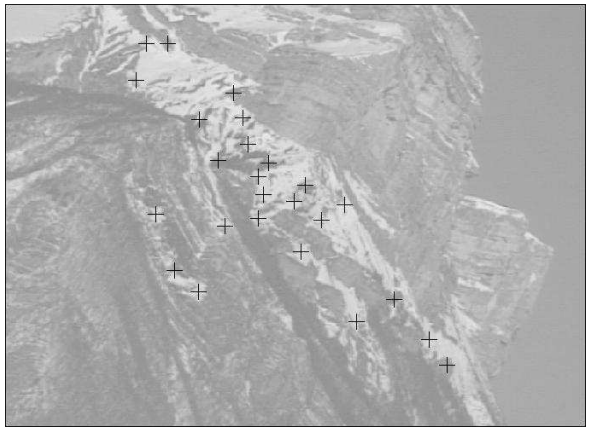
- All scales must be examined to identify scale-invariant features
- An efficient function is to compute the Difference of Gaussian (DOG) pyramid (Burt & Adelson, 1983) (or Laplacian)

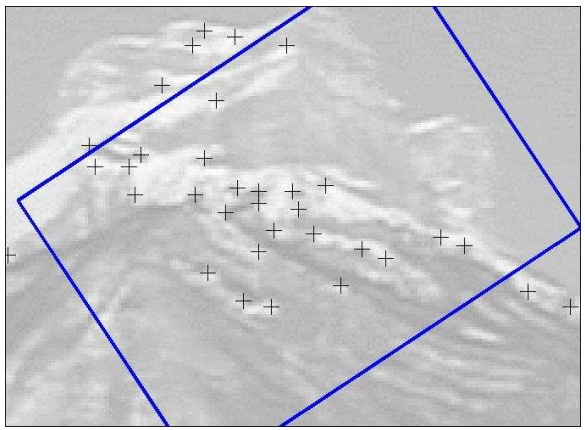


Detect maxima and minima of difference-of-Gaussian in scale space

Scaling Invariance Results





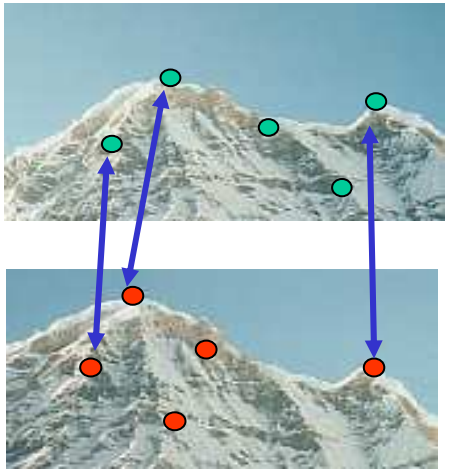


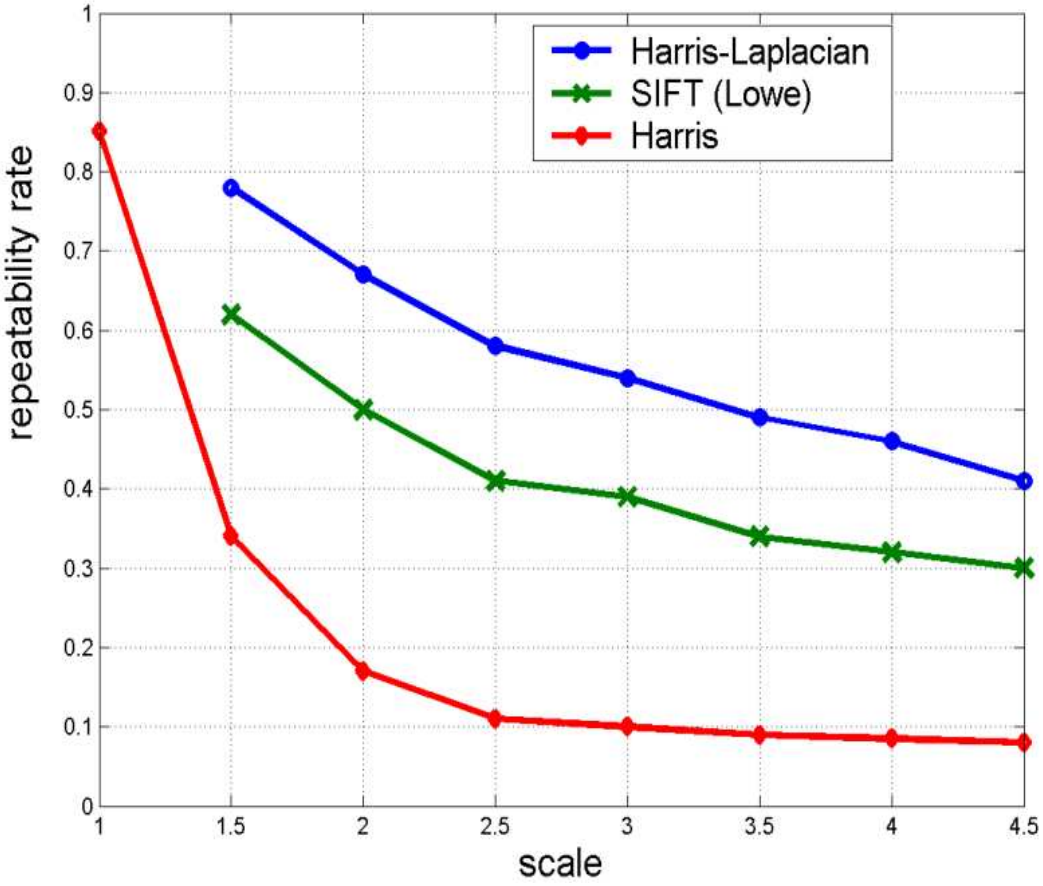
Scaling Invariant Approaches Comparison

Repeatability rate:

$$\frac{\text{\# correspondences}}{\text{\# possible correspondences}}$$

(points present)

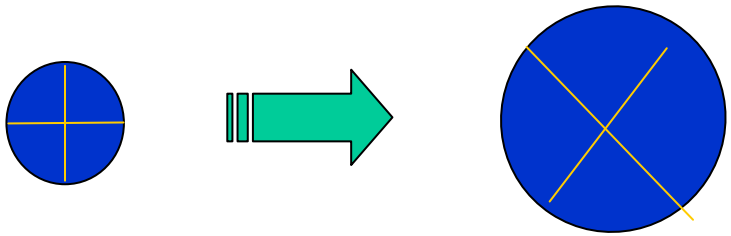




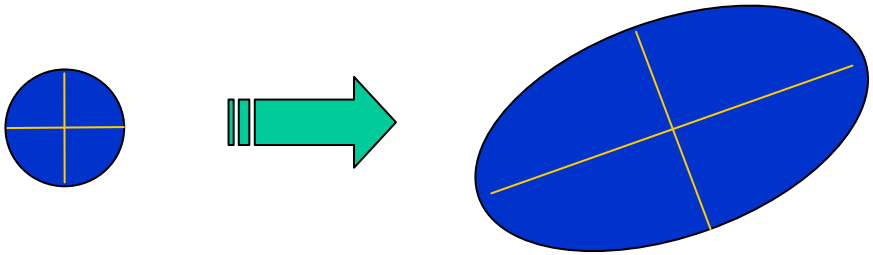
K.Mikolajczyk, C.Schmid. “Indexing Based on Scale Invariant Interest Points”. ICCV 2001

Affine Invariance

Above we considered: Similarity transform (rotation + uniform scale)

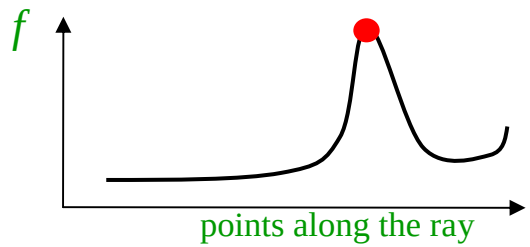


Now we go on to: Affine transform (rotation + non-uniform scale)



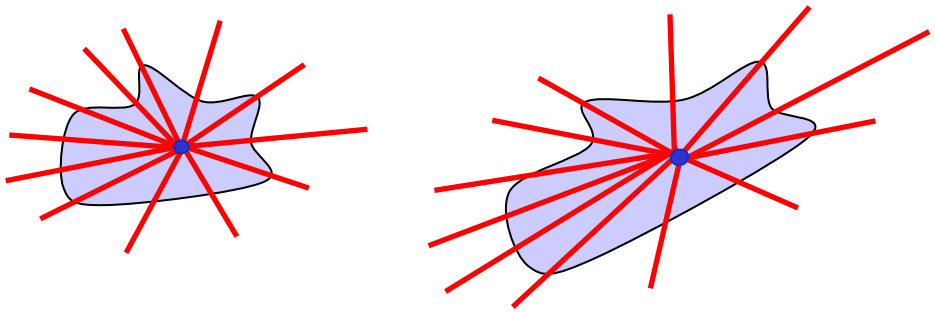
Affine Invariance

Take a local intensity extremum as initial point, then go along every ray starting from this point and stop when extremum of function f is reached



$$f(t) = \frac{|I(t) - I_0|}{\frac{1}{t}\int_0^t |I(t) - I_0| \, dt}$$

We will obtain approximately corresponding regions

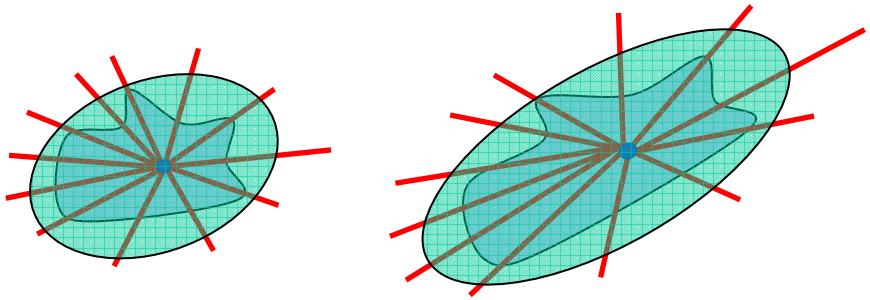


Remark: we search for scale in every direction

T.Tuytelaars, L.V.Gool. "Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions". BMVC 2000

Affine Invariance

The regions found may not exactly correspond, so we approximate them with ellipses …

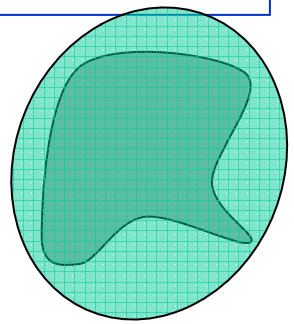


Geometric Moments:

$$m_{pq} = \int_{\mathbb{R}^2} x^p y^q f(x,y)\,dxdy$$

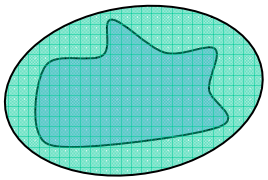
Fact: moments m_{pq} uniquely determine the function f

This ellipse will have the same moments of orders up to 2 as the original region



Affine Invariant Regions

Covariance matrix of region points defines an ellipse:



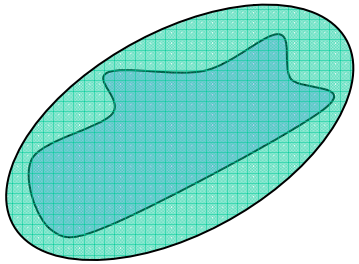
$$p^T \Sigma_1^{-1} p = 1$$

$$\Sigma_1 = \left\langle p p^T \right\rangle_{\text{region 1}}$$

($p = [x, y]^T$ is relative to the center of mass)

$$q = Ap$$

$$\Sigma_2 = A \Sigma_1 A^T$$



$$q^T \Sigma_2^{-1} q = 1$$

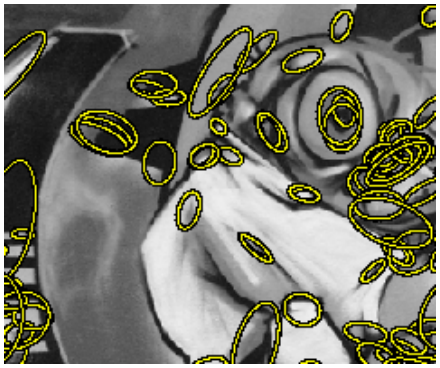
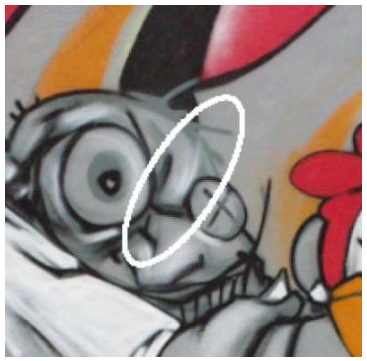
$$\Sigma_2 = \left\langle q q^T \right\rangle_{\text{region 2}}$$

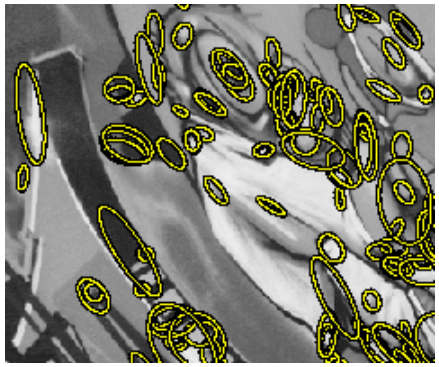
Algorithm summary (detection of affine invariant region):

- Start from a local intensity extremum point
- Go in every direction until the point of extremum of some function f
- Curve connecting the points is the region boundary
- Compute geometric moments of orders up to 2 for this region
- Replace the region with ellipse

Affine Invariant Regions

Normalization example



Affine Invariant

Maximally Stable Extremal Regions

- Threshold image intensities: $I > I_0$
- Extract connected components ("Extremal Regions")
- Find a threshold when an extremal region is "Maximally Stable", i.e. local minimum of the relative growth of its square
- Approximate a region with an ellipse

J.Matas et.al. "Distinguished Regions for Wide-baseline Stereo". Research Report of CMP, 2001.

Affine Invariant

Experiments with MSER

J. Matas, O. Chum, M. Urba, and T. Pajdla. "Robust wide baseline stereo from maximally stable extremal regions." Proc. of British Machine Vision Conference, pages 384-396, 2002.

Summary

What is a feature?

How to find features (aka detectors)

- Harris
- Features from Accelerated Segment Test (FAST) ??
- Scale Invariant
- Affine Invariant

How to describe a feature (aka descriptors)

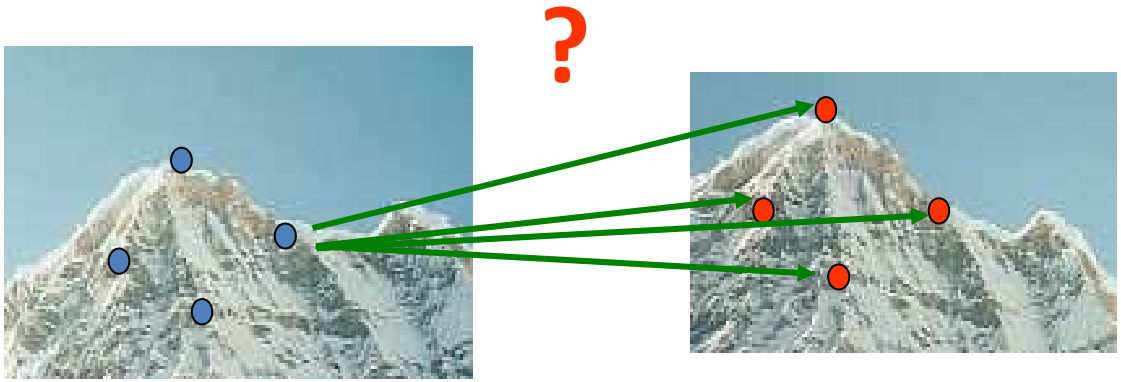
- Patch
- Scale Invariant Feature Transform (SIFT)

How to make the matching robust

- RANdom SAmple Consensus (RANSAC)

Matching with Features

Problem: “for each point correctly recognize the corresponding one”



We need a reliable and distinctive ***descriptor***

Image Patch

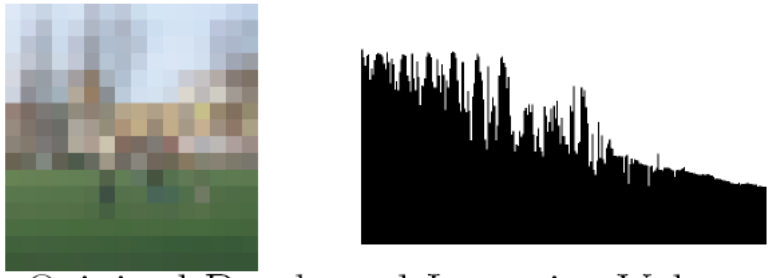
Cross-Correlation

$$CC(P_1, P_2) = \frac{1}{N} \sum_{i}^{N} P_1[i] P_2[i].$$

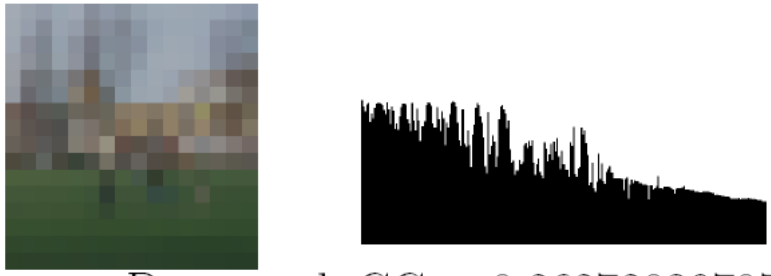
Output in range [-1,+1]

Not invariant to affine photometric transformations

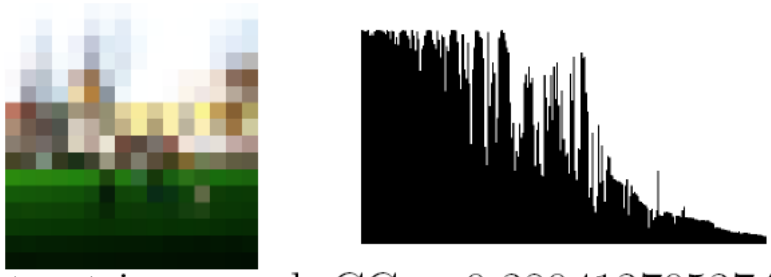
$$I \rightarrow a\, I + b$$



Original Patch and Intensity Values



Brightness Decreased, CC = 0.2627203970780 39



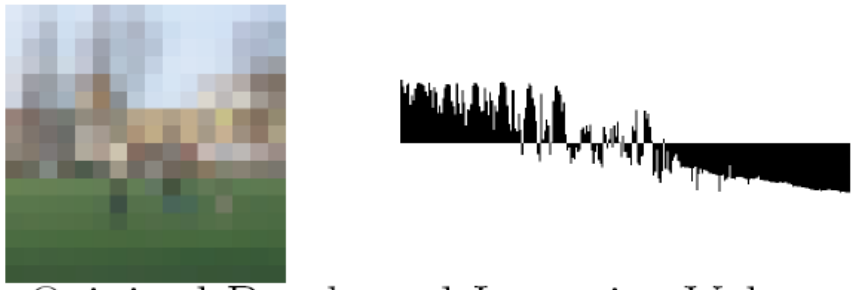
Contrast increased, CC = 0.380413705374859

Image Patch Normalization

Make each patch zero mean:

$$\mu = \frac{1}{N}\sum_{x,y} I(x,y)$$

$$Z(x,y) = I(x,y) - \mu$$

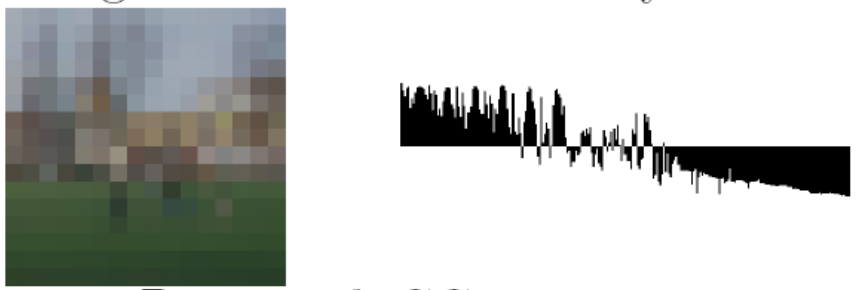


Original Patch and Intensity Values

Then make unit variance:

$$\sigma^2 = \frac{1}{N}\sum_{x,y} Z(x,y)^2$$

$$ZN(x,y) = \frac{Z(x,y)}{\sigma}$$



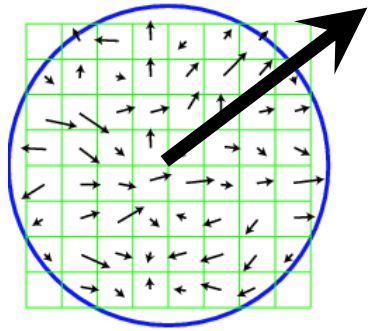
Brightness Decreased, CC = 0.99998895629

Contrast increased, CC = 0.969868160814

SIFT (Descriptor)

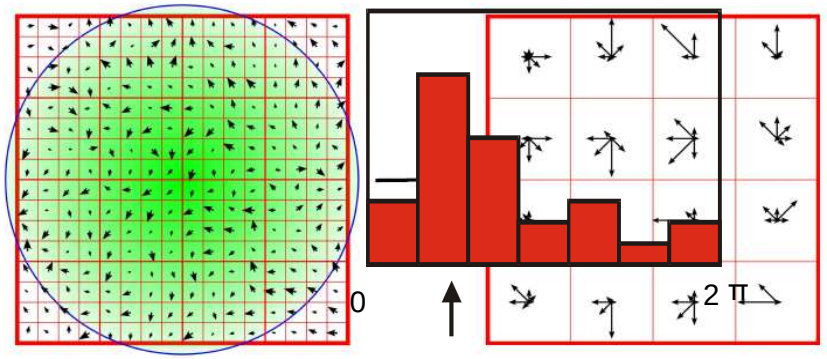
Scale Invariant Feature Transform uses an alternative representation for image patches

- Location and scale given by DoG detector
- Perform orientation normalization
- Find dominant orientation by building smoothed orientation histogram



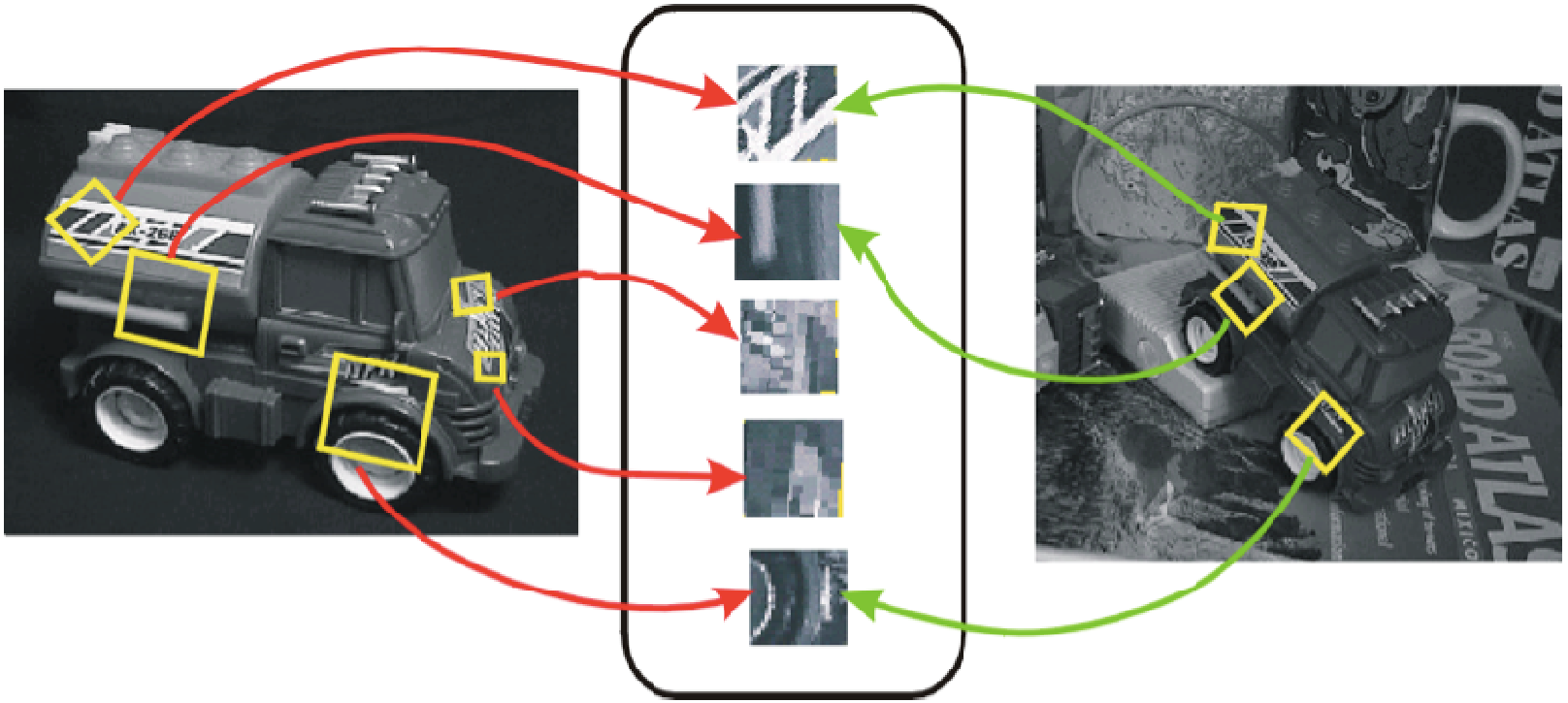
With a few tricks

- Gaussian center-weighting
- Normalized to unit norm
- 4x4 spatial bins (16 bins total)
- 8-bin orientation histogram per bin
- 8 x 16 = 128 dimensions total



Lowe, David G. (1999). "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision 2: 1150–1157.

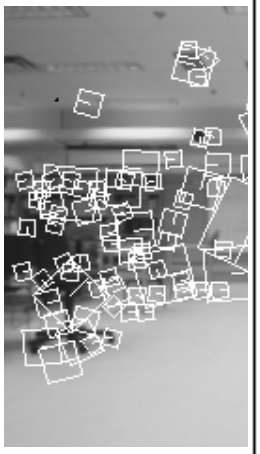
SIFT Examples

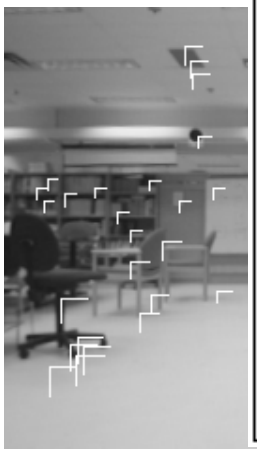


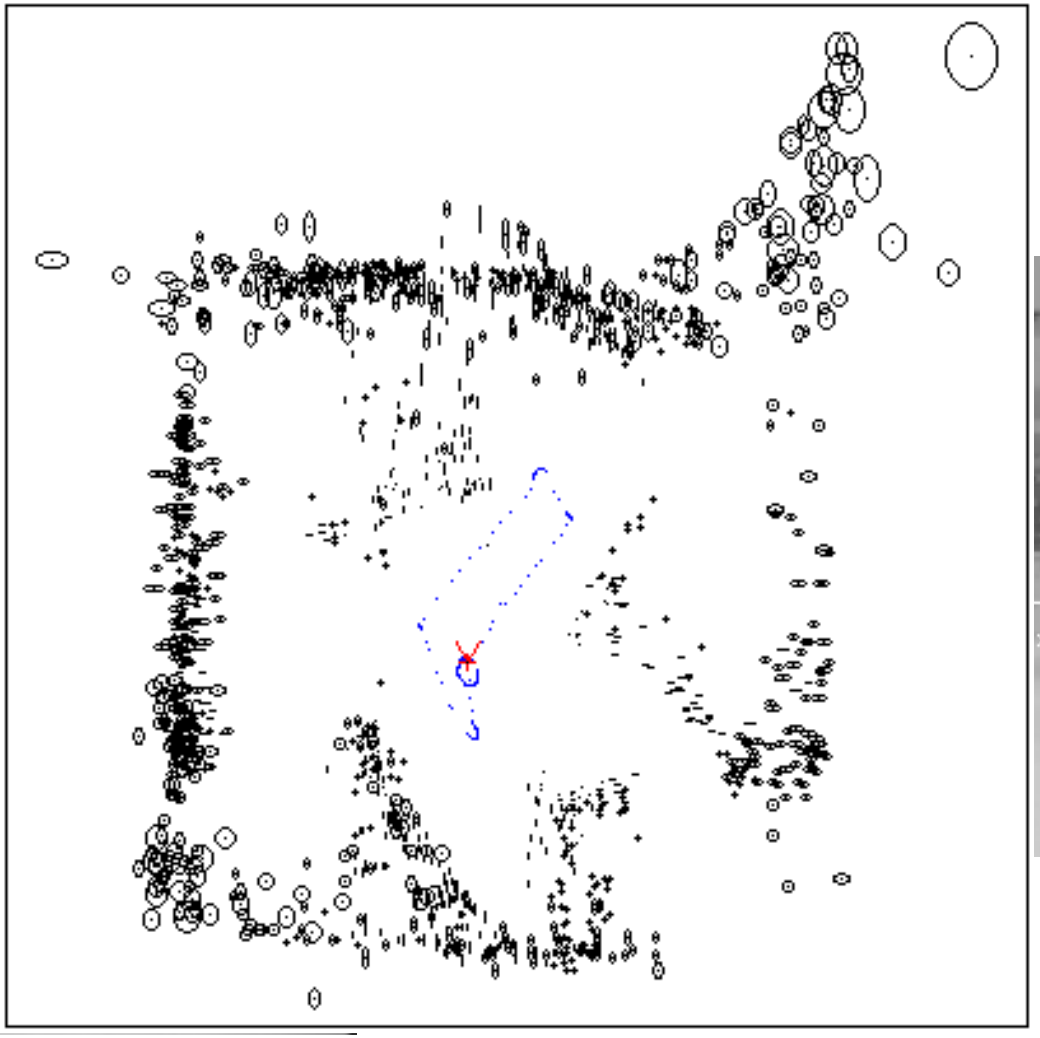
SIFT Examples

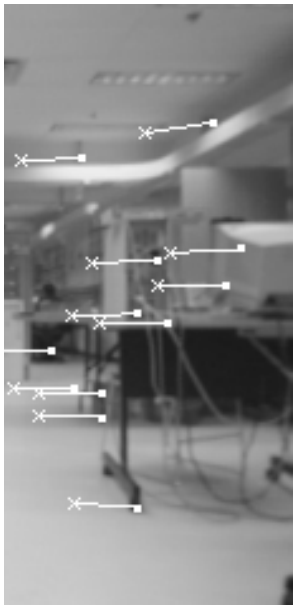
SIFT

Results









Summary

What is a feature?

How to find features (aka detectors)

- Harris
- Features from Accelerated Segment Test (FAST) ??
- Scale Invariant
- Affine Invariant

How to describe a feature (aka descriptors)

- Patch
- Scale Invariant Feature Transform (SIFT)

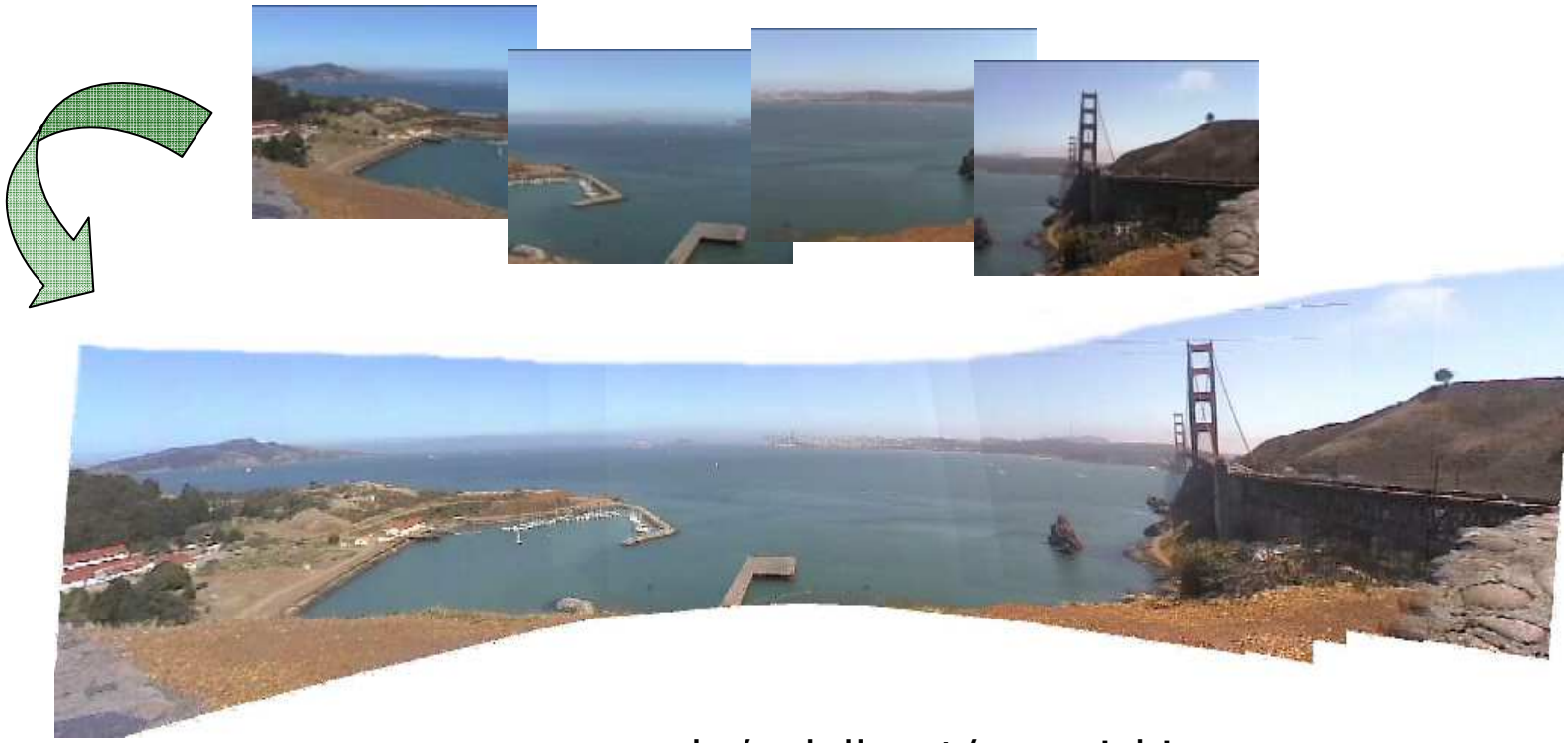
How to make the matching robust

- RANdom SAmple Consensus (RANSAC)

Remind me why …

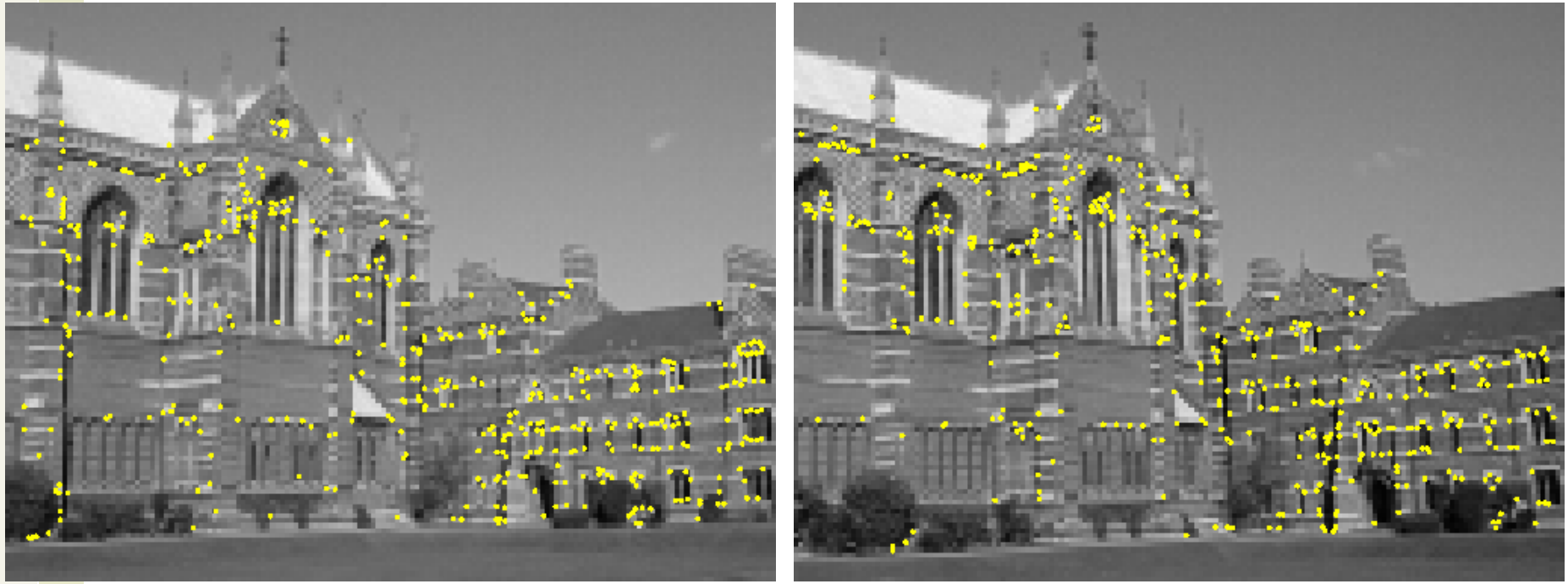
Estimating motion models (transformations) using corresponding points

- Translation
- Homography
- Fundamental matrix

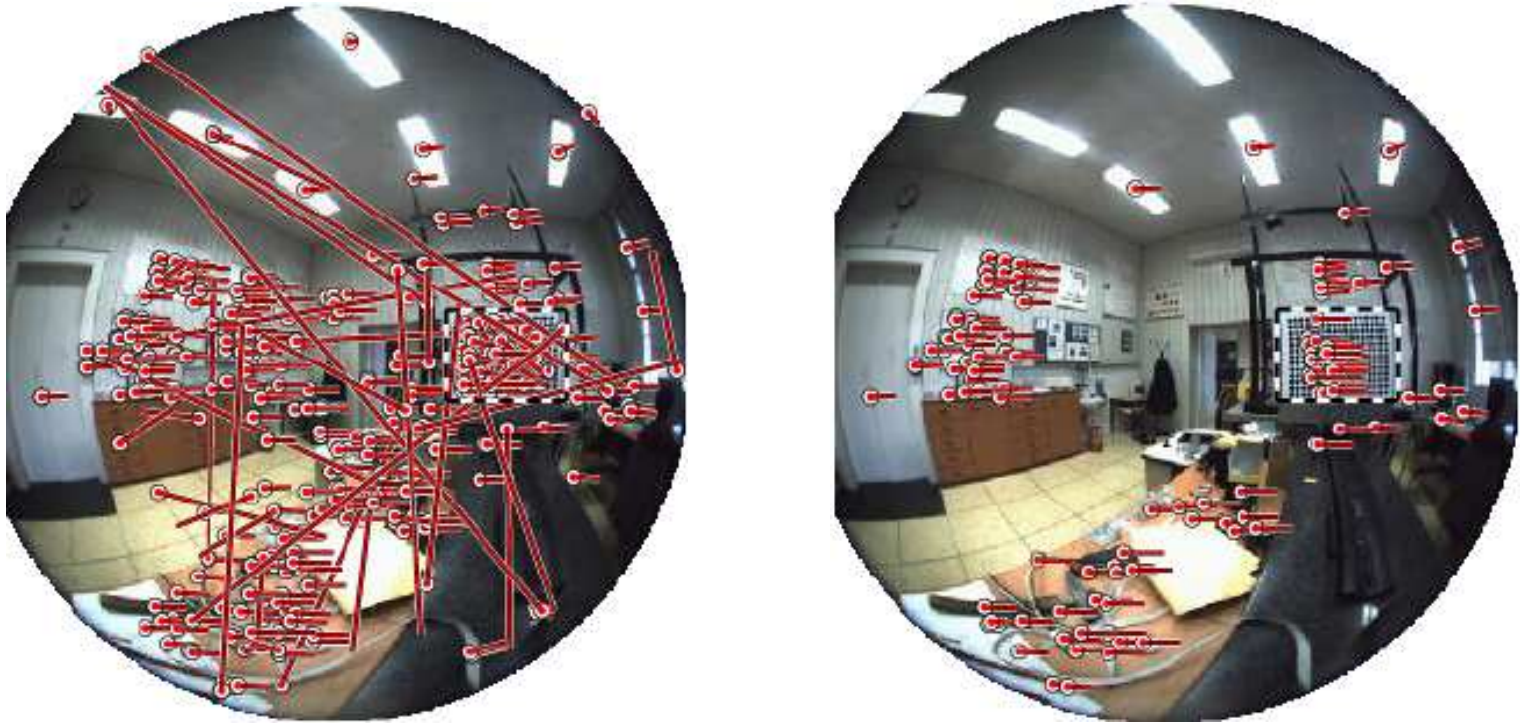


www.cs.cmu.edu/~dellaert/mosaicking

Fundamental Matrix

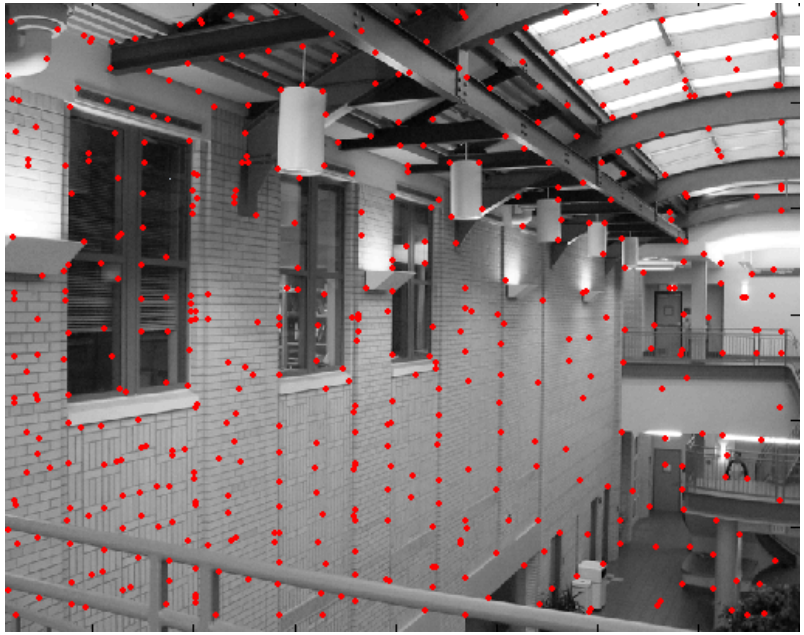


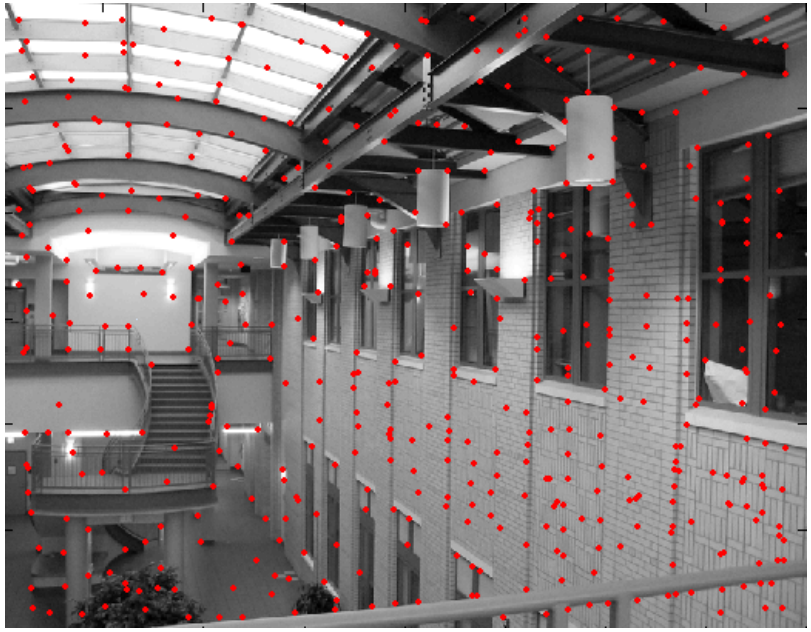
An Omnidirectional Example

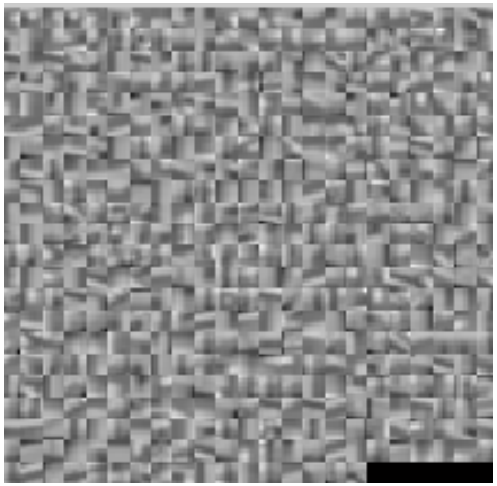


Images by Branislav Micusik, Tomas Pajdla,
cmp.felk.cvut.cz/ demos/Fishepip/

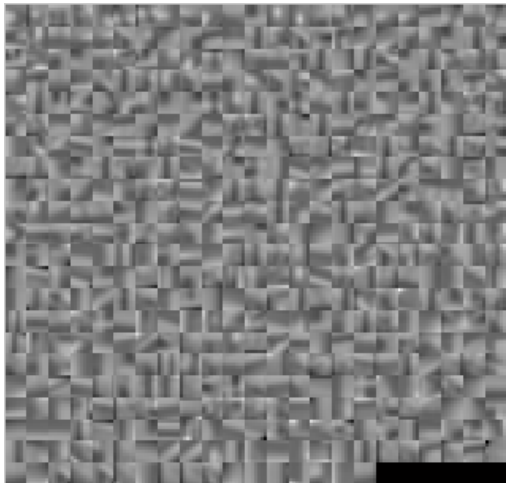
Pick one …







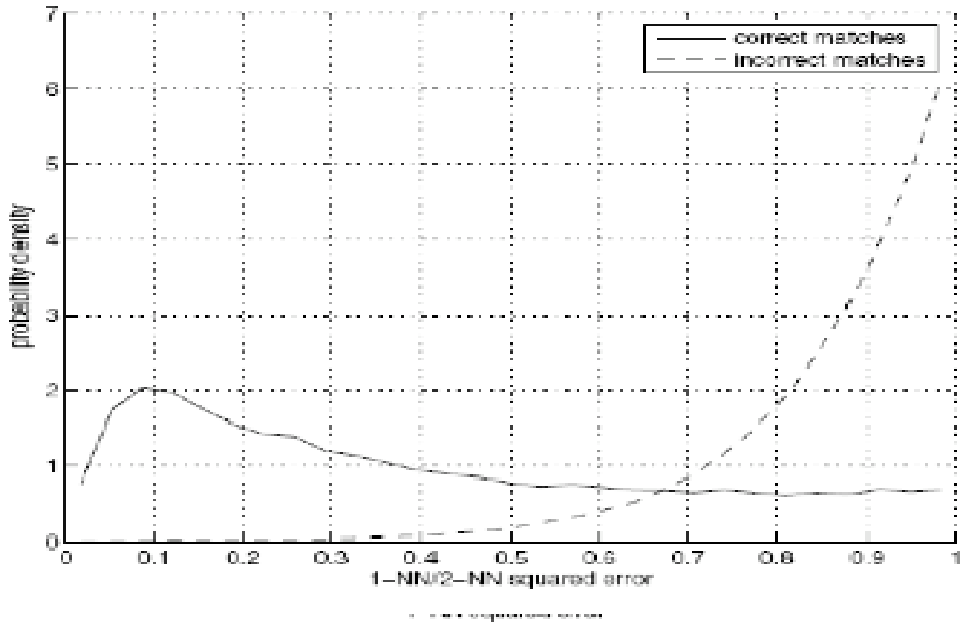
?



Improved Matching Criteria

Let's only those features "similar enough"

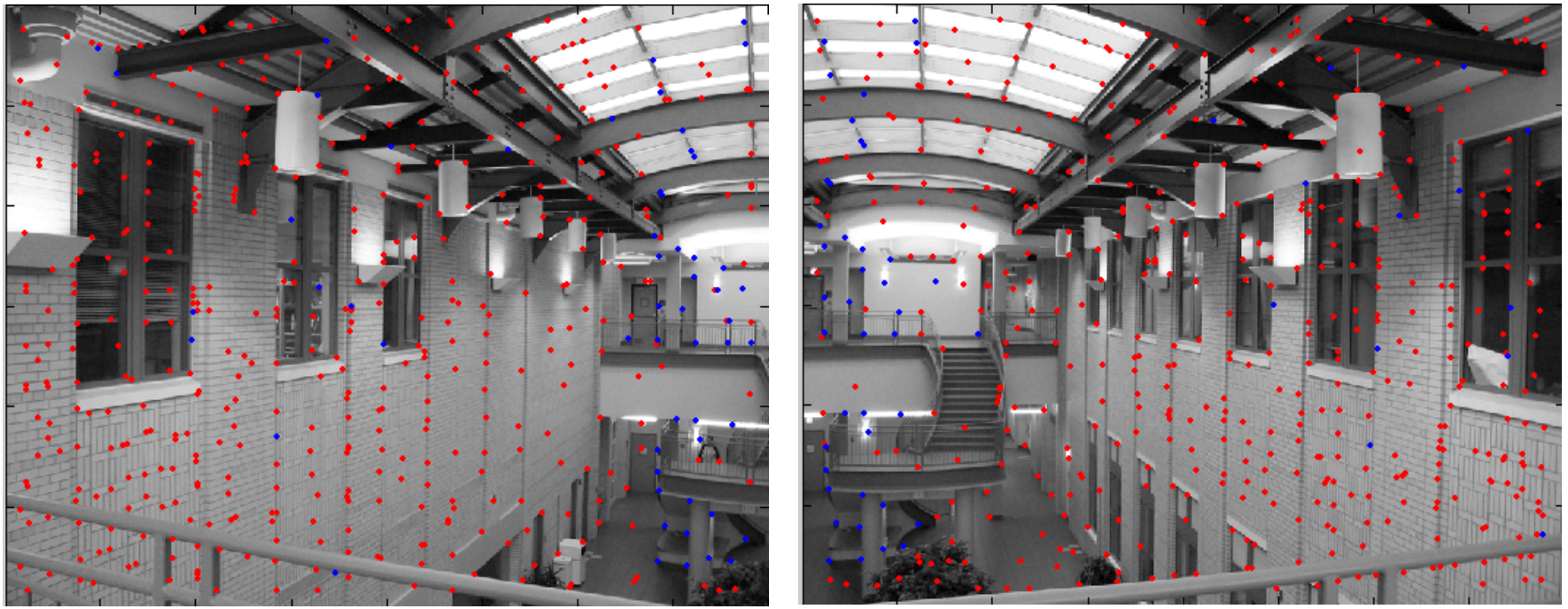
- SSD(patch1,patch2) < threshold
- How to set threshold?



A better way [Lowe, 1999]:

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

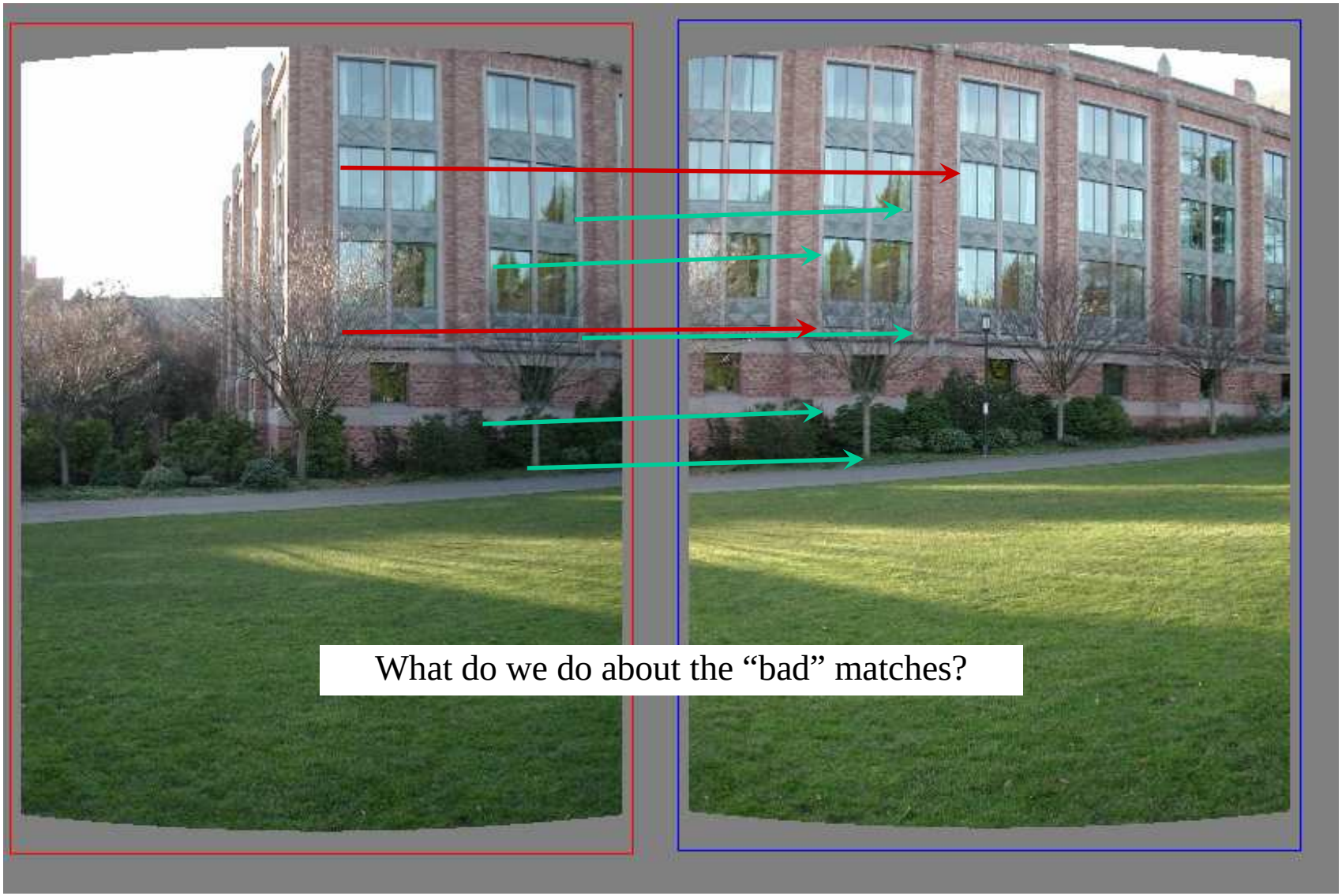
… try it again …



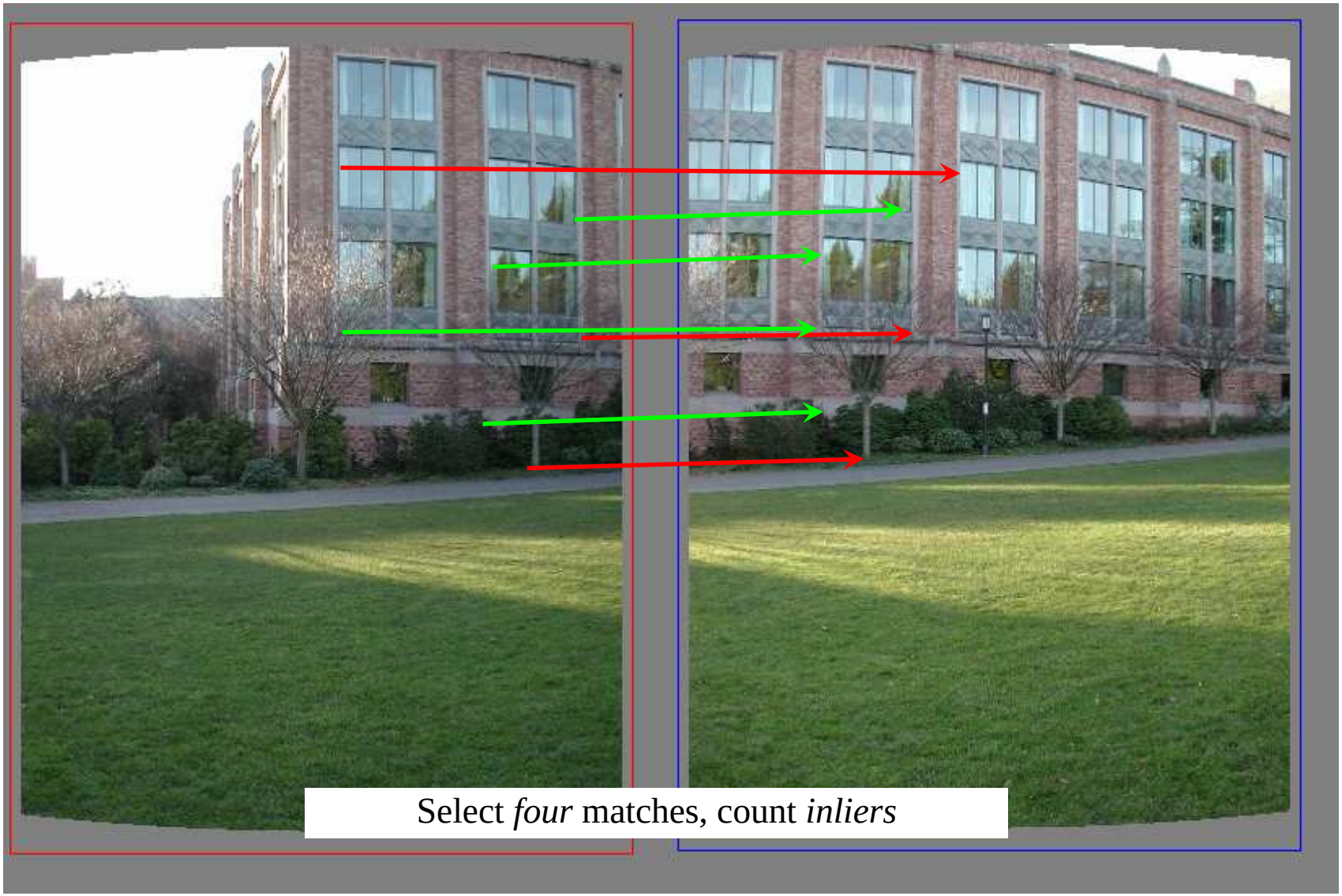
Can we now compute homography from blue points?

- No! Still too many outliers …

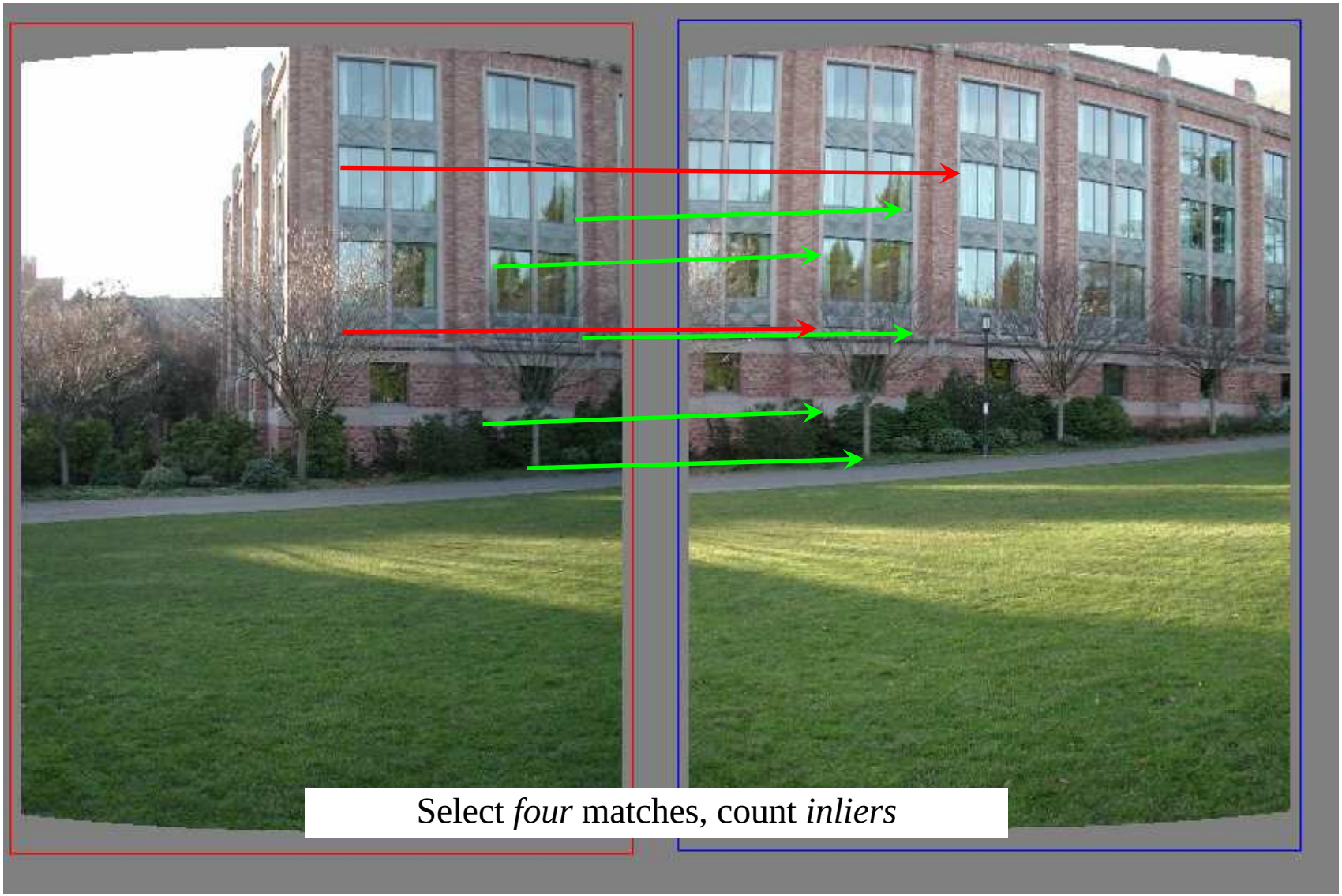
Dealing with Outliers



Dealing with Outliers



Dealing with Outliers



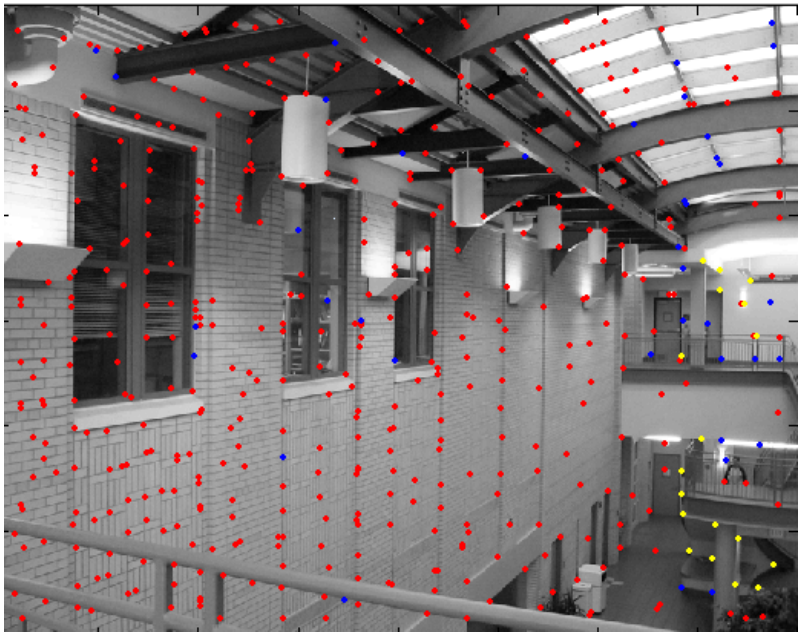
Dealing with Outliers

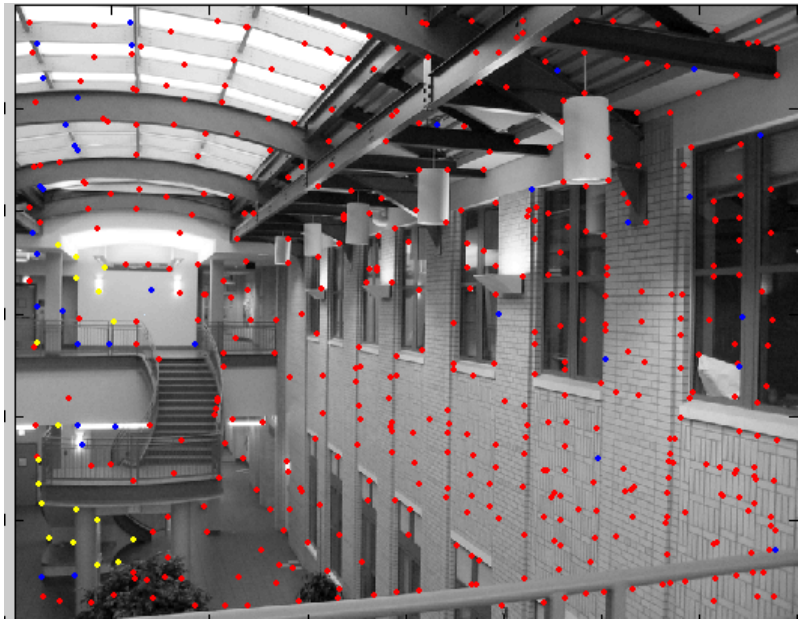
Loop for homography:

1. Select four feature pairs (at random)
2. Compute homography H (exact)
3. Compute inliers where SSD(pi', H pi) < ε
4. Keep largest set of inliers
5. Re-compute least-squares H estimate on all of the inliers

The best homography has most support

… oh yeah!





RANSAC

RANSAC: RANdom SAmple Consensus

Objective: Robust fit of a model to data S

Algorithm:

1. Randomly select s points
2. Instantiate a model
3. Get consensus set Si
4. If |Si|>T, terminate and return model
5. Repeat for N trials, return model with max |Si|

But how many N to guarantee at least one set of initial points containing all inliers with probability p = 0.99?

RANSAC: How big is N?

Let consider a model with *s* parameters and all the points selected independently, then:

- w = proportion of inliers = 1- etha
- P(sample with all inliers)=w^s
- P(sample with at least an outlier)=$1-w^s$
- P(N samples an outlier)=$(1-w^s)^N$

We obtain for target probability p

$$(1-w^s)^N<1-p$$

$$N \log(1-w^s)<\log(1-p)$$

$$N > \log(1-p)/\log(1-w^s)$$

For p=0.99, w=0.5, s=4 we have N>11

Homography for Rotation

Parameterize each camera by rotation and focal length

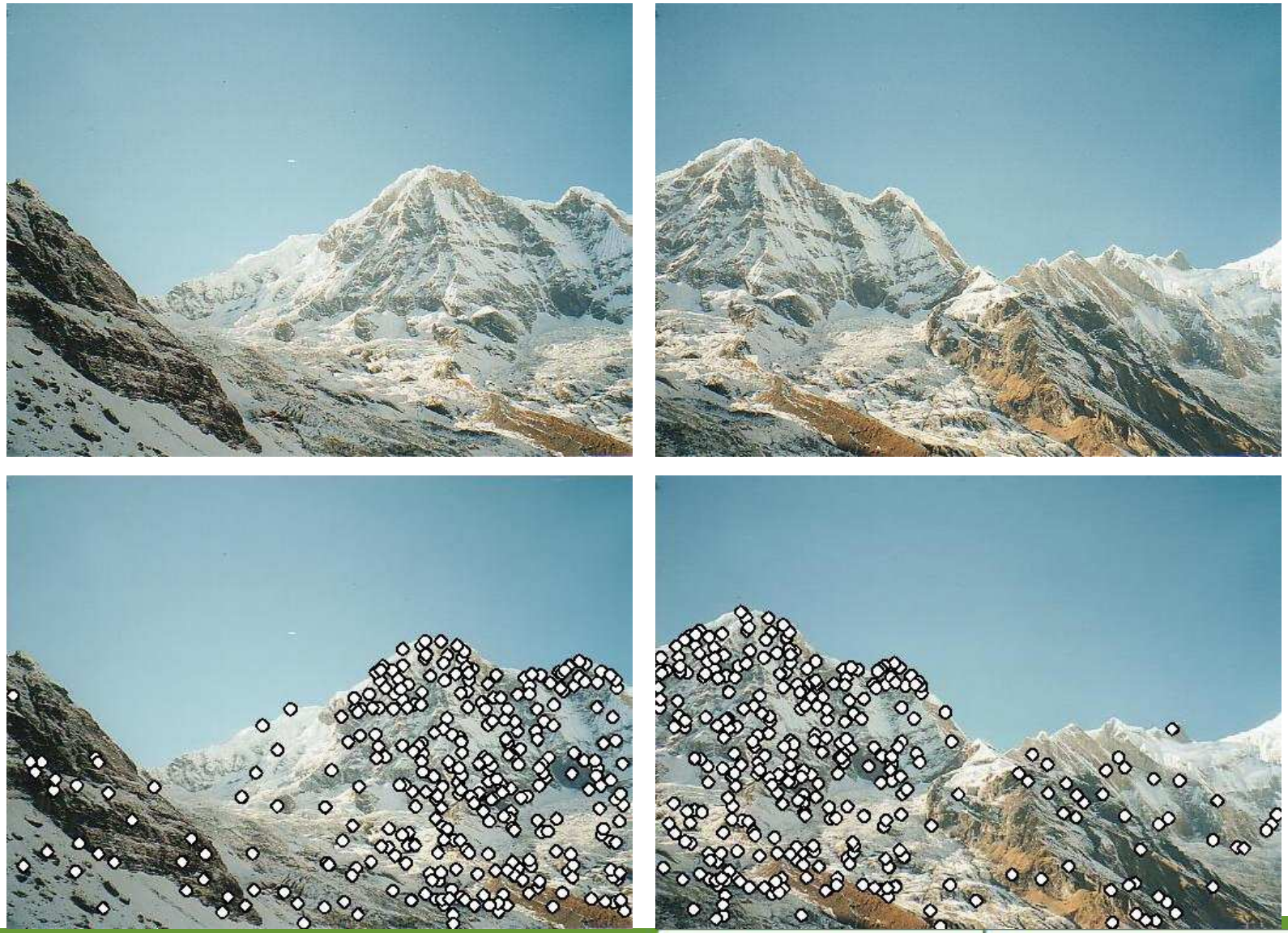
$$\mathbf{R}_i = e^{[\boldsymbol{\theta}_i]_\times}, \;\; [\boldsymbol{\theta}_i]_\times = \begin{bmatrix} 0 & -\theta_{i3} & \theta_{i2} \\ \theta_{i3} & 0 & -\theta_{i1} \\ -\theta_{i2} & \theta_{i1} & 0 \end{bmatrix}$$

$$\mathbf{K}_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

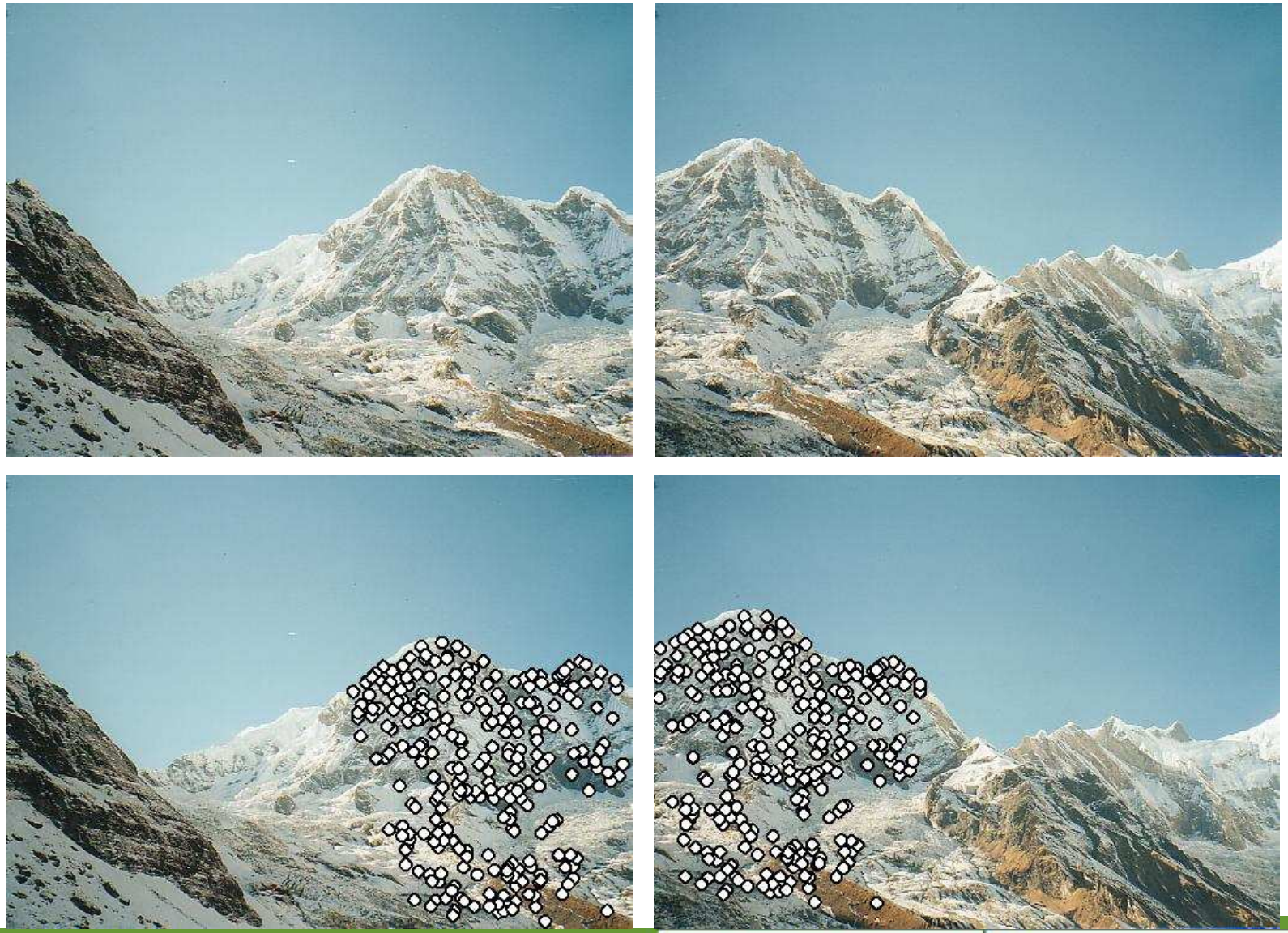
This gives pairwise homographies

$$\tilde{\mathbf{u}}_i = \mathbf{H}_{ij}\tilde{\mathbf{u}}_j, \;\; \mathbf{H}_{ij} = \mathbf{K}_i\mathbf{R}_i\mathbf{R}_j^T\mathbf{K}_j^{-1}$$

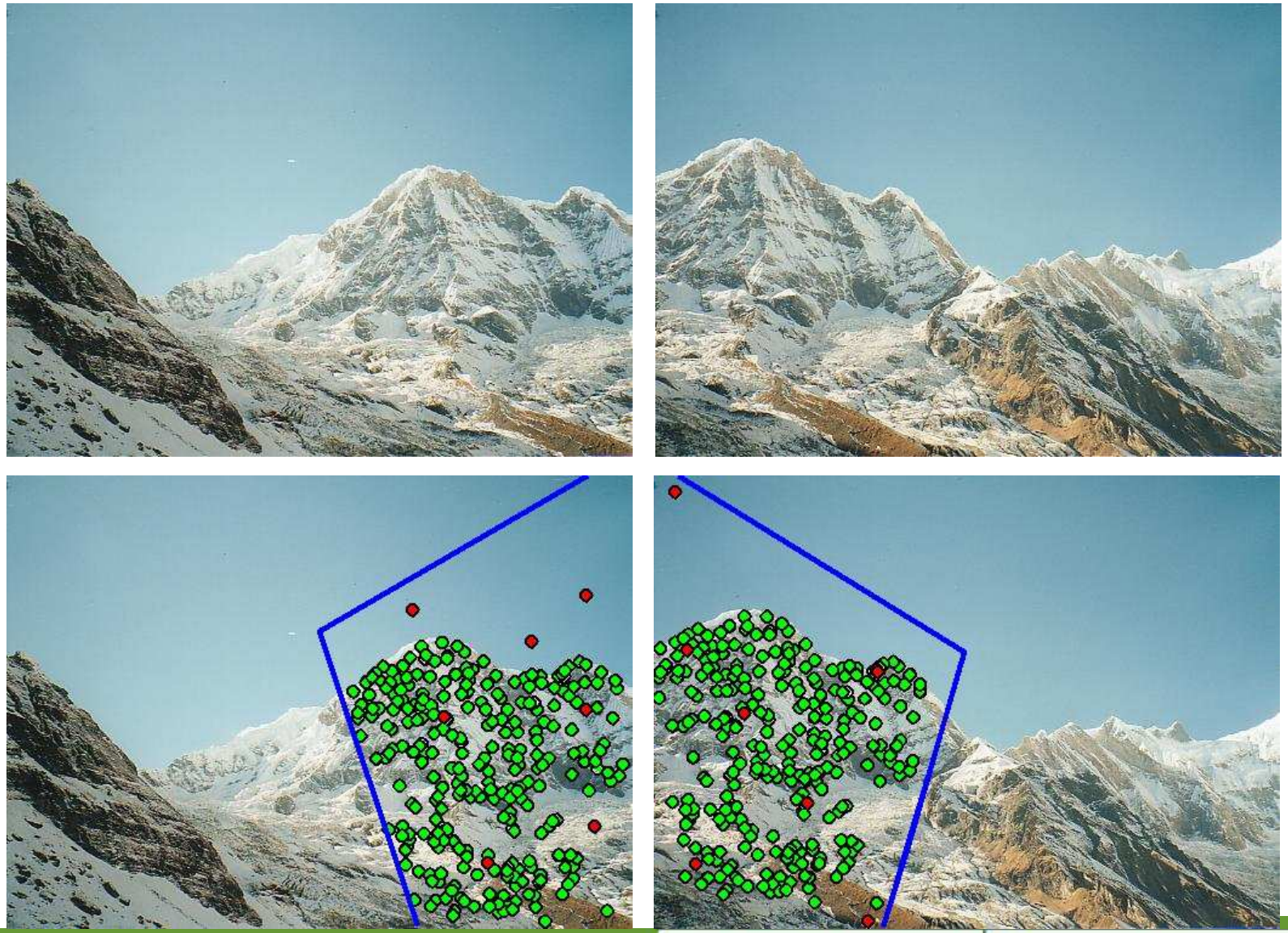
RANSAC for Homography



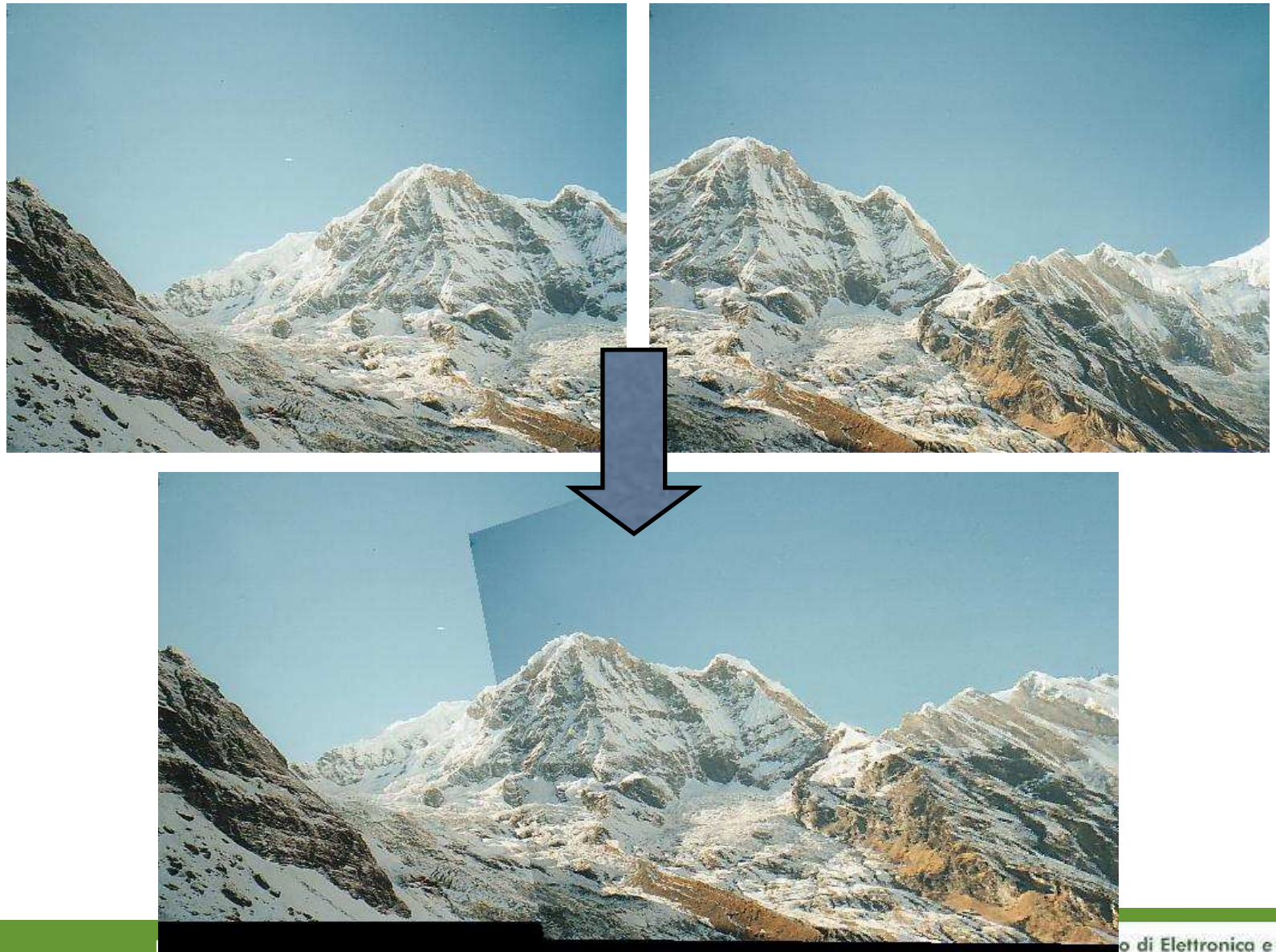
RANSAC for Homography



Probabilistic verification



RANSAC for Homography



Example: Recognizing Panoramas (Brown & Lowe 2003)

M. Brown and D. Lowe,
University of British Columbia

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

* M. Brown and D. Lowe. Automatic Panoramic Image Stitching using Invariant Features. International Journal of Computer Vision, 74(1), pages 59-73, 2007* M. Brown and D. G. Lowe. Recognising Panoramas. In Proceedings of the 9th International Conference on Computer Vision (ICCV2003), pages 1218-1225, Nice, France, 2003

Example: Recognizing Panoramas (Brown & Lowe 2003)

1D Rotations (θ)

- Ordering ⇒ matching images

Example: Recognizing Panoramas (Brown & Lowe 2003)

1D Rotations (θ)

- Ordering ⇒ matching images

Example: Recognizing Panoramas (Brown & Lowe 2003)

1D Rotations (θ)

- Ordering ⇒ matching images

Example: Recognizing Panoramas (Brown & Lowe 2003)

1D Rotations (θ)

- Ordering ⇒ matching images

2D Rotations (θ, φ)

- Ordering ⇒ matching images

Example: Recognizing Panoramas (Brown & Lowe 2003)

1D Rotations (θ)

- Ordering ⇒ matching images

2D Rotations (θ, φ)

- Ordering ⇒ matching images

Example: Recognizing Panoramas (Brown & Lowe 2003)

1D Rotations (θ)

- Ordering ⇒ matching images

2D Rotations (θ, φ)

- Ordering ⇒ matching images

http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Science fiction?

POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

3D Structure from Visual Motion

M. Matteucci, *Vincenzo Caglioti, Marco Marcon,*
Davide Migliore, Domenico G. Sorrenti

matteucci@elet.polimi.it
Dipartimento di Elettronica e Informazione, Politecnico di Milano
Artificial Intelligence and Robotics Lab